

///Brady

Special
Rate Offer
on QuickKeys
Software Inside!

THE AUTOMATIC MAC

 QUICKKEYS™ To Time-Saving Macros

More
than 600
ready-to-use
macros for
automating
tasks and
increasing
productivity



Demonstration version of
QuickKeys, the best-selling
macro-generating
program for the Mac

•
All the macros covered
in the book, ready-to-run

•
Four exclusive QuickKeys
extensions not available
anywhere else



Fred Terry

The Automatic Mac: QuickKeys to Time-Saving Macros

Fred Terry



Brady Books

New York London Toronto Sydney Tokyo Singapore

Copyright © 1992 by Fred Terry

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



Brady Books

A Division of Prentice Hall Computer Publishing
15 Columbus Circle
New York, NY 10023

ISBN: 0-13-52820-X

Printing Code: The rightmost double-digit number is the year of the book's printing; the rightmost single-digit number is the number of the book's printing. For example, 92-1 shows that the first printing of the book occurred in 1992.

95 94 93 92 4 3 2 1

Manufactured in the United States of America

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: Most computer and software brand names have trademarks or registered trademarks. The individual trademarks have not been listed here.

CE Software, QuicKeys, and the QuicKeys logo are trademarks of CE Software.

Dedication

For Leesa...

who gave me the strength and confidence to finish

Credits

Publisher
Michael Violano

Managing Editor
Kelly Dobbs

Editor
Tracy Smith

Developmental Editor
Perry King

Production Editor
Kristin Juba

Technical Reviewer
Jeff Knee

Book Designer
Scott Cook

Cover Designer
HUB Graphics

Editorial Assistant
Lisa Rose

Production Team
Terri Edwards
Mark Enochs
John Kane
Phil Kitchel
Juli Pavey
John Sleevea

This book was written in Microsoft Word 5.0 and typeset using Aldus PageMaker 4.2. Macintosh IIfx computers with Radius Color Pivot monitors were used. Exposure Pro was used to capture the screen shots, and Canvas 3.0 was used to create the graphics. The following fonts were used: Univers, Palatino, Helvetica, PIXymbolsCmdKeys, PXymbolsKeys, S2113PIXymbols, Tekton.

Acknowledgments

I first wrote about QuicKeys in an early 1988 article for *MACazine*. I was amazed with the capabilities QuicKeys brought to the Macintosh. In 1990, CE Software asked me to rewrite the user's manual for QuicKeys 2. As we rushed to meet the deadlines, I realized there weren't enough examples in the manual to show the power and breadth of the program. This book comes from that realization.

It is a trite, but harshly real, statement that a book is not the result of one individual's efforts. Having said that, let me name the people and their efforts to keep me honest.

I want to thank Leesa, my wife, who puts up with my ongoing distraction about Macish things and who read several drafts of the manuscript. Sommer and Keegan provided the daily reality checks that balanced my otherwise unbalanced condition. I also want to thank—in no particular order—Rick Brownrigg, Brett Bennett, and Dan Littman for their encouragement, as well as Rick Barron, Robert Soloman (wherever you are), and Dave Winer for letting me bother them with questions. Leonard Rosenthol gave me eleventh hour assistance with Apple events, and Aladdin provided a solution to a rather large problem. Particularly helpful were the folks at CE Software; Dave Loverink, Jeff Knee, Sue Nail, and Darrel Christensen provided me with the latest information and constant feedback. Jeff's technical edit and suggestions have saved me from most of the embarrassing mistakes so common to writers. Any errors remaining in the book are the result of my own wetware failures and occur in spite of all the help these friends have given me.

Matt Wagner, my agent, and Tracy Smith, my editor, helped make this book a reality. Matt, I owe you money, and Tracy, I owe you a debt of gratitude for not giving up on this project. I also want to thank Perry King at Brady Books for his edits and for helping push this book through to completion. I would be more than remiss if I didn't mention Kristin Juba whose attentiveness and skill turned my coffee-stained manuscript into the finished book you hold in your hands.

Simeon Leifer, Fred Reed, and Christopher Wysocki deserve special mention. Simeon wrote the ClipboardAction, MenuAction, MessageAction, and WindowAction extensions for the book that made it

possible for QuicKeys to do very amazing things. Fred Reed rewrote his popular Rebound INIT so it would be compatible with System 7 and allowed me to include it with the book. Chris Wysocki allowed me to include his Escapade control panel device with the book. Thanks, guys.

Finally, early morning sessions wouldn't have been possible without massive doses of Rush, Zappa, Clapton, Rundgren, and Stranger Chicken from the Magic Wok.

Most importantly, I want to thank Don Brown for writing one of the coolest programs I have ever used. Without his beginning, this book wouldn't be.

Fred Terry
Lawrence, KS
August 1992

Foreword

I won't pretend. I can't possibly convince you that you don't need this book. Sure, we do everything we can to make QuicKeys easy to learn and easy to use, but that's a tall order for a product that can't even be described in one sentence. When I try (e.g., QuicKeys is the centerpiece in how any individual interacts with his or her computing environment; QuicKeys is a tool with which any Macintosh user can create macros and shortcuts; or, QuicKeys is an easy way to make using your Mac easier and faster), I inevitably understate the breadth and potential of the product. I should know—I was a QuicKeys “expert” consultant before I came to CE Software, but even I was amazed to find out what I didn't know about QuicKeys.

That's because QuicKeys is actually many products, depending on what you need and what you want it to be. It's something to replace one or more mouse actions with a keystroke. Or, it's something to replace one or more keystrokes with a mouse action. It's something to simply substitute one keystroke for another. Or, it's something to write elaborate control programs for your database manager. Whenever I make a presentation at a user group meeting or trade show, I inevitably find that experts who have used the product for years discover new capabilities they never knew existed and that Macintosh users who have never used QuicKeys discover new ways of working they never dreamed of. It's why I love the product, why managing its marketing is a challenge, and why you need this book.

I want you to find out what QuicKeys can do for you—things you may never have dreamed of or thought you could do or things nobody else has discovered. I want you to absorb this book and find your own new use for QuicKeys. I want you to discover the power and freedom that awaits you and that can grow every day in all you do with your Macintosh. And then, pass that power and freedom along to a friend.

Jim Sheldon-Dean
QuicKeys Team Leader

About the Author

Fred Terry grew up on a farm in southwestern Oklahoma at a time when computers weren't in every household appliance. He started life as a medieval scholar, but decided he liked playing with computers and networks better. Fred has contributed articles to a number of computer magazines, including *MacWEEK* and *Computer Shopper*, and has written a fair number of user manuals, including the QuicKeys manual. He has a statistically correct family and lives (and sleeps, occasionally) in Lawrence, Kansas.

Contents

Introduction	xv
Who Should Buy This Book?	xv
It's a Cookbook	xv
It's a Great Deal	xv
How to Use the Book and the Disk	xvi
What You Need	xvi
What the Chapters Contain	xvii
 Chapter 1. Macros and QuicKeys	1
A Brief History of Macros	1
Bringing Macros to the Macintosh	2
The Mouse Tracking Programs	2
Tempo	2
QuicKeys	3
MacroMaker	4
The Future of Macros	6
An Aside About Terminology	8
Installing the QuicKeys Demo	8
If You Don't Own QuicKeys... ..	9
If You Already Own QuicKeys... ..	11
 Chapter 2. Macintosh Macro Strategies	15
Planning Your Macros	15
Creating a Macro to Open Your Document	16
Strategies for Flawless Macros	20
I've Run Out of Keys!	23
Getting Organized	27
Let Macros Make You More Productive	28
Let's Get on with It	29
 Chapter 3. Macros for Everywhere	31
Standardizing Your Programs' Shortcuts	31
Menu Macros	32
A Menu QuickKey	33
Text Editing Macros	37
Using Alias QuicKeys	38

Replacing Your Mouse	40
Managing Windows	41
Hide Window Macro	41
Drag Window Home Macro	44
Drag to the Right	46
Tiling Windows	48
Instant Screen Saver	51
Making Programs Smarter	52
Archiving Cuts	53
Transpose Character	55
Transpose Word	57
Print Setup	58
Spell Check, Then Print	62
Running Your Macintosh on Auto-Pilot	65
Start Me Up	65
Save All	69
Timed Archives	72
Time Logger	76
Change Creator	79
Extended Find	83
And Now	86

Chapter 4. Text Tricks87

General Macros	87
Finding Two Spaces	87
Inserting a Mark	91
Finding a Marker	93
Jumping Between Pages	97
Combining Files	99
Adding Graphic Markers	106
Inserting Graphics	109
More Powerful Formatting Commands	114
General Formatting	114
Boldfacing a Paragraph's First Line	116
Creating Styles	117
Creating Templates	119
Letter Template	120
Creating a Bibliography Template	122
Print Letter and Envelope	125
Moving On	130

Chapter 5. Macros for Spreadsheets 131

Interface Changes	131
Adding Keyboard Shortcuts	132
"Fixing" the Enter Key	134
Changing Worksheet Views	135
Automatic Calculations	144
Sum Macro	145
Equation Solver	146
Formatting with a Single Keystroke	151
Formatting Cells	151
Formatting a Worksheet	156
Automated Forecasting	164
Moving On	166

Chapter 6. Macros for Graphics Programs 167

Selecting Tools	167
Selecting Tools with a Click	168
An Aside About Pop-Up Palettes and Windoids	168
Selecting Difficult Tools	170
Using Apple Events to Select Tools	172
Toggles for Rulers	177
Reduce and Magnify	180
Automatic Formatting	184
Intelligent Tools	184
Wrap Text	190
Smart Functions	192
What's Next?	199

Chapter 7. QuickKeys and HyperCard 201

Personalizing HyperCard	201
Menu Shortcuts	202
Selecting Tools or Patterns	204
Navigating HyperCard	207
Open Home in a New Window	210
Automating Your HyperCard Sessions	212
Scripting Shortcuts	212
Cut to Background	216
Changing the User Level	218
Configuring Buttons and Fields	224
Now What?	228

Chapter 8. Macros for Communications.....229

General Communications Macros	230
Intelligent Shortcuts	230
Automate Log-Ons	234
Adding Missing Features to Programs	239
Automating E-Mail and Remote File Sharing	243
Getting Your Mail from the Office	243
Backing Up Files Remotely	250
Disconnecting Remote Access	257
CompuServe Information Manager	259
Get Mail for Information Manager	259
Set Up a Reply	263
America Online	265
Get Electronic Mail	265
Open New Files Folder	274
Automated Communications	278
What Next?	286

Chapter 9. QuickKeys and Databases287

Automating Database Creation	287
Adding Keyboard Shortcuts	287
Setting Up Fields	291
Paging Through Database Records	296
Aligning Objects	301
Automating Data Entry	302
Entering an Address	303
Preparing a Search	304
Automating Conversions	310
Now What?	314

Chapter 10. Finder Tricks.....315

Automating the Finder	315
Keyboard Shortcuts	316
Trash Old Files and Shut Down	318
Empty the Trash	321
Printing Files While You Sleep	325
Automating File Sharing	329
Setting File Sharing Privileges Automatically	333
Rebuilding the Desktop	340
Quitting the Finder	342
Managing Files	345
Showing Buried Folders	345

Backing Up Project Files	347
Backing Up Modified Files	357
Last But Not Least	361

Chapter 11. QuicKeys and System 7363

What Are Apple Events?	363
Making System 7 More Automatic	364
Startup File Sharing	365
Automate Publish and Subscribe	374
The Future of Macintosh Computing	376
Look Up an Address	377
Send Apple Events to StuffIt Lite	383
QuicKeys and Scripting Programs	390
UserLand Frontier	390
ControlTower	390
Where to Now?	391

Appendix A. Where Do I Find This Stuff?393

Commerical Software and Hardware	394
Communications and Electronic Mail	394
Databases and PIMs	395
Graphics and Page Layout	395
Hardware and Other Stuff	396
HyperTextual	397
Macroing and Scripting	397
Spreadsheets	397
Utilities	398
Word Processors and Text Editors	399
Shareware and Public Domain software	400
Other Addresses and Numbers	400
And Lastly... ..	401

Appendix B. Getting to Know QuicKeys403

The QuicKeys Editor	403
The Memory Indicator	404
QuicKeys Menus	406
The File Menu	406
The Define Menu	407
The Options Menu	411
The QuicKeys Listing	412
QK Icons	413
Instant QuicKeys	415

Some Limitations	416
Ten Keyset Limit	416
Don't Give Keysets the Same Name	417
Limit Keysets to 32K	417
Importing Sequences	417

Appendix C. Macros and the Disk423

Macros Listed by Chapter	423
Contents of the Disk	427
Demo QK2.sea	427
Macros Folder	427
Extensions Folder	427
Rebound 1.0	428
Escapade 1.3.2	428

Index429

Introduction

Who Should Buy This Book?

Given the number of books on the shelf that vie for your time (not to mention your money), let me help you decide if this book is right for you. I'll try to be as brief as possible before I stick my hand in your pocket.

To get the most out of this book and QuicKeys, you should have some basic Macintosh skills. You should know how to copy files from one disk to another and double-click a program to run it. You should know how to use the mouse to choose a command from a menu and be able to navigate the standard Open and Save dialog boxes in Macintosh programs. If you have not already acquired these skills, please refer to your *Macintosh User's Guide*. This book covers a broad spectrum of macros. It contains macros that any beginning Macintosh user would find useful, and my hope is that intermediate and advanced users will find tricks here they have never seen before.

It's a Cookbook

The Automatic Mac is a cookbook of macros for QuicKeys, CE Software's popular macro program. Every chapter is loaded with macros that make your computing life simpler, quicker, and ... well ... more automatic. Like any cookbook worth its salt, *The Automatic Mac* provides the recipe for creating each macro, offers hints and tips for getting the macro right the first time, and then offers suggestions for modifying the macro to perform other tasks. In some instances, the macros may not be exactly what you are looking for. Let's face it, we don't all have the same ingredients in our kitchens; however, I provide enough information about the way I've created the macros that you can very easily modify them to suit your purposes. While most cookbooks don't come with the dishes already prepared, *The Automatic Mac* includes a copy of the final macros. All the macros described in this book are included on the enclosed utility disk. If the macros suit your needs in their original form, all you have to do is install them and begin automating your life.

It's a Great Deal

Finally, this book/disk combination is a great deal. If you aren't familiar with QuicKeys, a demonstration version of the program is included on

the disk, and you'll find a \$15 rebate coupon at the back of the book. The demo version of the program works for 30 days. This should give you enough time to test drive it and find out if you want to buy the full version. When you decide to buy the full version, you can get it from your favorite dealer or software warehouse; wherever you can find it the cheapest (my favorite place is MacConnection in Marlow, NH). Then send CE Software a copy of the receipt and the coupon from the back of the book. In return, they will send you a check for \$15. You can't beat that.

How to Use the Book and the Disk

This book is meant to be browsed while you are sitting at your Macintosh. You should have QuickKeys loaded and your program running. This book isn't a novel; it's a cookbook. You yank a cookbook off the shelf and thumb through the pages when you are completely out of ideas for dinner. *The Automatic Mac* can serve a similar function. When you are really tired of doing the same task over and over again, I hope it supplies the exact macro you need to automate the task. However, since no two people work the same way, have their Macintoshes configured the same way, or have the same programs, I hope that at the very least this book provides you with ideas for creating a macro and automating your task. If I have created a macro using a different program than you have, don't despair. I have tried to adapt the macros to the more popular programs and have included keysets for those programs on the disk.

I have tried to make the chapters as discrete as possible, so you can read them in any order that you want. For example, you may only want to read those chapters that apply to the programs you use every day. Of course, you may miss a few juicy tidbits if you don't read the book from cover to cover. You should read Chapter 3 since it describes strategies you will want to use as you create your own macros. If you are new to QuickKeys, you should read Appendix B, *Getting to Know QuickKeys*; it is a brief description of the various components and features of QuickKeys.

What You Need

For starters, you should have a Macintosh Plus or greater with a hard disk. You should have System 6.04 or later installed. If you don't have the System 6.04 or later installed, you can get a free copy from your dealer or your local Macintosh User's Group. Macintosh system software is also available for downloading from the America Online, CompuServe, CONNECT, and GENie electronic services. These services charge for every minute that you are connected, so you may want to find a dealer or a user group before you resort to downloading the

software. If you have a connection to the Internet through your company or university, you can get system software directly from Apple by connecting to their computer at the Internet address `bric-a-brac.apple.com` using Telnet. The Internet is a worldwide network that connects corporations, universities, government agencies, and military sites. If this sounds like a foreign language, you can find out more about the Internet from your local computer wizard.

What the Chapters Contain

The opening chapters introduce you to the concept of macros and QuickKeys and give you some pointers for creating and managing macros. The remaining chapters in the book give you time-saving macros for the programs you use every day and show you how to create and modify them for your own work habits.

Chapter 1 contains a brief history of macros in the computer industry, starting with the earliest mainframes and ending with the Macintosh sitting on your desk. At the end of the chapter, we install the QuickKeys demo and get ready to create some macros.

Even though we begin Chapter 2 by creating a macro together, it is mostly philosophy. I wouldn't call it the Tao of macros, but I do outline some simple strategies to make creating and testing macros easier. I also explain how to organize and use macros efficiently.

In Chapter 3, you actually begin creating macros. The macros in this chapter are for use in every kind of program. My favorite macros in this chapter let me change programs to behave like Microsoft Word. Yes, that is a bit perverse.

Chapter 4 describes macros for manipulating and managing text in word processors and text editors. Do you need to combine and format a dozen files? This chapter shows you how to do that.

Chapter 5 shows you how to combine QuickKeys with your favorite spreadsheet for power formatting. While most spreadsheet programs have built-in macro capabilities, the macro language is confined to the program. This chapter shows you how to combine spreadsheet macros with the systemwide power of QuickKeys.

Chapter 6 demonstrates how you can use QuickKeys to perform graphics tricks. You will find keyboard shortcuts for selecting tools, creating objects, and jumping between views. There are also macros that make

your tool selections smarter tools. These macros select a tool and your settings for that tool.

Chapter 7 contains HyperCard scripting shortcuts as well as macros for selecting tools and creating templates.

Chapter 8 shows you how to make telecommunications programs easier to use. It has macros for automating scriptless telecommunications programs like America Online and CompuServe's Information Manager.

Chapter 9 provides macros that automate setting up fields and entering information in databases. There are also macros that sort your database and generate reports at the press of a keystroke.

Chapter 10 contains macros that tame the Finder. In addition to keyboard shortcuts for changing the Finder view, emptying the trash, restarting and shutting down your Macintosh, you will find macros for moving and deleting files. This chapter also contains macros that automate file sharing and allow you to take advantage of the Finder-specific Apple Events in System 7.

Finally, Chapter 11 demonstrates the power that System 7 can give your macros. It shows you how to use Apple events to look up an address and automatically insert it in your letter without ever leaving your word processor. The chapter also shows you how to use Apple events to create a chart in your spreadsheet and incorporate it in your document and how to get the most out of System 7's publish and subscribe features.

You will also find three appendices at the end of the book. Appendix A, "Where Do I Find This Stuff?," tells you where you can find the software and hardware I mention in the book. It includes the companies' addresses and phone numbers. Appendix B, "Getting to Know QuicKeys," is an introduction to QuicKeys for people who have never used it before. It describes all of QuicKeys components. This appendix is not a replacement for the QuicKeys manual, but it will get you started. Appendix C, "Macros and the Disk," contains a list of the macros in the book and an explanation of the disk contents.

Though this overview of the chapters is scanty, you should be able to get the flavor of the book. Of course, the real test of any cookbook is to take it home and make a couple of dishes from it. So, if you are hungry for an "automatic Macintosh," we will take a quick look at the history of macros, install QuicKeys (or launch it if you already have it installed), and make some macros together.

1

CHAPTER

Macros and QuicKeys

If you aren't familiar with the term macro, I'll give you a little history at the beginning of this chapter. I'll also talk a little about the future of macros on the Macintosh and the much-heralded arrival of AppleScript. At the end of the chapter, I'll walk you through the installation of the QuicKeys demo and explain the different programs and folders installed on your hard disk.

A Brief History of Macros

The term macro—in one form or another—has been around since the earliest days of the computer, but its connotations have shifted from those early days. It is a clipped form of the longer word “macro-instruction” and was originally used in connection with the macro assemblers on mainframe computers. The programmer would enter a short form of an instruction, which the assembler would expand. In programming circles, the term is still used to refer to languages with a macro expansion facility. Over the years, the term has shifted to refer to any special-purpose control language as well as to the terminate and stay resident (TSR) or initialization programs (INIT) that enhance the keyboards on our personal computers.

Batch files, which first appeared on the personal computer when DOS 2.0 was released in early 1983, have a special kinship to macros. Batch files are a macro language for the operating system; all of the commands in a batch file are DOS (disk operating system) commands. DOS reads each line of the file and executes it as though you were typing it at the DOS prompt. While batch files are capable of responding to conditions and branching to other batch files, they are confined to the operating system. They cannot span the operating system and programs.

The market for macro programs has evolved from those early stages of crude batch processing programs to very slick programs that can operate at both the system and individual program level and do most of the work for you.

Bringing Macros to the Macintosh

From the beginning, Macintosh users have wanted the same capabilities for the Macintosh that they were familiar with on other computer platforms. But where the commands in batch files are issued at the system level, the Macintosh interface posed special problems for developers. How do you create a macro language for a computer that doesn't have a command line interface? Neff Systems circumvented the interface by creating the MacPS utility program. MacPS was released in 1987 as a commercial product and had a language that allowed you to write "playscripts" for system functions. The playscripts were text files that you could edit with any text editor. Even with its limited vocabulary, MacPS could open documents, desk accessories, and programs, change the default disk and directory, copy files, create new folders, and delete files and folders. Like the DOS batch language, it could branch to another playscript and then return.

MacPS was a radical departure from the growing trend of "mouse tracking" programs. Most developers tried to capitalize on the Macintosh's interface by recording keyboard input and mouse actions.

The Mouse Tracking Programs

The very first macro program for the Macintosh was MacTracks. Written by Allen Wooten, MacTracks was marketed in 1985 by Assimilation Process, a company formed by two ex-Apple employees. The program began shipping in April or May of 1985. Unfortunately, the company did not last very long, and MacTracks, which was plagued by some annoying problems, was not picked up by another company.

Tempo

Tempo first shipped in January 1986. It was created by MacMan of Houston, TX and marketed by Affinity Microsystems. Rick Barron of Affinity approached MacMan about writing Tempo after he saw a preview of Lotus' Jazz and noticed that it did not have macros. Tempo was the first program that tacked itself onto the end of the menu bar and successfully coerced the Clipboard. It was also the first program to go

from one program to another without traversing the Finder and even survive a launch to another program. Tempo II shipped a little over two years later in May 1988. The Tempo Plus Tools upgrade shipped in April 1990 and was followed by Tempo II Plus in January 1991.

Tempo is the only program for the Macintosh that has built-in conditional branching. In other words, if a certain dialog box or window appears, the program performs one macro. If a different dialog box or window appears, the program performs a different macro. The one major drawback to Tempo is that it is difficult to edit a Tempo II Plus macro. If you want to change portions of the macro after you have created it, you must step through it until you reach the point where you want to make the change.

QuickKeys

QuickKeys was first released in January 1987. The best thing about QuickKeys was that it was easy to edit sequences. The Sequence editor interface was much like it is today, and you could see all of your steps in a scrolling window. Unfortunately, in the early versions of QuickKeys, sequences could only have 30 steps, and each QuickKey in a sequence had to have a keystroke assigned to it. Users quickly ran out of free keys on their keyboards. When QuickKeys 2 was released in September 1990 all of that changed. The individual QuickKeys in sequences don't have to have keystrokes assigned, and sequences are only limited by disk space.

Another important feature in QuickKeys is that it, like Tempo II Plus, can be extended. You can add new features to the program by adding new extensions. Extensions are small programs that work within QuickKeys. When QuickKeys shipped, it had extensions to mount and unmount AppleShare servers, select printer devices, change the sound or the speaker volume, or open control panels. Since then, many different extensions have been written for QuickKeys that let you compress files, wait for menu and window changes, and add repeat loops and messages to your macros.

Like the other programs, QuickKeys creates macros by watching what you are doing and recording your actions as separate QuickKeys. If, for instance, you record a macro that opens the Page Setup... dialog box, changes the settings, then opens the Print... dialog box in your word processor, and clicks the OK button, QuickKeys would record your actions as individual menu selections, several clicks for the settings, and a click for the OK button (Fig. 1-1). This makes QuickKeys' macros very easy to edit. You can change any of the individual QuickKeys by opening

it in the Sequence editor. You can insert changes by moving the insertion arrow and adding more individual QuicKeys with the Define menu or the Record More, Import, or Literal buttons.

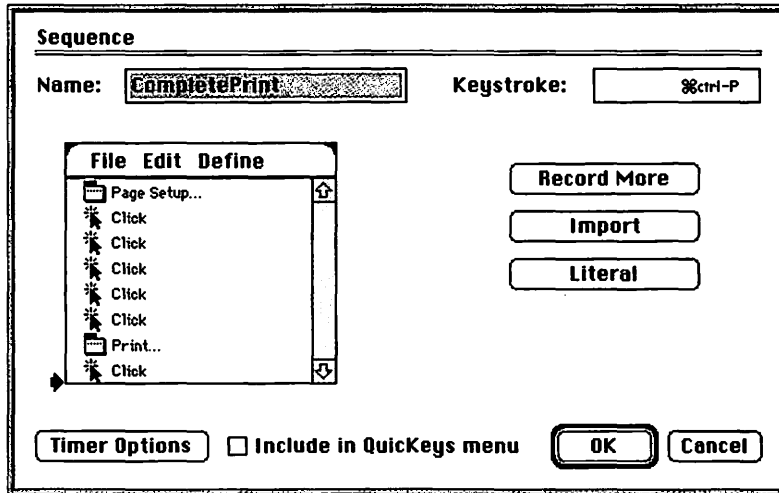


Figure 1-1: The QuicKeys Sequence editor.

Like its sister programs, QuicKeys suffers from a weakness. It currently does not let you create conditional macros. Simeon Leifer has written several extensions that branch or jump to another QuicKey. Simeon has almost single-handedly doubled the number of extensions available for QuicKeys. You can find CursorWait, MenuAction, MenuDecision, ProcessSwap, WhichPrinter, WindowDecision, and WindowWait on any of the commercial electronic services. Besides the CursorWait extension, which is included on the companion disk, Simeon has written several extensions exclusively for this book. When you install the QuicKeys Demo, you'll notice ClipboardAction, MenuAction, MessageAction, and WindowAction in the extensions menu. If you are already a QuicKeys owner, you'll find Simeon's handiwork in the Extensions folder on the utility disk. You will need to install them using the Extensions Manager and following the directions in your QuicKeys manual.

MacroMaker







Ironically, Apple's macro program was the last to be released. MacroMaker wasn't ready until 1988 after Apple had been shipping its extended, or battleship, keyboard for a year. Since very few Macintosh

programs had built-in function key assignments, the keyboard had a maddening wealth of keys that you couldn't do anything with unless you owned QuickKeys or Tempo. MacroMaker has been provided free with the System 6.x, but when System 7 began shipping it was conspicuously absent from the disks. Apple had not updated the program because they were hoping to ship Apple Script, the new system-level scripting language, with System 7. Unfortunately, Apple Script was delayed and didn't ship with System 7 in May 1991.

Go Technology is currently marketing Hot Keys, a collection of macros and keyboard templates for popular applications. The Hot Keys macros are MacroMaker-based, and Go Technology has licensed MacroMaker from Apple and updated it to be semicompatible with System 7.



Figure 1-2: Apple's MacroMaker program. The recorded macros appear in the scrolling box on the right.

Probably the best thing about MacroMaker was that it was provided free with the system software. While the tape cassette metaphor of its interface didn't catch on, MacroMaker validated several new symbols that had been filtering into the Macintosh vernacular (Fig. 1-2). The  (Caps Lock),  (Control),  (Shift),  (Option), and the  (Numeric Keypad) symbols joined the  symbol. The primary flaw of MacroMaker was that you could not edit a macro. If you made a mistake while you were recording the macro or decided that you wanted to add something to the macro later, you would have to record it again.

The Future of Macros

With Apple's announcement of AppleScript, Macintosh macros are coming full circle and returning to the form of DOS batch files and MacPS powerscripts. With the introduction of Apple Events and InterApplication Communication in System 7, Apple has made communication between the system and other programs possible. Apple is providing the hooks into the system software that developers have needed all along.

UserLand Software has already started shipping Frontier. It allows advanced users to write scripts to automate and simplify the Macintosh operating system. I say advanced users because Frontier's scripting language is really closer to a programming language (Fig. 1-3). Frontier's scripts could easily fit into our broad definition of macros. Because they allow you to create systemwide sets of commands, they are somewhat similar to batch files. However, they allow programs to communicate with each other—something batch files were never able to do.

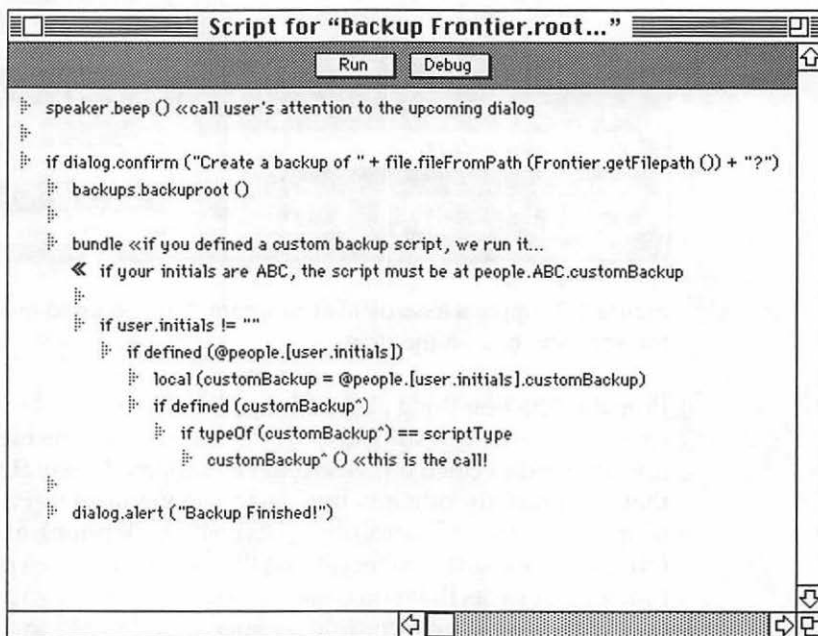


Figure 1-3: One of Frontier's scripting windows. The script looks like an outline.

Simple Software has begun shipping ControlTower. While ControlTower allows you to communicate between programs, it is not a programming language. Its interface is probably the easiest for novice users. It contains boxes where you specify the target program to which you want to send the Apple Event, the message you want to send, and the data you want to send (Fig. 1-4). When you double-click in the message window, a dialog appears with all of the Apple Event suites that the program can accept. To create functional scripts, the user doesn't really need to know very much about the structure or syntax of an Apple Event.

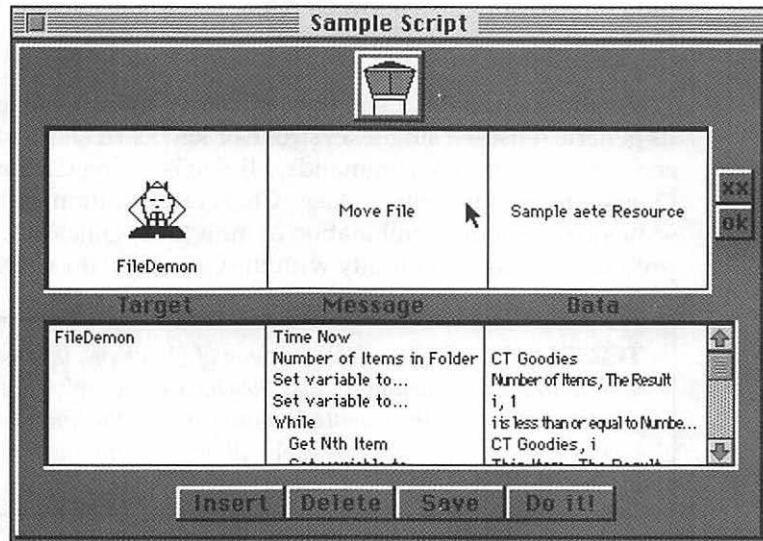


Figure 1-4: ControlTower's interface is much less intimidating to users. You click the target, message, and data boxes in the window to create your Apple Event.

Apple's entry into the system-level scripting market is AppleScript, and it is due to ship in early 1993. Like ControlTower and Frontier, AppleScript will let users write scripts in an editor to automate tasks, and the scripts will allow communication between programs and the Macintosh operating system. Unlike Frontier, AppleScript will be based on an English-like language. It will also have a "watch me" recorder to provide users with a point and click means of generating a script—a feature that Frontier lacks. But for all of their power, these programs cannot replace some of the simplest things QuicKeys does, and many users may find that QuicKeys provides them with all of the power they need in a form they can easily understand and benefit from.

An Aside About Terminology

The original QuickKeys box carried the slogan “Macros Made Easy,” and the phrase was used in the first edition of the user’s manual. When they rewrote the user’s manual, they decided to drop that phrase from the manual in favor of QuickKey. In place of macro, they used the term “QuickKey” to refer to the individual clicks, mousies, specials, and keyboard actions. They used the term “sequence” to refer to groups of individual QuickKeys. Their hope was that dropping a term already loaded with meaning would give the user a new vocabulary to work with and a new way to look at things. Unfortunately, it is sometimes difficult to distinguish between references to the QuickKeys program and QuickKeys.

Throughout this book, I will bow to history and use the term macro in its generic sense—a single keystroke or keyboard shortcut that invokes a command or series of commands. I’ll also be using CE Software’s terms: QuickKeys refers to a single Alias, Click, FKey, Button, or Menu/DA; sequence refers to a combination of individual QuickKeys. Doing this provides a certain continuity with the QuickKeys’ documentation.

TERMS: *If you own the latest version of QuickKeys, you may have noticed in the documentation that they have begun referring to macros as shortcuts. To avoid the potential confusion with the times I talk about keyboard shortcuts or keystrokes, I will not be using the term shortcuts the way they do.*

Now you know a little bit about the history of macros and the different macroing programs for the Macintosh. Now let’s install the demo version of QuickKeys, if you haven’t already, and get down to business.

Installing the QuickKeys Demo

As I mentioned above, the disk included with the book contains a demo version of QuickKeys. It also contains a folder with all of the macros described in this book. If you’ve never used QuickKeys before, you’ll need to install it following the instructions below. If you are already using QuickKeys, you only need to copy the different keysets in the Macros folder using QuickKeys. The “If you already own QuickKeys...” section below tells you exactly how to do it.

If You Don't Own QuickKeys...

To install the demo version of QuickKeys, copy the Demo QK2.sea file to your hard disk. It is a self-extracting archive. You only have to double-click the file to open it, and it extracts the DemoQKInstall program. To install the demo version of QuickKeys under System 6, you just double-click the program. It places QuickKeys, CEIAC, and CE Toolbox in your System folder and creates a QuickKeys folder for your Keysets and Sequences in your Preferences folder. If you are using System 7, the installer places QuickKeys in your Control Panels folder, inserts CEIAC and CE Toolbox in your Extensions folder, and creates folders for your Keysets and Sequences in your Preferences folder (Fig. 1-5).

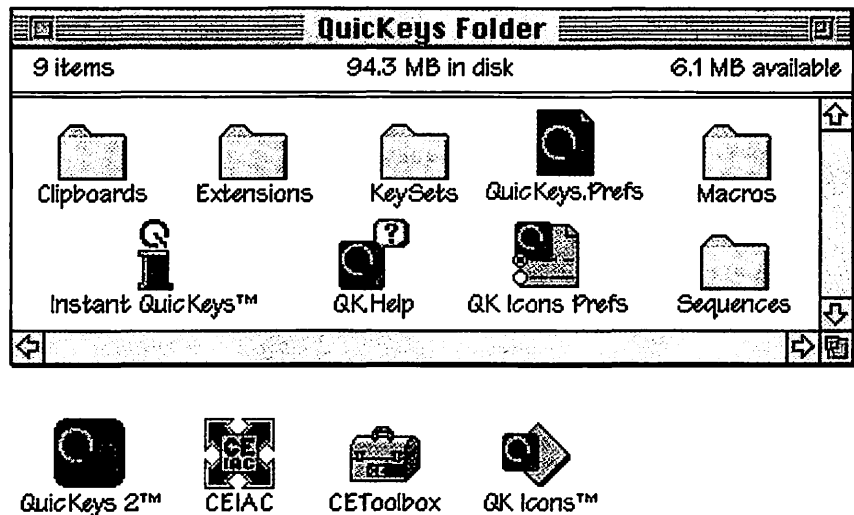


Figure 1-5: The many different components of QuickKeys. You will find the QuickKeys folder in your System folder (or in the Preferences folder of your System folder, if you are running System 7). If you have run Instant QuickKeys, your QuickKeys folder may also contain TeachText files.

When the installation is complete, you are prompted to restart your Macintosh. As your Mac restarts, the QuickKeys icon will appear at the bottom of the screen. You will also notice that extra menu items have been added under the Apple menu (Fig. 1-6).

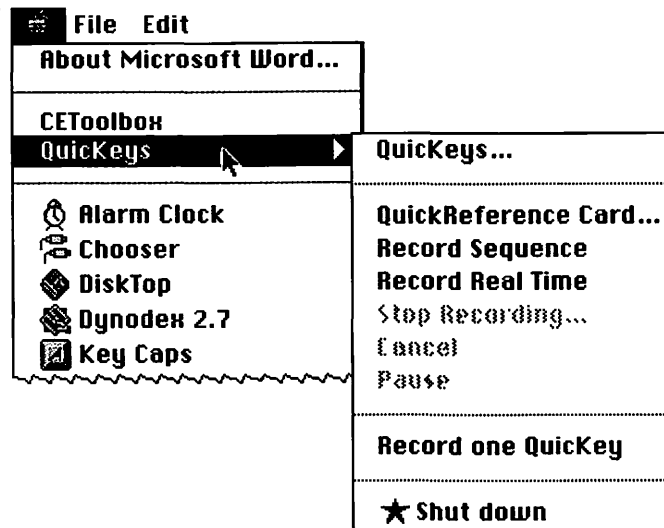


Figure 1-6: After you restart your Macintosh, CEToolbox and QuickKeys will be added to your Apple menu.

CE Toolbox allows QuickKeys to launch programs, display keystrokes, and pop-up a QuickKeys menu when a combination of keys are held down while clicking the mouse. The QuickKeys hierarchical menu lets you open QuickKeys, display the Quick Reference Card, and record sequences and real-time macros. If you are running System 7, you will also notice that CEIAC loaded and is available from the applications menu (Fig. 1-7).



Figure 1-7: If you are running System 7, CEIAC is automatically loaded when you start your Macintosh and is available from the applications menu.

The CEIAC (CE InterApplications Communication) system extension allows QuickKeys to send and receive Apple Events between programs. Apple Events is the messaging language of System 7's InterApplications Communication, which allows applications to exchange information dynamically on the same Macintosh or over a network. With the Apple

Events extension in QuicKeys, you can send Apple Event messages to another application. The Finder Event extension also lets you send Apple Events. It is a specialized extension that sends Open document, Print document, or Get Info Apple Events, among others, to the Finder.

Now that you have QuicKeys installed, you are ready to begin experimenting with macros. If you want to install the macros from the disk, read the next section for QuicKeys owners. All of the macros are contained in the keysets. You will need to open the program for which you want to load the keysets, open QuicKeys, and then open the keyset file.

If You Already Own QuicKeys...

If you are currently using QuicKeys, you will need to copy the keysets from the Macro folder into QuicKeys.

NOTE: Some of the keysets are rather large, so you may want to increase the buffer size in your QuicKeys control panel. Depending on the size of your universal keyset, the QuicKey Expert (64K) setting should be large enough. For more information, refer to Appendix B.

1. Open QuicKeys by selecting QuicKeys from the Apple menu. Then select Open from the QuicKeys File menu (Fig. 1-8).

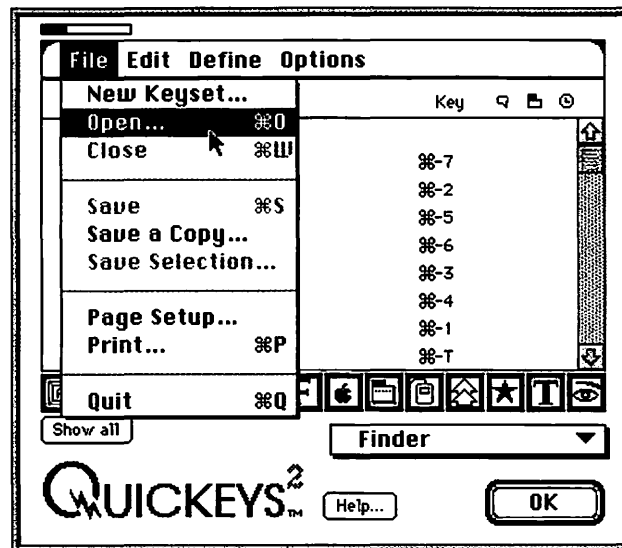


Figure 1-8: The Open command in QuicKeys editor window lets you open a keyset that isn't currently loaded in QuicKeys.

2. QuickKeys automatically opens to the Keysets folder in your System folder. You'll need to use the Desktop or Drive buttons to navigate to the Macro folder on the disk (Fig. 1-9).

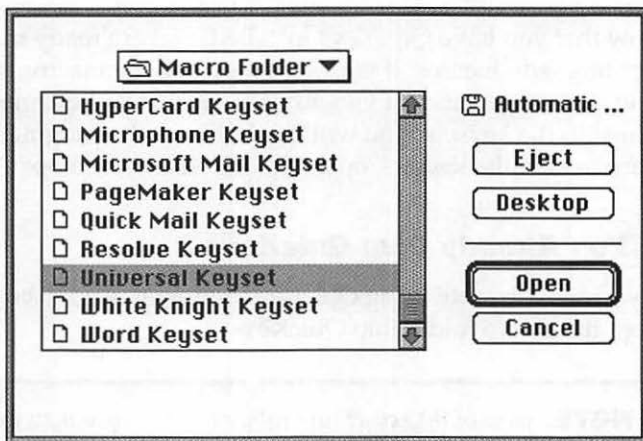


Figure 1-9: Inside the Macro folder on the utility disk.

3. Select the Universal Keyset, for starters, and click the Open button.

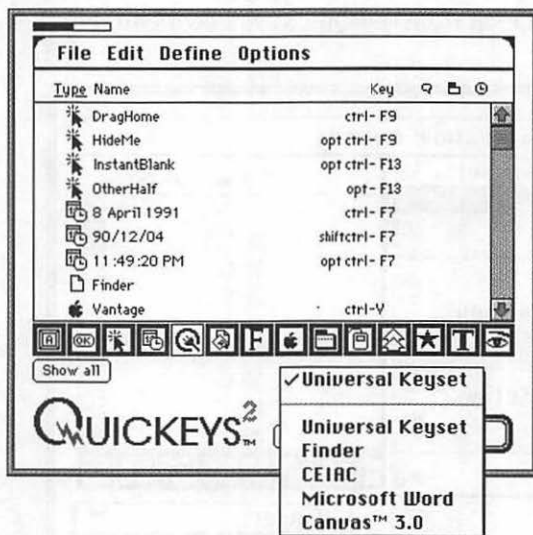


Figure 1-10: The pop-up menu in the QuickKeys editor window. The keysets for all of the currently opened applications are listed at the bottom of the menu. The book's Universal Keyset is above the line.

4. QuickKeys opens the keyset and adds its name to the pop-up menu at the bottom of the QuickKeys window (Fig. 1-10). Now you can copy all or a few of the macros and add them to your existing Universal keyset.
5. Select the macros by holding down the **[Shift]** key, clicking on the first macro in the list, and dragging to the bottom of the list. If you only want to copy a few of the macros, hold down the **[Shift]** key and click on the macros you want to copy.
6. Select Copy from the Edit menu to copy the macros to the Clipboard and then select Close from the File menu. Next, select the Universal keyset from the pop-up menu and then select Paste from the Edit menu to paste the macros into your Universal keyset.

NOTE: *If you are using QuickKeys 2.12 or earlier, you should check to make certain that adding these macros does not cause your Universal keyset to exceed 32K. You can find out how large your Universal keyset is by clicking on the Memory indicator in the QuickKeys window. For more information about the 32K limitation, refer to Appendix B, "Getting to Know QuickKeys."*

You will need to repeat this procedure for each set of macros you want to move into your keysets. Remember to paste them into the correct program's keyset.

NOTE: *Don't drag the keysets into your Keysets folder in the QuickKeys folder. The keysets are compacted and contain the information for sequences. If you do not copy them in the QuickKeys editor window, they will not be unpacked properly.*

Now you are ready to begin experimenting with macros. The next chapter walks you through the creation of a macro. If you are new to QuickKeys you may also want to read Appendix B, "Getting to Know QuickKeys." Chapter 2 also outlines some principles for creating and testing macros as well as strategies for remembering and using your macros.

2

CHAPTER

Macintosh Macro Strategies

If you have QuicKeys installed, you are probably ready to start making macros, but let's take a moment to talk about strategies for creating macros. If you are new to QuicKeys, I realize it seems a little backwards to talk about strategies before we actually go about creating some macros. You will be more productive from the beginning, however, if you develop the habit of thinking about the best way to create a macro. If you are an old-hand with QuicKeys, this discussion may help you think of better ways to create macros. We'll begin by creating a macro together. If you are an old-hand, you may want to skim through the first part of this chapter.

Planning Your Macros

While the Macintosh's graphical user interface is a godsend for users weary of command-line interfaces, it makes creating macros more than a little difficult. It's easy for you to look around in a window until you find the particular document you want, but a macro program trained to open a document by its location on screen may do the wrong thing if the document isn't where it used to be. If you scroll the window so the document has moved or you open another window over the one containing the document, the macro won't work. Or, even worse, it may work in ways you hadn't planned!

Most things you do on the Macintosh depend on screen location, and macros—because they are mimicking your actions—also depend upon documents, applications, menus, and dialog boxes remaining in the same place. If you are running System 6 and have a macro to open the Control Panel and select the fifth cdev (Control Panel device) in the

scrolling list, adding another cdev to your System folder displaces your fifth one; that macro no longer works.

Let's say you want to open the same document every morning. You can create a macro to click on the document icon and then select the Open command from the Finder's File menu. If the icon is moved, the file won't be opened—but something might open instead (Fig. 2-1). The macro will open whatever is selected at the time it is invoked—a disk icon, the trash can, another file.



Figure 2-1: If your document has not moved, the click on the left will select it. If, however, your document has been moved slightly to the left (as the example on the right shows), your click will not select the document.

The first principle of creating a macro is to think of the most reliable way to accomplish the task. For example, avoid creating macros heavily dependent on screen location. You should be warned that this isn't always easy to do. Let's use the example above to show you another approach. A more reliable means of opening a document is to launch the application and then open the document or, even better, just open the document directly. You can select the File... command in QuickKeys' Define menu or, if you are running System 7, the Finder Events from the Extension hierarchical menu.

Creating a Macro to Open Your Document

In QuickKeys, you would create a macro to start up Word, or whatever word processor you use, and open a document by selecting the File command under the Define menu.

1. Open QuickKeys by selecting QuickKeys... from the QuickKeys menu under the Apple menu. You can also open QuickKeys by pressing the keystroke you have assigned to the QuickKeys editor.
2. Select File... from the Define menu (Fig. 2-2). The File Open dialog box appears where you can select a document.

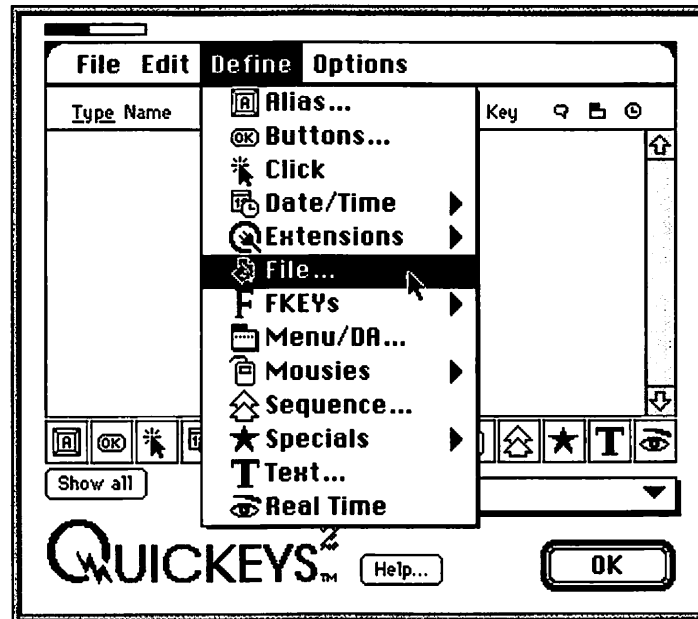


Figure 2-2: Select File... from the Define menu in the QuickKeys editor window.

3. Select a document and click open (Fig. 2-3).

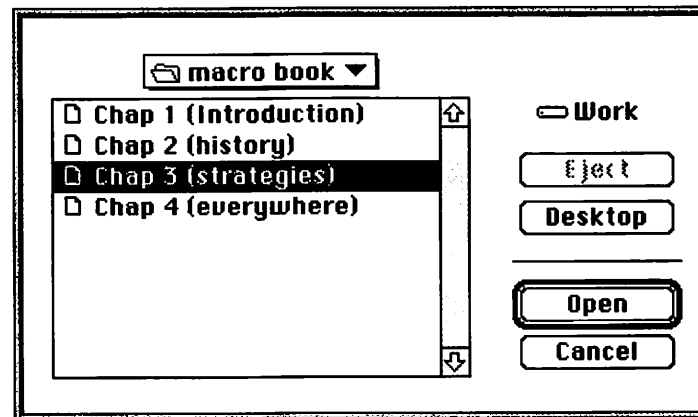


Figure 2-3: The File Open dialog box where you can select the document you want the File... macro to open.

QuickKeys' File dialog box appears with the filename at the top and the path to it in a drop-down menu (Fig. 2-4). You can change the

file that QuickKeys opens by clicking on the Change button. To assign a keystroke to the macro, press a key or combination of keys while the Keystroke box is highlighted.

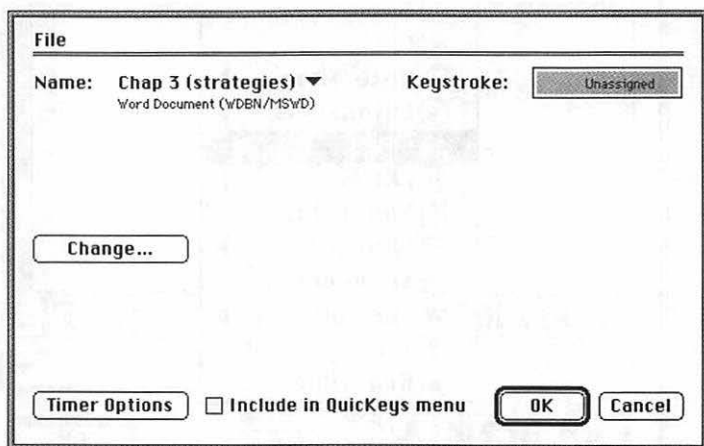


Figure 2-4: A File... macro. You can see the path to the file by clicking on the triangle next to the document's name.

4. Clicking OK saves this macro with your other macros (Fig. 2-5).

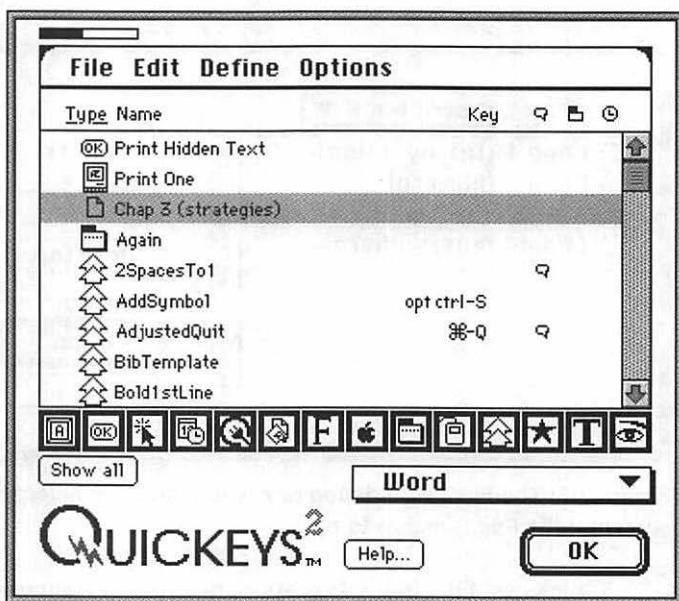


Figure 2-5: The File macro saved in Word's keyset.

Now that you've created your first macro, you can play it by pressing the keystroke you assigned to it. You can also have the macro run automatically by assigning a time to it. If you want it to run at a specific time after your Macintosh starts, click under the clock icon and next to the macro's name (Fig. 2-6) to open the Timer Options dialog box (Fig. 2-7).

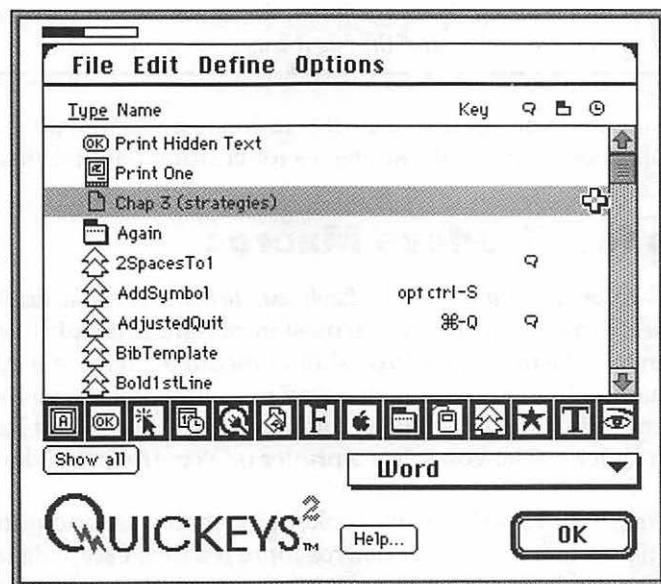


Figure 2-6: To set a time for the macro to run, move the cursor under the small clock icon in the QuicKeys editor window and click.

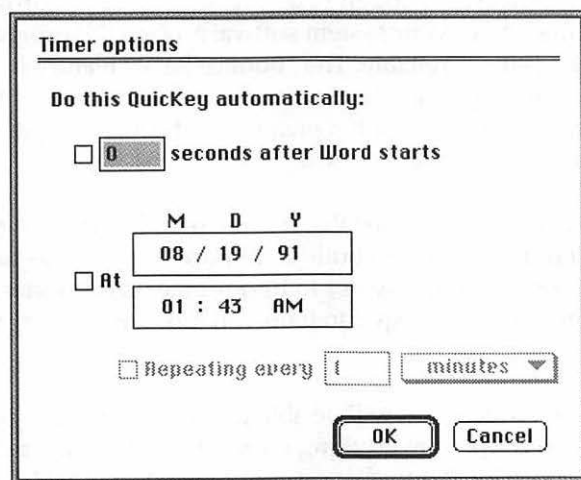


Figure 2-7: The Timer Options dialog box.

The number you enter in the seconds after your Macintosh starts box determines how long before Word will be launched and your document opened.

NOTE: *If you create a macro at the office and pass it on to someone else or take it to your home computer, it may not work if the computer isn't configured exactly like the one it was created on.*

Now that you see how easy it is to create a macro with QuickKeys, let's talk about some of the strategies for creating flawless macros.

Strategies for Flawless Macros

Think of the simplest, most reliable way to accomplish the task without relying on screen location. This is the most important principle of creating macros for the Macintosh, and you should meditate on it the way you would a mantra. If you can think of a way to circumvent screen location problems, use it. Like the File... command, the Choosy and Panels extensions in QuickKeys let you select a printer or a control panel device accurately.

You should also be on the lookout for shareware and public domain utilities that make your macros more reliable. Escapade (a freeware program included on the companion disk) is a prime example. It lets you "click" a button in a dialog box by entering a command key equivalent. The keystroke is more reliable than creating a macro that clicks the button's position on screen. You should also think about the commercial programs you own. Though you may have bought a program to add a certain function to your system software, often the program can make your macros more reliable. The "bounce back" feature in Boomerang, Super Boomerang, and Shortcut makes it easier to select the last file that was opened in a folder or the next file that could be opened in a standard Open dialog box.

Build complex macros as modules. If you are building a long, complex macro, it is a good idea to build it in individual chunks or modules. The main reason is that it is easier to find problems in shorter macros than in longer ones. This strategy can help you more quickly create working macros.

In some programs, you will be able to combine all your modules into one big macro when everything works. However, this practice can make a macro much harder to change if only one part of it "breaks."

Steal from existing macros. If you practice the previous rule, you should be able to steal from all of your macros. For instance, if you have a complicated macro that works, you might be able to steal part of it for the macro you are creating. Stealing it saves time since you aren't reinventing the wheel. Don't assume, however, that you won't have to test the new macro. Just because the chunk you extract works in another situation doesn't mean it will work in the new situation. For example, a macro that clicks on the buttons in a dialog box in one application may not work in a second application. The most common reason for this is that developers often add their own items to extend the Macintosh's basic dialog boxes. Both Microsoft Word and Aldus PageMaker have customized Page Setup and Print dialog boxes. If you create a macro that opens the Page Setup dialog box and clicks on the Wide orientation button in PageMaker, the macro will fail in Word because Word has an icon instead of a button.


Don't make all of your macros universal. QuicKeys is MultiFinder and System 7.0 aware, that is, it can switch keysets when you switch programs under MultiFinder or System 7. There are good reasons for this beyond the need for software compatibility. You generally create a macro for a specific program, so you want that macro saved within the keyset for that program. In QuicKeys, you just select the program's keyset from the pop-up menu. Another reason that QuicKeys keeps separate keysets for all of your programs is just a matter of housekeeping. The Macintosh has a finite set of key combinations, so you don't want ⌘-T reserved only for the Empty Trash command in the Finder. You may want to use that shortcut again in another program. You may want to use the shortcut mnemonically in other programs. For example, you could assign ⌘-T to select the Times font in MacDraw Pro and Word, but leave that key unassigned in PageMaker and Illustrator.












As you are creating a macro, you should give careful thought to whether it should be universal or program-specific. Any macro for window management should be universal so you can use it in every program. Macros that zoom, resize, close, or scroll a window are applicable across all programs. Even though a macro to enter "smart quotes" (the curly quotation marks around the previous phrase) whenever you type the double quote key (") is appropriate in all word processing programs, you do not want it to enter smart quotes in your terminal emulator or programming editor.

Any limitations imposed by the macro program also determine whether a macro should be universal or application specific. In our example above, the QuicKeys macro can launch the application and open the

document at a specified time if it is a universal macro. If it is a Word 4.0 specific macro, it runs a set period of time after Word starts up.

Reserve one key as the testing key. Whenever you are testing a new macro, you should temporarily assign it to a testing key. This appears to be a bit of compulsive behavior, but it has two distinct advantages. You never forget to which key you assigned the macro. You don't decide which keystroke is the macro's permanent home until after you have thoroughly tested it. You could also use that same key for macros you are not planning on using for a long time. Instead of thinking, "umm, which key should I assign this macro to?" just assign it to a key reserved for temporary macros. If you find that you really want to keep the macro, you can give it a permanent home later.

Assign command key shortcuts mnemonically. Assigning keyboard shortcuts based on function has been a practice since the Macintosh was introduced (try not to think too hard about using -Z for Undo). The keyboard shortcuts for Open, Save, Quit, Print, Find, and Go To are based upon the first character of the command and are largely consistent across programs. Whenever possible, you should mimic this convention so it will be easier for you to remember the keyboard shortcut for a macro.

When you assign a keystroke to a macro, you should remember that the best keyboard shortcuts are single characters on the alphanumeric keyboard combined with modifier keys (the Shift, Option, Control, and Command keys modify the behavior of the other keys on the keyboard). The only real problem is finding a combination that hasn't already been assigned by the developer. A number of the -character combinations have already been assigned in programs, and more and more programs are using  with the  and  keys. While you might not want to assign -T to a macro that selects the Times font in Word (Word already has assigned that keystroke to open the Define Style dialog box, and your keystroke would override Word's), you could still use --T. Also, don't use massively modified keyboard combinations for macros that you use regularly. Save those knuckle-breaking, ----F4 combinations for macros you won't use very often.

Group macros and assign them to keys according to their function. Like the previous tip, this one makes it easier to remember the keystroke for a macro. Command key grouping should be an obvious macro aid, but very few people or developers ever use them. When you have two or three macros that perform similar functions, assign them to similar

keystrokes. For instance, if you have a macro that pastes the contents of the clipboard directly into the scrapbook, you could assign it to **⌘-F4**. This assignment makes it easy to remember since **F4** is the function key equivalent to Command-V and your macro is a variation of this paste command. It also helps that **F4** is one of the few function key assignments that is almost universal across programs. If you have a standard keyboard, you might want to use **⌘-Option-V**. Now when you are trying to remember the keystroke for your macro, you have a connection that jogs your memory.

I've Run Out of Keys!

What do you do when you run out of keyboard shortcut combinations? If you are using one of the smaller keyboards—like Apple's Standard ADB keyboard or a Plus keyboard—the answer isn't an easy one. After you have used all of the possible **⌘**, **Shift**, **Option** and character key combinations on these keyboards, you can use those modifier keys with the keys on your numeric keypad and the arrow keys. QuickKeys assigns the numeric keypad keys separately from the number keys in the top row of the alphanumeric keyboard. The newer Apple Standard keyboards, while lacking the function and special keys (like Home and End), do have a **Ctrl** key. You can also consider using the QuickKeys' Quick Reference Card to play the macros you use the least. You shouldn't sell using the Quick Reference Card short either. If you don't use the macro often enough to remember the keystroke, why waste a perfectly good keystroke on it? You can also use the SoftKeys extension to create groups of keys and assign those to the 1 through 0 keys on your keyboard. The only other suggestion I can offer is to get a keyboard with more keys on it.

Besides the Apple Extended Keyboard II, DataDesk and Key Tronic sell keyboards with 15 function keys and 6 special keys (Help, Home, PageUp, PageDown, End, and Del).

With all the possible combinations of function keys and modifier keys on an extended keyboard, you will have 240 additional keystrokes to assign macros to. Since all of the macro programs allow you to have separate macro sets for each application, you have 240 additional keystrokes for each application.

If you are happy with your standard keyboard and don't mind spending more money, you could get the UnMouse touchpad from MicroTouch Systems. It is a 6 1/2 x 4-inch, touch-sensitive tablet that connects to the ADB port of your Macintosh. The UnMouse has a special microswitch



Figure 2-8: The Key Tronic extended keyboard. (Photo courtesy of Key Tronic.)

beneath its glass surface; pressing down on the surface activates the mouse and “clicks” the mouse button. You move the cursor by dragging your finger or a pencil with a conductive shaft (MicroTouch provides one) across the surface of the UnMouse. A slot in the top of the UnMouse allows you to slide templates or a drawing to trace beneath the glass surface.



Figure 2-9: The UnMouse tablet from MicroTouch Systems. (Photo courtesy of MicroTouch.)

While the UnMouse is marketed to replace your mouse and double as a drawing tablet, it can also function as a macro keypad. In the Power KeyPad mode, you can invoke a macro by touching one of the function key areas on the tablet. Templates supplied with the tablet show you where the function key areas are. You can set the UnMouse for either 16- or 60-key mode. In 16-key mode, the first 15 keys on the tablet correspond to the 15 function keys on the Macintosh extended keyboard, and the last key corresponds to the Return key. In 60-key mode, function keys **F1** through **F12** are mapped to the top three rows of keys on the tablet. The last row of keys corresponds to the modifier keys on the Macintosh keyboard: **Shift**, **Option**, **Command**, and **Ctrl**. To access function keys 13 through 60, you press one of the modifier keys on the tablet and then one of the twelve function keys in the first three rows. Seven sets of prerecorded MacroMaker macros and their templates are supplied with the UnMouse.

Another way you can play your macros (and increase the power of QuicKeys) is by using a mouse to play your macros. Unfortunately, you'll have to replace the mouse that came with your Macintosh. Advanced Gravis and Logitech sell mice with three buttons, and you can assign the buttons on the mice to press keystrokes. Of the two mice, the Logitech is the least flexible; you can only assign a single function to each mouse button and you will need to use one button as a single click. This only leaves you two buttons to assign keystrokes to.

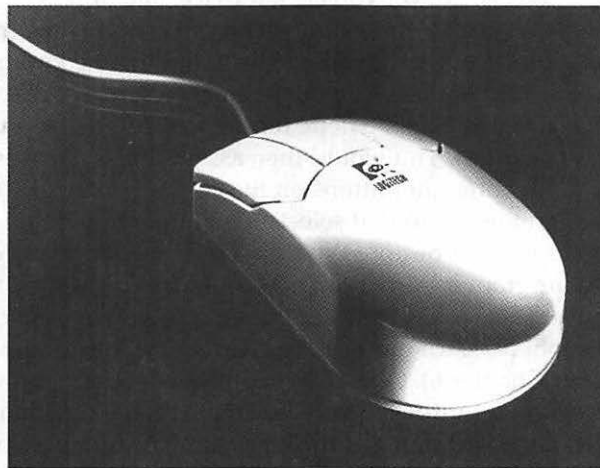


Figure 2-10: The Logitech three-button mouse. (Photos courtesy of Logitech.)

The Gravis mouse is the more functional of the two since you can assign three different functions to each mouse button. You activate the functions by double- and triple-clicking the button. The SetMaker program provided with the Gravis mouse lets you create specialized sets of button functions (Fig. 2-11).

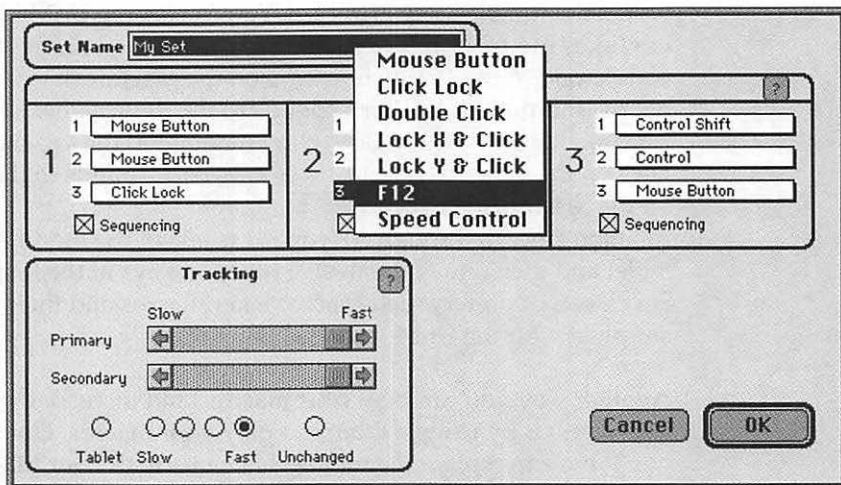


Figure 2-11: You can create a set of button functions for the Gravis mouse by selecting the function from the pop-up menus.

In a function set, for instance, you might set the left button of the mouse to perform a standard mouse click whenever you click it once, perform a click and hold whenever you click it twice, and “press” a keystroke whenever you click it three times. You can combine the mouse’s button keystroke function with QuickKeys by assigning the keystroke of the macro you want to play. You might create a series of macros that select the tools on Canvas’ tool palette (or the tool palette of your favorite graphics program). You would then assign the keystrokes of those macros to the different buttons on the Gravis mouse. For example, you could assign the macro that selects the Pointer tool to the triple-click function of the first button. Whenever you needed to select the Pointer tool in your graphics program, you could triple-click the first button on the mouse. This would be faster than mousing over to the palette to click the tool (to learn more about creating macros to select drawing tools, read Chapter 6). Obviously, you cannot assign as many macros to the three mouse buttons as you can to the fifteen function keys on an extended keyboard. But a graphic artist might find playing macros from the mouse more practical than reaching for a function key on the keyboard. Now let’s talk about how to organize all of the macros you are going to create.

Getting Organized

As you create more and more macros, you may discover it is harder to remember which macros you've made and what keystrokes they are assigned to. If you are using QuickKeys, you can always open the Quick Reference Card to check key assignments. You can also use the QuickKeys Template Printer to create a template for your keyboard with the macro names and keystrokes printed on it.

If you want something more permanent than paper templates, several companies sell templates that are made from heavy stock. DataDesk International sells templates that fit in a custom holder above the function keys of their Mac-101E keyboard. The templates come fifteen to a pack, and an erasable marker is included. Unfortunately, these templates really don't work on any of the other keyboards.

Multicomp of Abilene, TX has the most complete template product I have seen. Their F-Key Flipbook has five separate templates held together with two, clear plastic rings. Each template is made of heavy stock and covered with plastic so you can write your macro names with the red and blue permanent markers provided. An eraser is included to make changes easy. One side of the template has lines above each of the function keys and the Escape key, while the other side has boxes for the macro names, function keys, and modifier keys. One end of the templates is notched so you can write the program name on each template and flip quickly between them.

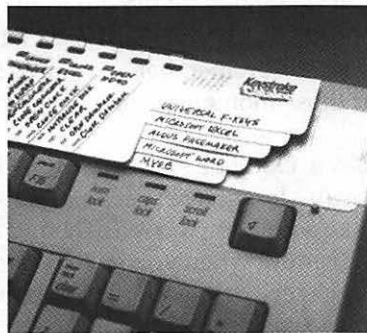


Figure 2-12: The F-Key Flipbook. (Photo courtesy of Multicomp.)

The F-Key Flipbook comes in three styles to fit the Apple Extended, Extended II, and Mac-101 keyboards. The base of the Flipbook is a piece of heavy plastic. Multicomp provides adhesive strips that let you attach it to most keyboards, but I've found the plastic base to be heavy enough to hold the Flipbook in place on the Key Tronic keyboards.

Let Macros Make You More Productive

Unfortunately, just knowing the strategies for creating flawless macros won't make you immediately more productive. You have to think about how you use your Macintosh and consider ways to let your Macintosh do even more work for you.

Be a macroing fool. If you make the same complicated menu selection or formatting change more than two times, you should turn it into a macro. For example, if you have to change the font, style, and size of all the headlines in your page layout, you should take the time to make a macro. At first, creating the macro seems to be a bother, but you can save yourself the time it takes to make three menu selections multiplied by the number of changes you have to make. It is a good idea to reserve a key on your keyboard as the temporary key. You can assign any temporary macros to this special key. If you use the macro on more than three separate occasions, you may want to give it a permanent activation key.

Add a few macros at a time. Even though it is a good idea to make macros of all complicated tasks, you shouldn't try to simplify your life in one afternoon. Your brain won't learn all of the macros at once. Add a macro, learn it, use it, and then add another. This approach has the additional benefit of making it easier to test your macros.

Label your macros. It is always a good idea to label your macros. By the end of your vacation, you may forget the keystrokes for macros you use every day. Your brain cannot remember all the different macros you make. If you have the function keys on your keyboard labeled, you can glance at the labels when you can't remember a keystroke. QuickKeys has a built-in reference card for those moments when memory fails you.

Explore for ten minutes every day. Don't be satisfied with what you already know. When you really don't have time to make a macro of a complicated procedure, make a note to yourself. Then you can try creating a macro for that procedure when you have a few free minutes. If you try out something new every day, you will learn more about your Macintosh and the tools you are using.

Teach someone else what you learn. Any teacher will tell you that she learns more and remembers more of what she has learned by passing on the knowledge. When you've learned a nifty trick or created a dazzling macro, teach it to someone else. Teaching will not only help reinforce what you've learned, but in a short time, your co-workers will be more productive.

Don't wait on a macro. If your macro plays for a long time, don't wait for it to finish. Once you have tested and used a macro a couple of times, it should run without your intervention. Take the extra time to clean off your desk or get a cup of coffee. When your Macintosh is saving you time and effort, don't waste it by being idle.

Let's Get on with It

Now that I have pontificated about the principles of creating, using, and remembering macros, you're probably bored senseless. But, this is important information. Honest. You may even want to refer back to this chapter occasionally (I certainly will). If you want to know more about the hardware and macro aids I've mentioned above, you can refer to Appendix A, "Where Do I Find This Stuff?"

3

CHAPTER

Macros for Everywhere

In most of this book, each chapter deals with creating and using macros for a specific kind of software. The reason for this is simple. Macros for formatting text in your word processor aren't usually appropriate for your spreadsheet. However, some types of macros apply to all kinds of software. A macro to transpose characters is just as useful in a spreadsheet or database program as it is in your word processor. This chapter shows you how macros can standardize the shortcuts in all your programs, make your programs smarter, and automate your work. I have tried to create macros for the most popular programs, but I realize that not everyone uses the same programs. In particular, I've created macros for PageMaker, Microsoft Word, MORE, TeachText, and DiskTop. I try to suggest comparable programs in case you are not using the same program as I am. In most cases, I have created the same or similar macros for those other programs; you will find them on the companion disk. However, if you do not have one of the programs I suggest, and I have not included a macro for the program you use, you should be able to use the techniques I describe to create the macro for your favorite program.

Standardizing Your Programs' Shortcuts

To ensure that all of the programs created for the Macintosh had a consistent user interface, Apple put the basis of the interface—its windows and menus—in its read-only memory (ROMs) chips.* Jef Raskin, who originally conceived of the Macintosh and wrote the Macintosh Papers, said that he “wanted to build the user interface into the [Macintosh] to trick programmers into using the interface. They

*Guglielmo, C., The soul of a not-so-new machine, *MacWEEK*, January 7, 1989.

might not like it, but they would at least build a prototype of the software because the interface already there was easier.” This forced programmers to make their programs conform to a common interface. And, even though Apple standardized some keyboard shortcuts (like **⌘-C** for Copy and **⌘-X** for Cut) in its first programs, it didn’t dictate every keyboard shortcut for the different commands. As a result, some default shortcuts have grown out of common usage.

Microsoft Word was one of the early programs—if not the first—to use **⌘-W** as the keyboard shortcut to close the frontmost window. When Living Videotext released MORE, it followed the same convention. Now you have to search to find a program that doesn’t use **⌘-W** to close the frontmost window.* Unfortunately, there will always be some program where **⌘-W** does something different—like wrapping text to fit the margins or, as with PageMaker, reducing the document to fit in the window. Wouldn’t it be so much simpler if all of the programs you used on a regular basis had the same or a similar set of keyboard commands? You wouldn’t have to remember which programs require you to type **⌘-W** to close a window and which ones don’t. Besides letting you add keyboard shortcuts to menu items that don’t have any, the Menu/DA... QuickKey makes it possible for you to standardize the keyboard shortcuts across all your programs.

Menu Macros

Standardizing the keyboard shortcuts in your programs may seem like a trivial use of QuickKeys. After all, you aren’t creating complex sequences that open several programs and swap information between them. Nevertheless, giving all of your programs the same—or nearly the same—set of command key equivalents is very powerful. It saves you the time and brain space necessary to learn and remember a different set of keyboard shortcuts for all the programs you use regularly. A good example of this is the Close shortcut we mentioned above. For example, you could create a macro that closes your PageMaker document whenever you press **⌘-W**. Because QuickKeys catches keyboard events before they get to the program, your shortcut is invoked instead of the shortcut the programmer built into the program.

*A quick survey of my hard disk turns up Vantage 1.5 (**⌘-period**), Think Pascal and Think C 4.02 (no equivalent—although the new versions of both programs use **⌘-W**), PageMaker 4.01 (no equivalent), FullWrite 1.5s (no equivalent), FreeTerm 3.0 (no equivalent), and Canvas 2.1 (no equivalent—though version 3.0 uses **⌘-W**).

A Menu QuicKey

Before we begin: You should already have QuicKeys Demo or your regular copy of QuicKeys installed and the program for which you want to create the macro running. In this example, we are running PageMaker (any version). If you don't use PageMaker, you may use another program with a non-standard keyboard shortcut. You can create this macro for that program.

Create the macro:

1. Select QuicKeys... from the Apple menu.
2. When QuicKeys opens, select PageMaker from the pop-up menu at the bottom of the window (Fig. 3-1). This causes the QuicKey that you create to be saved as a program-specific QuicKey (PageMaker, in this case) and not a Universal QuicKey (we'll talk more about this decision in a moment).

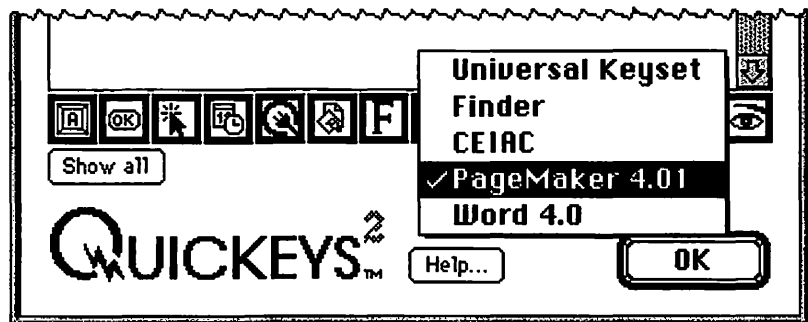


Figure 3-1: The program pop-up menu in the QuicKeys window.

3. Select Menu/DA... from QuicKeys' Define menu. You will notice that all of PageMaker's menus are activated, the cursor becomes a miniature menu, and a dialog box appears prompting you to select an item from the menu (Fig. 3-2).

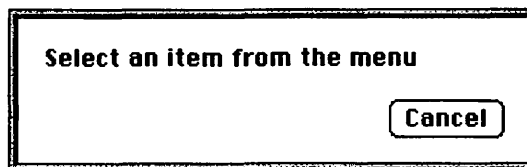


Figure 3-2: QuicKeys prompts you to select a menu item.

4. Select Close from PageMaker's File menu. The prompt dialog box is replaced with a Menu editor containing the various settings for your QuickKey (Fig. 3-3).

TERMS: Each of the dialog boxes associated with a command in QuickKeys' Define menu or the Date/Time, Extensions, FKeys, Mousies, and Specials hierarchical menus are called editors. They contain all of the settings available for the QuickKey you have created and are the point for tweaking macros to make them perform better.

Menu

Select from menu Keystroke: **Unsigned**

☒ by Text: ☒ Match exactly

☐ by Position:

☒ Look for menu by title:

☐ Search all menus

☐ Only ⌘ menu

While selecting from menu, hold down:

☐ ⌘ ☐ Shift ☐ Option ☐ Control

☐ Don't complain if the menu choice can't be found

Timer Options ☐ Include in QuickKeys menu **OK** **Cancel**

Figure 3-3: The Menu editor for the Close QuickKey.

Notice that QuickKeys has automatically selected the by Text radio button and the Match exactly checkbox and has entered the command name, its numerical position in the menu, and the menu name in the text boxes of the dialog box. The selected settings mean that QuickKeys looks for the command by name in the File menu. If QuickKeys cannot match the command exactly (that is, if it cannot find a Close command in the File menu), it warns you. When you are first creating macros, you may want QuickKeys to warn you whenever it cannot find a menu item. This will help you refine your macros.

NOTE: The commands and menu items in some programs change to reflect changed conditions. For example, the Again command in Word's Edit menu changes to Edit Again or Format Again to reflect your editing or formatting actions. If you create a Menu QuickKey for the Again command and leave the Match exactly checkbox selected, QuickKeys wouldn't be able to find the command. Generally, it is a good idea to leave this option checked so your macro doesn't do something unanticipated.

5. Press **⌘-W** to insert your keyboard shortcut in the keystroke box and click the OK button. The Menu dialog box closes, and you can see that your macro has been added to the PageMaker QuickKeys list (Fig. 3-4).

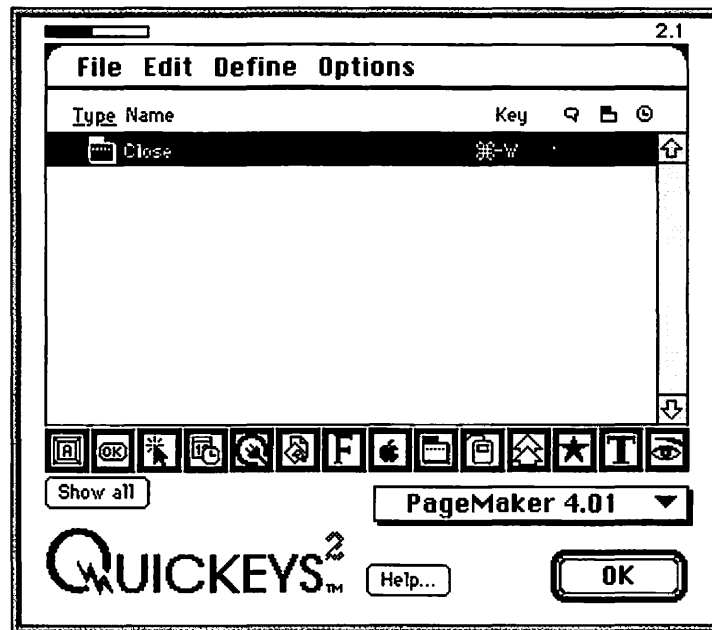


Figure 3-4: The Close QuickKey added to the QuickKeys list.

Now, whenever you type **⌘-W** in PageMaker, the document window closes. If you have used QuickKeys before, you may wonder why we didn't choose Close Window from the Mousie hierarchical menu (Fig. 3-5). The Close Window mousie closes the frontmost window. Using it to close the document window would actually close PageMaker's

Toolbox and Styles windows (depending on which ones you have open) before it closed the document window. Selecting the Close command from PageMaker's File menu ensures that the document window is closed every time.

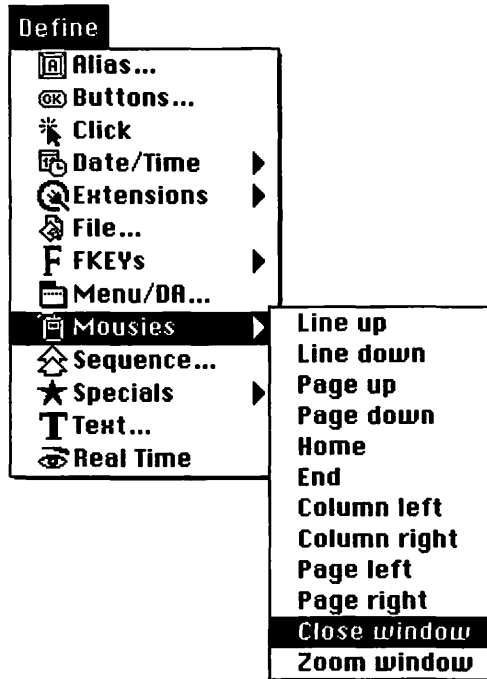


Figure 3-5: The Mousies hierarchical menu.

I should also explain why in step 2 we chose to make this a program-specific QuicKey instead of a Universal QuicKey. The commands in some programs do not work correctly if you create a QuicKey using the same keystroke for the command as the program does. For instance, the Quit commands in 1st Aid Kit and HyperCard 2 do not work if you have created a Quit QuicKey using the Menu/DA... command and assigned the keyboard shortcut **⌘-Q**. To avoid this potential problem, it is better to make macros program-specific whenever possible. Only make a macro universal if you think you might use it in every program. If you experience a conflict and need to create the same macro for several different programs, you should create it once and then copy it to the other programs.

I should also address the issue of making a macro for a menu item instead of modifying the resources in the program with ResEdit or a

similar resource-editing program to add a keyboard shortcut. Some Mac purists might say that it is better to change the menu shortcuts permanently with ResEdit. ResEdit is a very useful tool, but using QuickKeys to create keyboard shortcuts for a program has some distinct advantages.

If others use your Macintosh, a program with non-standard keyboard shortcuts could prove to be very confusing. Turning off the macro program or switching macro sets “reverts” programs to their original state. This approach is better for Macintoshes with several users. Another argument for creating macros is that you won’t have to modify the program again. If you crash or delete your modified program, you will have to modify the original program again or reinstall a modified version of the program from a backup. When you create macros for a program, you still have them no matter what happens to the program.

Perhaps the most persuasive argument for creating macros to modify a program’s interface is that you cannot always use ResEdit. You cannot, for instance, change the keyboard shortcuts in PageMaker using ResEdit. Unlike most programs, PageMaker doesn’t have a menu resource. System 7.0’s Finder also has a non-standard menu resource. When System 7.0 first shipped, it was difficult to change the command key equivalents in the Finder. It was several weeks before someone figured out how to add shortcuts or change the current ones and created a ResEdit template to make the change easier for ResEdit-phobics. Macro programs allow you to modify the user interface of a program without fooling around in ResEdit.

Text Editing Macros

Another way to standardize a program’s interface is to give it the same keyboard shortcuts for moving the insertion point. Most programs allow you to hold down the **[Option]** or **[⌘]** key and press an arrow key to move the insertion point a word to the right or left. Unfortunately, these shortcuts are not always consistent.




I’m a reluctant Microsoft Word user. I started using Word in 1986 fresh from my DOS experiences with WordStar. As a touch typist, I had learned the WordStar “keyboard diamond”—the keyboard shortcuts for moving the cursor within the text. Quite naturally, I learned Word’s shortcuts for moving the cursor, a mixture of the **[⌘]**, **[Option]**, and character keys. As other word processors came along, I thought about changing, but each one lacked the same set of keyboard shortcuts for moving the cursor. FullWrite Professional has a very similar set of shortcuts, but its differences can prove infuriating. Like most users, I

didn't want to learn a new set of shortcuts. Trapped by shortcut inertia, I've hung on to Word.*

Macros can save you from the drudgery of memorizing a different set of text-editing keyboard shortcuts for each program. I'm using Word's shortcuts in this example; not because you would want to memorize such an awkward set of commands, but because it represents a worst case scenario. I must admit that Microsoft Word has a wonderful interface for changing its own keyboard shortcuts, and some could argue that I'm doing this backwards—I should be changing Word's commands to match those found in other programs. But let's remember that, for good or ill, I learned Word first. No matter how irrational your choice for a favorite program or collection of keyboard shortcuts, QuickKeys is the great equalizer: It lets you make almost any program mimic your favorite one.

Using Alias QuickKeys

The key to making one program mimic another is the Alias QuickKey. It allows you to set an alias keystroke and the keystroke that the program receives whenever you press the alias. If you select the Alias command from the Sequence editor's Define menu, the Literal editor opens.

Before we begin: As in the previous example, you should already have QuickKeys or the QuickKeys Demo installed and the program for which you want to create the macro running. The program must have some commands for moving the cursor. In most word processors and text editors, for instance, the  key moves the cursor a character to the right, and the - keys move the cursor a word to the right. I'll use MORE 3.1 because it has commands for moving the cursor a character and a word at a time. If you are creating the macro with me, open QuickKeys and select the program you are using from the pop-up menu.

Creating the macros:

1. If you still have QuickKeys open, select Alias... from the Define menu.

*While I was working on this book, Microsoft released Word 5.0. Ironically, Microsoft has changed the keyboard shortcuts by which I've judged all other word processors. After 15 minutes of using the new version, I opened Word 5.0's Commands dialog box and changed the keyboard shortcuts to match Word 4.0.

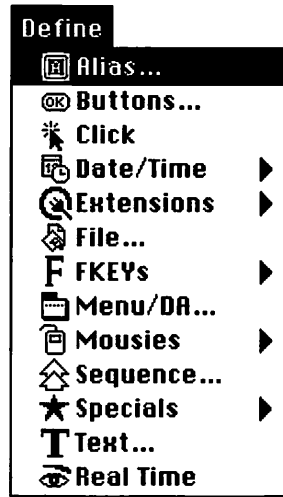


Figure 3-6: The QuicKeys Define menu.

2. When the Alias editor opens, the Key to type box is highlighted. Press the key.
3. Press to move to the Keystroke box and press --L. This is Word's shortcut for moving the cursor one character to the right (Fig. 3-7).

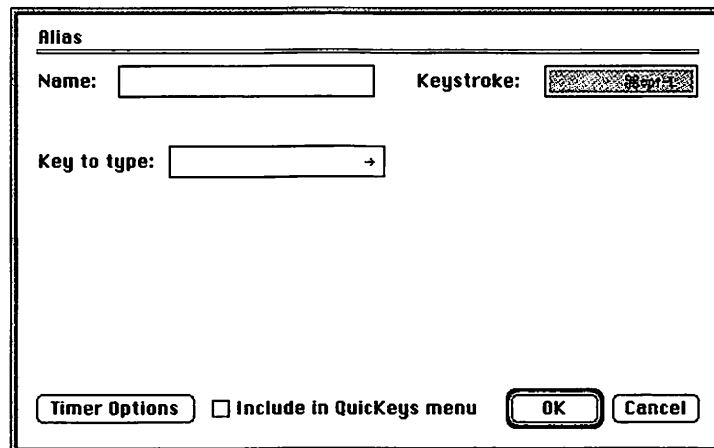


Figure 3-7: The Alias editor.

4. Press to close the Alias editor and save your changes.

Variations: In addition to moving the insertion point, you can create Alias QuickKeys that allow you to select text. In most programs, you can hold down the **[Shift]** key to select text while you are moving the cursor with a keyboard shortcut. In MORE, for instance, you hold down the **[⌘]-[Shift]** key combination and press the **[→]** key. You'll notice that the MORE keyset on the companion disk contains aliases to move the cursor a character or a word in every direction, move a page up or down, and select text while moving. You will also find sets for Vantage, MacWrite II, and Nisus on the disk.

When you are modifying these macros for use with another program, some of the assigned keystrokes may overlap a shortcut in the program. To quickly check for this, turn off QuickKeys with the Toggle QuickKeys On/Off special and then try the keystroke. You can start a keystroke domino effect if changing one keystroke forces you to change a second and then a third. For example, the command to move the insertion point one line up in Word (**[⌘]-[Option]-O**) selects the Open Any command in FullWrite Pro. QuickKeys catches your keystrokes before FullWrite Pro does, so the insertion point moves up and the dialog box doesn't open. You can create a Menu QuickKey for the Open Any command or, if the command isn't one you use frequently, you can always toggle QuickKeys off to invoke the command and then toggle QuickKeys back on. If you really felt it necessary, you could even reach for your mouse.

Replacing Your Mouse

In part, the basic assumption of creating keyboard shortcuts is that you would rather type on the keyboard than reach for a mouse. Some of us are inveterate keyboarders and prefer to reach for the mouse only as a last resort. If you hate your mouse, you need not worry that you have latent PC tendencies. A mouse is a somewhat better pointing tool than, say, a brick with a pencil tied to it, but it isn't always the quickest or the most accurate way to accomplish a task. And it is even better to let the Macintosh do the mouse-work for you since it is always quicker and generally more accurate.

I found someone with the same attitude toward mice when I interviewed Don Brown of CE Software for a *MacWEEK* article in 1987. He described the appearance at Computer Emporium of an Apple Lisa. He had nothing but disdain for the computer and its rodent. As he put it, "If God had meant for us to use a mouse to edit text, he wouldn't have created arrow keys." While I may not agree with the letter of Don's assertion (and I can't fault his logic), I can certainly agree with its spirit. If you can keep your hands on the primary or alphanumeric keypad of

your keyboard, you'll be a faster and more accurate typist. These are the most important reasons for creating keyboard shortcuts and macros whether or not you are a touch typist.

Managing Windows

QuicKeys has several built-in window management Specials and Mousies QuicKeys. The Select rear window and Select second window specials allow you to switch between windows in an application. The Close and Zoom mousies let you close or zoom the frontmost window.

You may ask "why would someone want to use a macro instead of the mouse to move a window?" Pressing **[Esc]** is always faster than reaching for the mouse, mousing over to the close box of the window, and then clicking it. And you also have to consider both kinds of desktop areas: on screen and on your actual desk. A full-page or two-page display may require some serious mousing before you get from the bottom of the screen to the window's close box. Add to this the difficulty of finding enough uncluttered space on your desk to correspond to the size of the screen.

While QuicKeys' four specials and mousies are very useful, you may discover that you have more specific window management needs. For instance, you can use macros to move and resize windows without having to lift your hand for the mouse. One trick that has yet to be built into every Macintosh program is hiding windows when the screen becomes too cluttered. StuffIt Deluxe, Exposure, and HyperCard all have keyboard commands that move their windoids just off screen to get them out of the way.

Hide Window Macro

Before we begin: You should be in the program you want to use. In this example, we are in the Finder with a disk window open. Open QuicKeys and select Universal from the pop-up menu. We are making this a universal macro so we can use it in any program. One word of warning is in order: This macro doesn't work properly with the windows and windoids in some programs. For example, this macro does not drag any of PageMaker's windows off the screen because the Style, Toolbox, and other windows interfere.

Create the macro:

1. Select Record one QuicKey from the QuicKeys menu.

2. Click the title bar of a window and drag it to the bottom of the screen so that only the title bar is visible and the rest of the window is off the screen. When you release the mouse button, the Click editor opens (Fig. 3-8). It shows you the screen location, window, and control area that you clicked within.

NOTE: You should click near the Close box. Doing so ensures that the click works with windows of all sizes.

Click

Name: Keystroke:

Click: From:(63,-15) from top-left corner
To:(22,715) from current location

Window: Window #1 from front

Control area: None

Click time(s)

Hold down: ☐ Command ☐ Shift ☐ Option ☐ Control

Timer Options ☐ Include in QuickKeys menu

Figure 3-8: The Click editor.

3. Click the Click button to open the Click Location editor (Fig. 3-9). It allows you to set the location of your click relative to the screen, document, or mouse position. The numbers that appear in the Mouse position boxes differ from one size of monitor to the next.
4. For the Click Relative to option, select the upper-left radio button in the small document window. Clicking relative to the document ensures that no matter where the disk or document window is on screen, QuickKeys always clicks in its title bar. Selecting the upper-left button causes QuickKeys to position the click from that corner of the document.

For the Drag Relative to option, select the Screen radio button. Dragging relative to the screen ensures that QuickKeys always drags the disk or document window to the same position on screen.

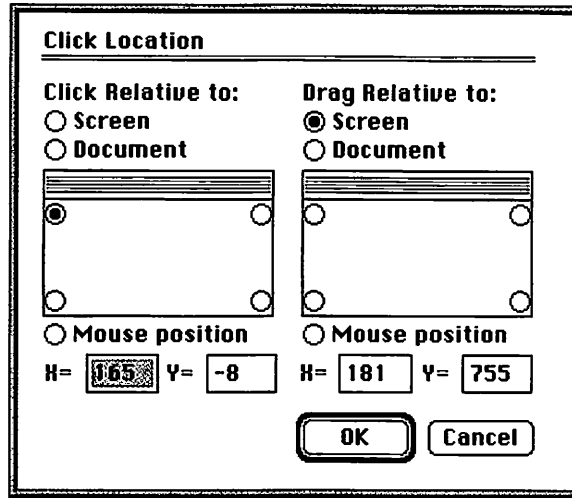


Figure 3-9: The Click Location editor for the Click QuickKey.

- Click OK or press **Return** to close the Click Location editor and save your changes. Name your macro something descriptive because it is hard to tell what the Click does by looking at the settings and then assign a keystroke (Fig. 3-10).

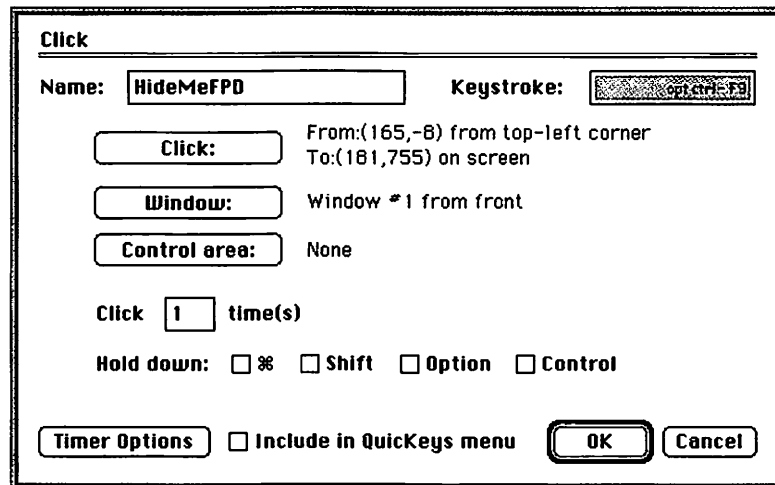


Figure 3-10: The Click editor for the HideMeFPD QuickKey.

- Click OK to close the Click editor and save your changes.

NOTE: You will find HideMe macros on the companion disk for 12-inch, 16-inch, and full-page monitors. Be sure to pick the one that corresponds to your screen. If you try to drag your window off screen by using the HideMeFPD macro on a 12-inch monitor, it will appear that nothing happens.

TERMS: You'll notice here and throughout the book that I tend to run the words in a macro name together. Since QuickKeys only allows 15 characters in a macro name, I make a conscious effort to get as much of the name in the Name text box as possible.

Since we've created a macro that hides the window at the bottom of the screen, it only makes sense to create a macro that moves it back to the home position.

Drag Window Home Macro

Some programs have a command to move the front window back to the home position, but the macro we are about to create works consistently in all programs. And there is a hidden benefit in the drag home macro: It will bring the window back to the home position even when it is completely off screen—that is, when you can't see the window at all. If you work on differently sized monitors and tend to switch those monitors around, you will eventually wind up with a window that is hidden off the main screen.

Before we begin: You should have a window open somewhere on your screen. You could even use the previous macro to move your front window just off screen. We will record this macro in the Finder, but you should open QuickKeys and select Universal from the pop-up menu so this macro will be available in any program.

Create the macro:

1. Select Record one QuickKey from the QuickKeys menu.
2. Click the title bar of the window and drag the window to the top of your screen. When you release the mouse button, the QuickKeys Click Editor opens. Notice that QuickKeys has already selected a screen position and a window for the click.

- Click the Click button to open the Click Location editor (Fig. 3-11). The editor allows you to set the location of your click relative to the screen, document, or mouse position.

NOTE: *QuickKeys enters the mouse location of clicks in pixels below the two document windows. X is the number of pixels the window is from the left side of the screen; Y is the number of pixels the window is from the top of the screen. As you create Click QuickKeys, you will notice that the pixel coordinates change, but QuickKeys ignores the pixel coordinates until you select the Mouse position button. Only use the Mouse position button when you want the click to be measured from the location of the cursor.*

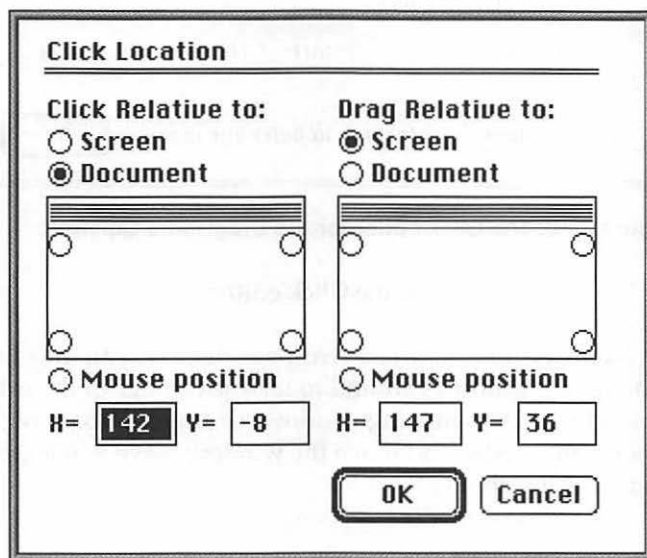


Figure 3-11: The Click Location editor for the Click QuickKey.

- For the Click Relative to option, select the upper-left radio button in the small document window. This setting ensures that QuickKeys clicks the title bar of the frontmost window no matter where it is positioned on or off screen. Selecting the upper-left button causes QuickKeys to position the click from that corner of the document.
- For the Drag Relative to option, select the Screen radio button. This setting ensures that QuickKeys drags the window back to the top of your screen.

- Click OK or press **Return** to close the Click Location editor and save your changes. Now name the QuickKey and assign a keystroke (Fig. 3-12).

The screenshot shows a dialog box titled "Click". It has two main sections. The top section contains a "Name:" field with the text "DragHome" and a "Keystroke:" field with the text "ctrl-F9". Below these are three rows of controls: "Click:" with a button, "Window:" with a button, and "Control area:" with a button. To the right of these buttons are text labels: "From:(142,-8) from top-left corner To:(147,36) on screen", "Window #1 from front", and "None". Below these is a "Click" field with the number "1" and the text "time(s)". Underneath that are four checkboxes labeled "Hold down:" followed by "⌘", "Shift", "Option", and "Control". At the bottom, there is a "Timer Options" button, a checkbox labeled "Include in QuickKeys menu", and two buttons labeled "OK" and "Cancel".

Figure 3-12: The Click editor for the DragHome QuickKey.

- Click OK to close the Click editor.

If you have a full-page or a two-page display, you may find yourself moving the windows around to take advantage of the extra screen space. I'm always moving windows so I can compare two or more files. Macros that resize and move file windows save you a great deal of time and mousing about.

Drag to the Right

When you are working in a word processor or text editor on a two-page monitor, you may have a great deal of wasted space on the right-hand side of the screen. A macro that moves a window to the right side of the screen would let you use that additional area for comparing or copying and pasting between two files.

Before we begin: You should have a program running and a window open. I'm using Word 5.0 for this example. Open QuickKeys and select Universal from the pop-up menu. You'll want to make this macro universal so it will be available in every program.

Create the macro:

1. Select Record one QuickKey from the QuickKeys menu.
2. Click on the title of the window (near the Close box) and drag it to the right half of the screen.
3. When you release the mouse, the QuickKeys Click editor opens. Open the Click Location editor (Fig. 3-13).

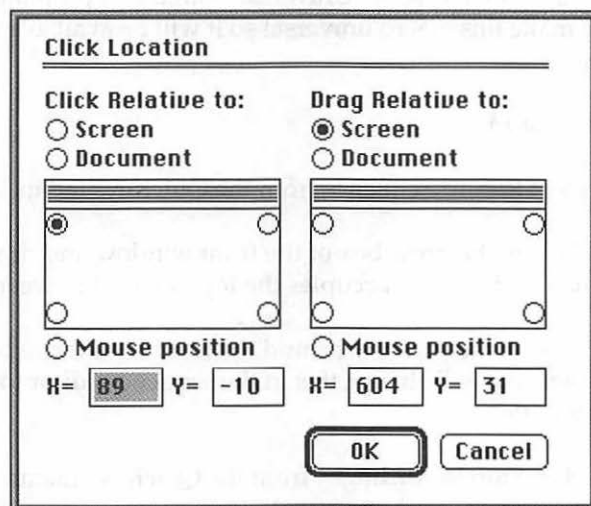


Figure 3-13: The Click Location editor for the DragRight macro.

4. For the Click Relative to option, select the upper-left radio button in the small document window. The radio buttons inside the document window let you set the corner of the window from which the click is measured. These settings are useful whenever the shape and size of the document window can change.
5. For the Drag Relative to option, select the Screen radio button. This setting ensures that the QuickKey drags the window back to the same location on your screen every time.
6. Click OK or press **Return** to close the Click Location editor and save your changes. Now name the QuickKey and assign a keystroke.
7. Click OK to close the Click editor and save your changes.

Tiling Windows

Another way to efficiently use the extra space on your monitor is to tile the windows. On a full-page display, for instance, you could place one document window at the top of the screen and a second at the bottom of the screen. Though some programs have a built-in tiling feature, making macros of your own ensures consistency across programs.

Before we begin: You should have a program running and two windows open. I'm using Word 5.0 since it doesn't have a built-in tiling feature. Open QuicKeys and select Universal from the pop-up menu. You'll want to make this macro universal so it will be available in every program.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Click on the grow box of the front window and drag it up until the window only occupies the top half of the screen.
3. Click on the window behind the front window to bring it to the front. We will change this in the sequence editor to make it more accurate.
4. Select Stop Recording... from the QuicKeys menu. We could resize the second window, but we can save some time by using the QuicKey we created in step 2.
5. When the Sequence editor opens, open the second Click QuicKey and then click the Click button to open the Click Location editor (Fig. 3-14). Select the lower-left radio button in the small document window under the Drag Relative to option. This causes QuicKeys to Click in the grow box no matter what size the window is. Close the Click Location and Click editors.
6. Now select the first Click and select Copy from the Edit menu. Since the insertion arrow (which appears beside the sequence listing) is at the end of the sequence, select Paste from the Edit menu. Copying and pasting the click that resizes the window ensures that the second window is exactly the same size as the first.
7. Select the second Click and select Cut from the Edit menu. Move the insertion point between the two Clicks and select the Select second window command from the Special hierarchical menu.

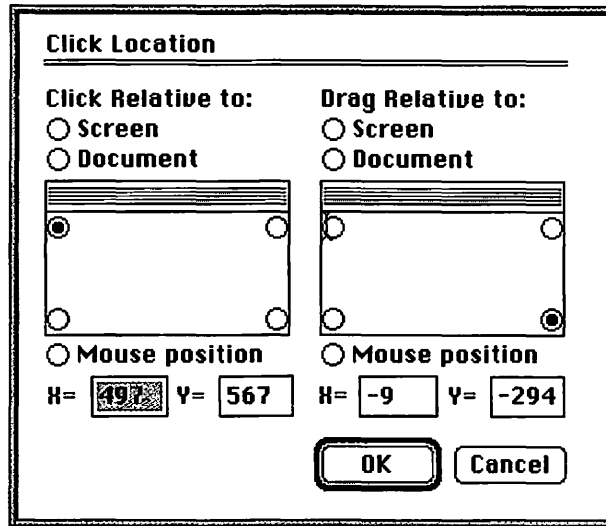


Figure 3-14: The Click Location editor for the Tile2Windows16 sequence.

Since this QuickKey does not depend on the location of the second window, it makes the sequence more accurate. Click OK to close the Special editor.

8. We still need the second window moved to the bottom of the screen, so let's add one more click and drag to the sequence. Move the insertion arrow to the bottom of the sequence.
9. Click the Record More button. QuickKeys closes, and the flashing microphone appears in the Apple menu.
10. Click on the title of the front window and position it below the back window.
11. Select Stop Recording. When the Sequence editor opens, name the QuickKey and assign a keystroke.

TIP: When you are playing a sequence, you can interrupt it anytime by clicking the mouse button or typing **⌘-.** (period). This will display a dialog box where you can choose to resume, pause, or cancel the macro (Fig. 3-15).

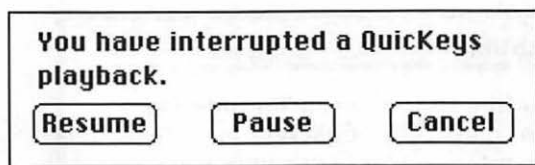


Figure 3-15: The interrupt dialog appears whenever you interrupt a macro during playback.

Variations: Even though this macro was originally created for use on larger monitors, it can be easily adapted to work on the 9- and 12-inch monitors. One variation on this macro would be to shrink a number of windows to their smallest size and tile them evenly on your screen. This would create more free space on your monitor, and you could use the Zoom mouseie to return the windows to their previous sizes.

While the Select rear window and Select second window specials let you switch between windows, they won't always switch between two windows when you have several windows open. If you have tiled your windows, you can create single clicks for each half of a full-page display. Then you can jump between the two with single clicks. This ClickUpLeft macro (Fig. 3-16) selects the window in the upper-left quadrant of a two-page display and has a ClickUpRight companion with a similar keystroke.

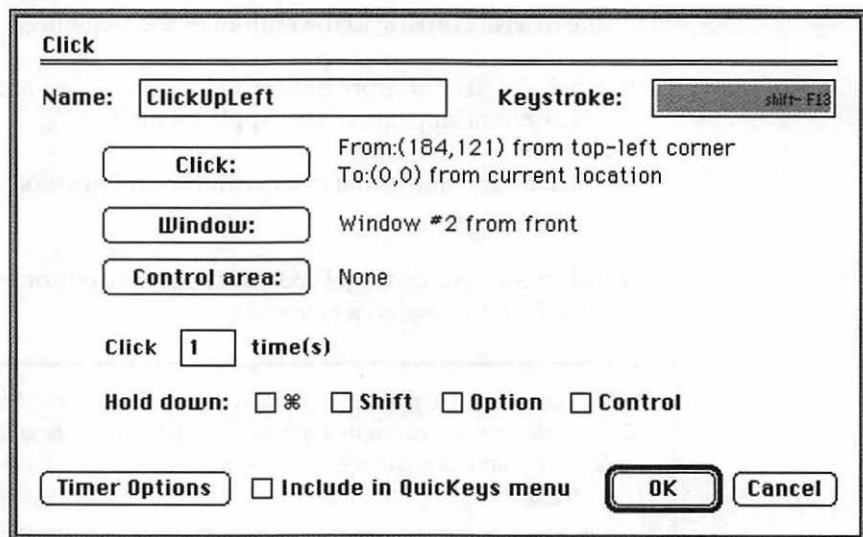


Figure 3-16: The Click editor for the ClickUpLeft macro.

Instant Screen Saver

Screen savers have become a very popular item for Macintoshes. After a set period of time, or whenever you move your cursor into a Sleep Now corner, the screen is blanked or replaced with an animated display. This prevents the desktop and menus from being burned into the phosphors of your Macintosh screen. Some public domain screen savers can be invoked with a single keystroke so your screen is immediately blanked. This is a desirable feature since your Macintosh screen doesn't have a chance to burn in for even a short period of time. While Pyro comes with an FKey that starts it immediately, this feature hasn't found its way into After Dark.

You can create a macro to immediately blank the screen whenever you leave your Macintosh. Let's define a Click QuicKey to move the pointer to the Sleep Now corner you defined in your screen saver's control panel.

Before we begin: If you haven't already defined a Sleep Now corner in your screen-blanking program, you will need to do that. Open your screen blanker and position the plus icon in one of the corners (Fig. 3-17). I'll use After Dark for this example. When you have specified the corner, you are ready to create the macro.

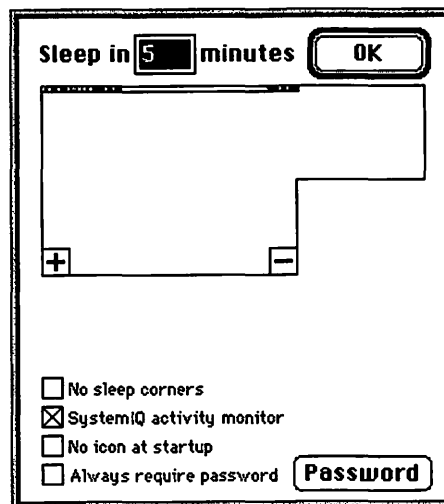


Figure 3-17: The After Dark control panel. The plus in the lower-left corner corresponds to the lower-left corner of your monitor and indicates the Sleep Now corner.

Create the macro:

1. Select Record one QuicKey from the QuicKeys menu.
2. Move the pointer and click in the Sleep Now corner. The QuicKeys' Click editor opens (Fig. 3-18).

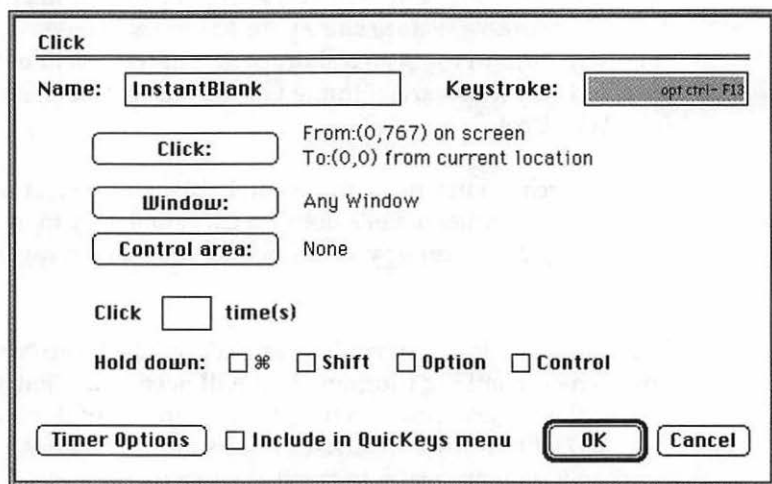


Figure 3-18: The Click editor for the InstantBlank QuicKey.

3. Set the Click x time(s) to 0. This moves the pointer into the corner without clicking and keeps the cursor in that position. If QuicKeys clicked in the corner, the cursor would move back to its original position and prevent the screen from blanking immediately.
4. Name the QuicKey something appropriate, assign a keystroke to it, then click OK to save it.

Now when you leave your Macintosh for an extended period of time, you can press the keyboard shortcut for the InstantBlank QuicKey, and After Dark immediately blanks the screen. This is especially helpful for those times when you have accidentally moved the cursor into the Never Sleep corner.

Making Programs Smarter

No matter how feature-rich the program, we have all probably had the thought "Why didn't the programmer add an option that lets me set the time to turn on the coffee pot?" While you can't add all of the features

that didn't make it into a program, you can make your programs smarter with macros. Best of all, the features you add with macros are the ones you need and that function the way you want. Some of the macros in this section add features I have longed for in the programs I use everyday, and some of the macros just make programs accommodate my quirky work habits.

Archiving Cuts

The Undo command in most Macintosh programs is only one level deep: Cutting or copying something replaces your last cut. Only a few programs let you append your cuts to the Clipboard. FullWrite Professional has an append to Clipboard command, and Nisus and QUED/M have multiple Clipboards. MultiClip also gives every program multiple Clipboards—though at the expense of some of your Macintosh's memory. If you don't want to send what you've just removed from your document into bit oblivion, you can create a macro that archives your cuts in the Scrapbook.

Before we begin: You should have the program for which you want to archive your cuts open. I'll use Word 5.0. Open QuicKeys and select Microsoft Word from the pop-up menu.

Create the macro:

1. Select something in your file to cut.
2. Select the Record Sequence from the QuicKeys menu.
3. Press **⌘-X** to cut the selected text to the Clipboard.
4. Select Scrapbook from the Apple menu.
5. Press **⌘-V** to paste in your cut.
6. Select Word from the Apple menu (or application menu if you are running System 7).
7. Select the Stop Recording... command from the QuicKeys menu. When the Sequence editor opens, name your macro and assign a keystroke. Assigning **⌘-Option-X** would make it easy to remember. Since this conflicts with the Move from command in Word, you could assign **⌘-Shift-X**.

Variations: You may want to close the Scrapbook after pasting to it, but the macro runs faster when you leave the Scrapbook open. This macro would be ideal for use with graphics programs or page layout programs. If you are working in a word processor and want to search what you have cut, you could archive your cuts to another file. You can set the file to automatically open whenever you open your word processor. This method has the added advantage of retaining the document's formatting. If you are creating this macro in Word, you can add the name of the archive file to your Work menu and select the name of your archive file from the Work menu to bring it to the front. (To add a document to Word's Work menu, press **⌘-Option**+ and click on the title bar of the document. If one hasn't already been created, a menu called Work appears to the right of the Window menu, and the document appears as a menu item under it. You will need to have saved your document before you can add it to the Work menu.) These are the steps in the macro. You will need to have a document and your archive file window open.

1. Press **⌘-X**.
2. Select the archive file name from the Window menu.
3. When it comes to the front, press **Return** twice to open some space at the top of the file.
4. Press **⌘-⌊** to return to the top of your file.
5. Press **⌘-V**.
6. Press **⌘-⌊**. This step is necessary to prepare your archive for the next cut you want to insert.
7. Insert a Select second window QuickKey to bring your document window back to the front.

If you use the same archive file for an extended period of time, you might want to time-stamp each day's archives. You could incorporate this into a sequence that opens your documents and inserts a Date/Time QuickKey at the top of the archive file. While this archive solution may not be as elegant as having the program automatically archive the cuts for you, it has the advantage of being flexible and doing something that none of the built-in clipboards do—saving your cuts even when you quit the word processing program.

Transpose Character

If you often transpose characters when you are typing, you may wonder why none of the popular word processing programs have a transpose character command. If you know you have transposed characters, it is faster to flip them than to spell check the word and correct it. You can create a macro that flips the characters for you and makes the process even faster.

Before we begin: You should have the program for which you want to create the transpose character macro open. The program should have keyboard shortcuts that allow you to select one character at a time. For example, you should be able to press **Shift-→** to select the character to the right (Word uses the **⌘-Option-Shift-L** knuckle-breaker). I'll be using Word 5.0. Open QuicKeys and select Microsoft Word from the pop-up menu.

Create the macro:

1. Position the insertion point after the second of the two characters that you want to transpose.
2. Select Record Sequence from the QuicKeys menu.
3. Press **Shift-←** to select the character.
4. Press **⌘-X** to cut the character.
5. Press **←** to move the insertion point before the first character.
6. Press **⌘-V** to paste the character.
7. Press **→** to move the cursor back to its starting location.
8. Select Stop Recording... from the QuicKeys menu.
9. When the Sequence editor opens, name your macro and assign a keystroke (Fig. 3-19). Try to use a mnemonic like **⌘-T**. Of course, you will have to make sure this doesn't conflict with any of the keyboard shortcuts already in the program with which you will use this macro.
10. Click OK to close the Sequence editor and save your changes.

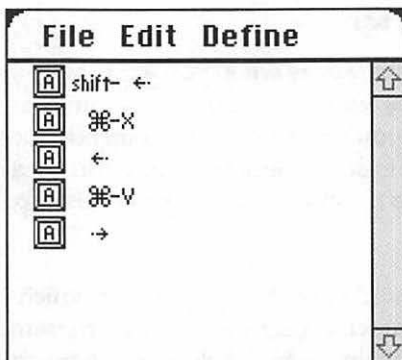


Figure 3-19: The completed GenTransposChar sequence.

Now when you transpose characters, you can press the keystroke to immediately flip them around.

Variations: This version of the transpose character macro should work in any program that lets you select a character as you move the insertion point. While not every program supports this, all of the word processors and some of the text editors do. In Microsoft Word, you could use its keyboard shortcuts to select the character and move the insertion point. A Word-specific version of the macro would look like Figure 3-20.

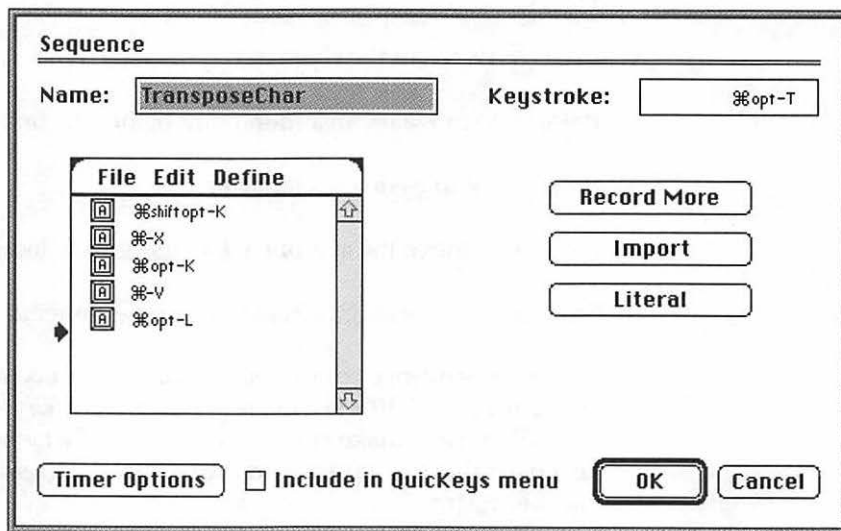


Figure 3-20: The Word-specific transpose character sequence.

Transpose Word

The transpose word macro is similar to the transpose character. It allows you to change the order of words as soon as you realize they are out of order. If you notice the words are transposed while you are typing, the macro changes their order with a single keystroke.

Before we begin: You should have a program open that has keyboard shortcuts for selecting words. I'll be using WriteNow, though Nisus, Vantage, and QUED/M all work the same way. Open QuicKeys and select WriteNow—or your program's name—from the pop-up menu.

Create the macro:

1. Place the insertion point exactly one space after the second word of the pair you want to transpose. If you place the insertion point immediately after the word, the space is not placed between the words when the macro finishes.
2. Select Record Sequence from the QuicKeys menu.
3. Press **[Option]-[Shift]-[←]** to select the second word.
4. Press **[⌘]-X**.
5. Press **[Option]-[←]** to move the insertion point before the first word in the pair.
6. Press **[⌘]-V** to paste the word.
7. Select Stop Recording... from the QuicKeys menu.
8. When the Sequence editor opens, name the macro and assign a keystroke (Fig. 3-21).

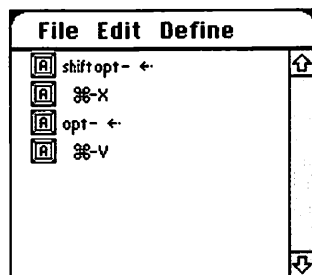


Figure 3-21: The complete TransposeWord sequence.

Variations: This version of the transpose word macro should work in any program that allows you to move the insertion point one word at a time and select as you move. While not every program supports this, all of the word processors and some of the text editors do. In Microsoft Word, you could use its keyboard shortcuts to select the word as you move the insertion point. A Word-specific version of the macro would look like Figure 3-22.

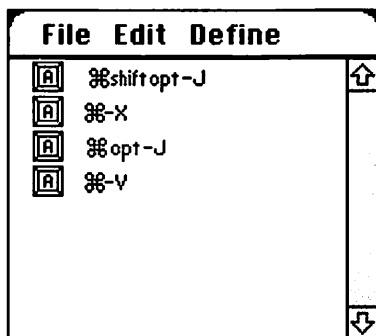


Figure 3-22: The Word-specific transpose word sequence.

The two transpose macros are pretty simple, aren't they? I am always surprised when another new word processor or an update of an existing program ships without built-in transpose commands. Maybe no one else makes transposition mistakes.

Print Setup

Have you ever pressed **⌘-P** to print your file and then walked away from your desk forgetting that you still have to press **Return** to start printing? I'm constantly forgetting to respond to all of the print dialog boxes in some programs. You can solve this problem by creating a macro to remember what you forget. For example, the macro could open the print dialog box, set up all of the options, and then send your file to the printer. If you assign it to **⌘-P**, you would never again have to wait for your file to print.

Before we begin: You will need to be in the program from which you want to print. I'll use Word 5.0.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Select Page Setup... from the File menu.

3. When the Page Setup dialog box opens, click in the Font Substitution, Font Smoothing, Graphics Smoothing, and Faster Bitmap Printing checkboxes.
4. Click the OK button or press **Return** to close the dialog box and save your changes.
5. Select Print... from the File menu and then click the Cancel button when the dialog box opens. This adds a step to our sequence that we'll have to remove, but it allows us to close the Print dialog before we open QuickKeys. In tight memory situations, opening QuickKeys on top of another dialog box can crash your Macintosh. We'll change this QuickKey in the Sequence editor.
6. Select Stop Recording... from the QuickKeys menu to open the Sequence editor (Fig. 3-23).

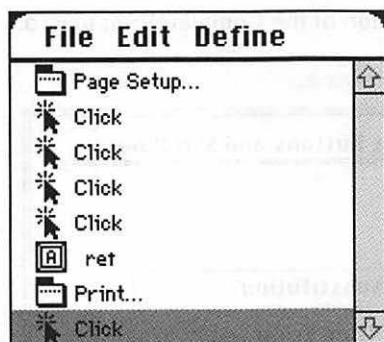


Figure 3-23: The Sequence editor for the CompletePrint macro.

7. Let's remove the click that you entered in step 6. Select the last click in the sequence and then select Cut from the Edit menu.
8. Click the Literal button and press **Return**. This clicks the Print button in the Print dialog box.

If you open the individual click QuickKeys and click the Control Area button, you'll see that each one contains the name of the checkbox you clicked. Each one also contains the number of the checkbox you clicked (Fig. 3-25). Since we want to turn the options off, you should make certain the Only click if button is on option is selected.

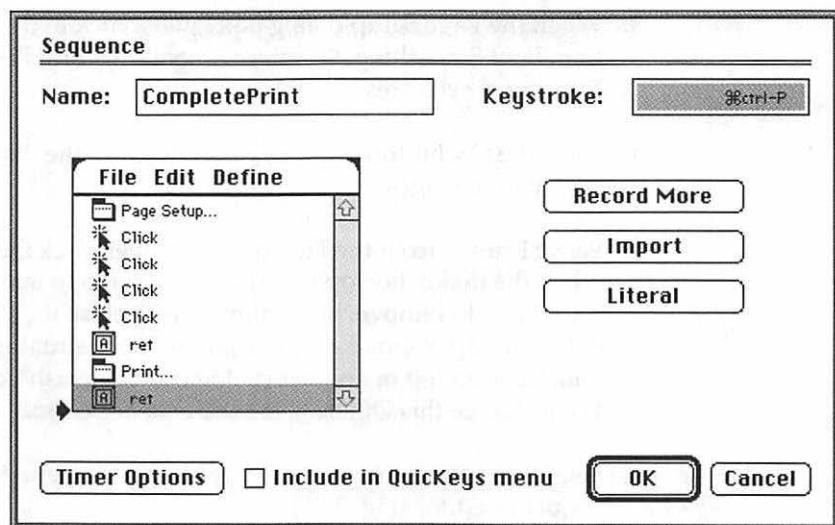


Figure 3-24: The final version of the CompletePrint macro.

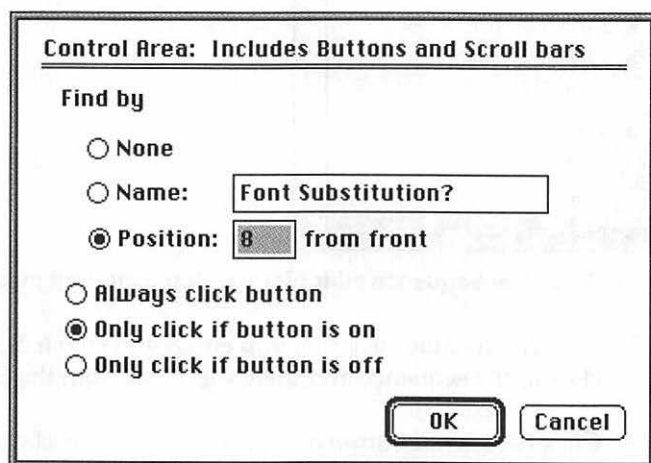


Figure 3-25: The Control Area dialog box from the Click editor.

9. Name your QuickKey and assign a keystroke that is similar to your program's Print shortcut. If you don't know what your program's print shortcut is, look next to the Print command in the File menu.

Variations: This macro lends itself to many different special-purpose macros. For example, you can use the Hidden command in Word's Character dialog box to insert notes in your document (Fig. 3-26). Select

the text you want hidden, open Character dialog box, and select the Hidden checkbox. The hidden text is not printed until you select the Print Hidden Text checkbox in Word's Print dialog box. Now you can modify the CompletePrint macro to toggle the Print Hidden Text checkbox off and on. Other possibilities are changing the macro to check Word's Print Selection Only option and unchecking the Reverse Printing checkbox in MacWrite II.

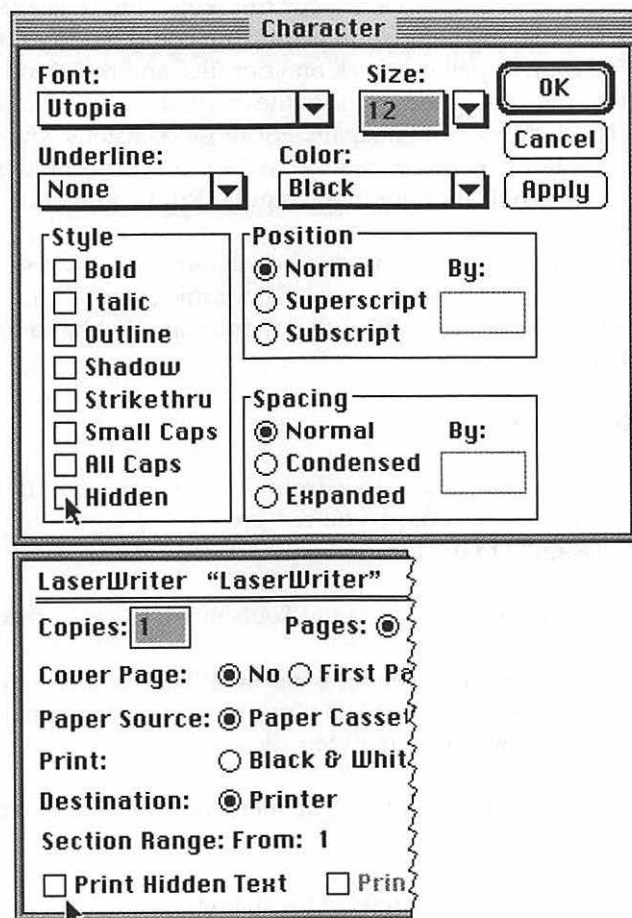


Figure 3-26: Word's Character dialog box. The Hidden checkbox in the Style group lets you hide the selected text.

I use a variant of this macro to print envelopes in Dynodex. If you have selected one of Dynodex's envelope layouts, the Print One command prints an envelope of the current record. Unfortunately, it opens both

the Page Setup and Print dialog boxes before it starts the print job, and I always forget to press **[Return]** for both dialog boxes. I have a simple PrintOne macro that selects the Print One command from the File menu, clicks the OK button in the Page Setup dialog box, and then clicks the OK button in the Print dialog box.

Spell Check, Then Print

Do you always find typos in your final copy, but never can remember to check the spelling before you print? You could modify the previous macro to run a spelling check on your files and then send them to the printer. If you assigned **[⌘]-P** as the shortcut, you would never forget to spell check the document again. For those occasions when speed demands that you throw caution to the wind, you could assign a slightly different shortcut to the regular Print command.

Before we begin: You will need to be in your word processor with a file already open. I'll be using Word 5.0 for this example, but you can modify the macro to work with any program that has a built-in spelling checker.

Create the macro:

1. Select Record Sequence from the QuicKeys menu. If you have an extended keyboard, you can press **[F8]** (or the key you have assigned to the Record Sequence command).
2. Select Spelling... from the Tools menu to begin checking the file.
3. Now close the Spelling window. When we finish the macro, we will insert two Wait QuicKeys so the sequence pauses until you are through checking your file.
4. Select Print... from the File menu or press **[⌘]-P** to open the Print dialog.
5. Set all of your options in the dialog.
6. Click the Cancel button in the Print dialog. We'll change this in the final macro, too.
7. Select Stop Recording from the QuicKeys menu to stop recording and open the QuicKeys dialog. If you have an extended keyboard, you can press **[F8]**.

8. Now we need to make two adjustments to the macro. Cut the Click QuickKey at the end of the sequence that closes the Print dialog box. In its place let's add a Literal QuickKey to click the Print button. Click the Literal button in the Sequence editor. When the dialog box appears telling you to press any key, press **Return** (Fig. 3-27).

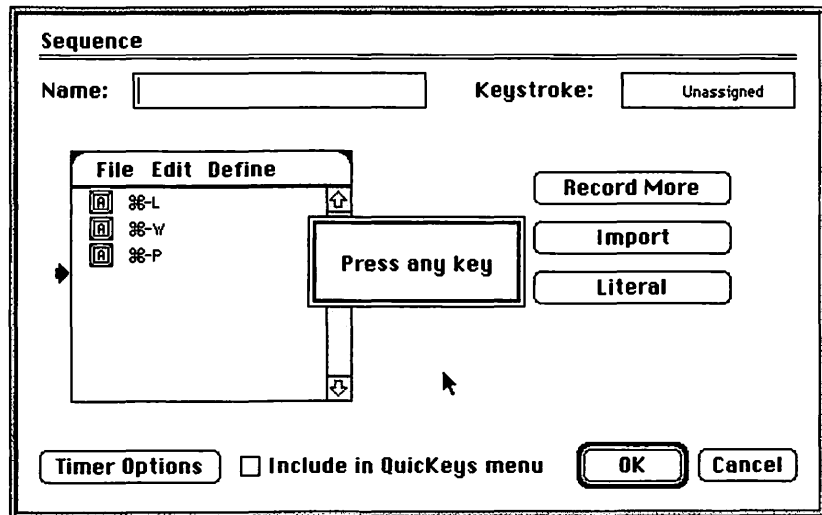


Figure 3-27: Adding a Literal to the CheckThenPrint macro.

9. Now let's add Wait... QuickKeys to pause the macro while you are checking the file. Move the arrow on the right side of the sequence editing window just below the Alias QuickKey that opens the Spelling window.
10. Select Wait... from the Extensions hierarchical menu. When the Wait extension opens, select Window Changes from the pop-up menu. The Wait extension allows you to pause a QuickKeys sequence while you enter information in a dialog box or window. The Wait QuickKey pauses the sequence while you check the spelling in your file and reactivates the sequence whenever the window changes. In this sequence the change occurs when the Spelling window closes.
11. Select Wait... from the Extensions hierarchical menu again. We add the second Wait QuickKey because Word displays a dialog box telling you it is through checking the file. So the window changes twice before your file window comes to the front again.

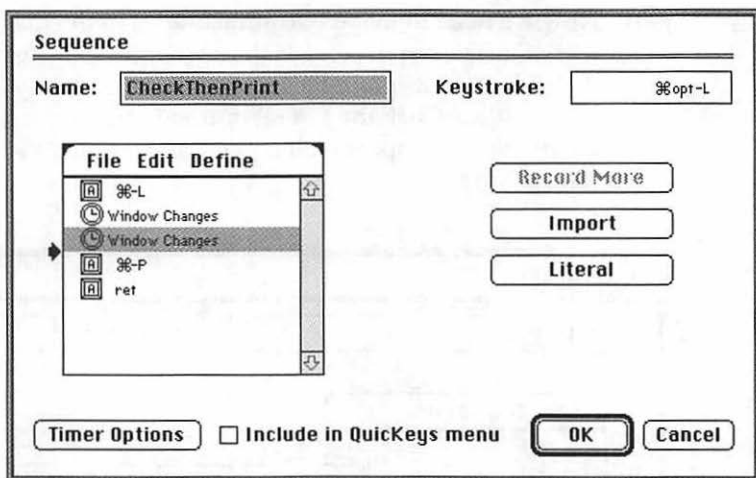


Figure 3-28: The final version of the CheckThenPrint macro.

12. Name your sequence and assign a keystroke. Your final sequence should look like Figure 3-28.

You might want to assign a keystroke that is close to the spelling shortcut. For instance, **Ctrl-⌘-L** is close to Word's **⌘-L** shortcut and doesn't conflict with any of Word's other shortcuts. Now you are ready to try your new macro. After it opens the Spelling window, you'll notice the Pause icon (two flashing bars similar to the pause button on a VCR or CD player) takes the place of the Apple. Clicking the OK button in the Finished checking dialog box causes the window to change twice (the dialog box and the Spelling window both close) and resumes the sequence.

Variations: You can modify this macro to work in just about any word processor. In MacWrite II, you only need one Wait QuickKey and you would change the first QuickKey to select the Check All command under the Spelling menu. For WriteNow, you'll change the first QuickKey to select the Check Spelling command under the Edit menu. You could also change the macro to click the OK button in the Finished Checking dialog box (Fig. 3-29). Select the Dialog in Front option from the pop-up of the second Wait QuickKey and insert a **Return** literal after it to click the OK button.

As you can see, macros can make your Macintosh and your programs smarter. And the macros make your programs work the way you work—accommodating your odd habits and not someone else's.

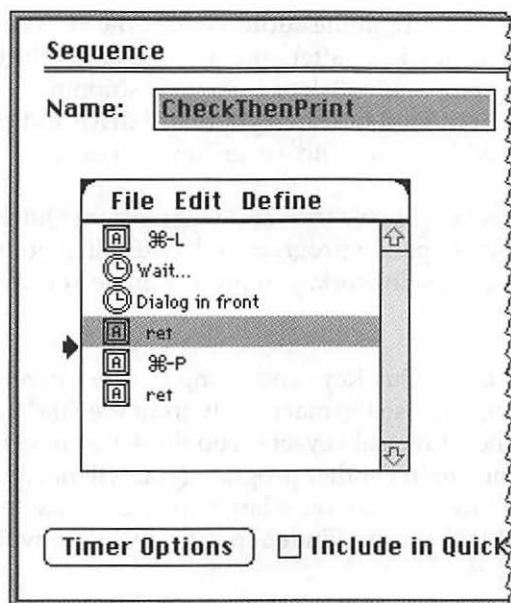


Figure 3-29: Modifying the CheckThenPrint macro to click the OK button in the Finished checking selection dialog box.

Now I want to show you how to make your Macintosh operate on auto-pilot.

Running Your Macintosh on Auto-Pilot

Your Macintosh has the potential to do most of your work for you. Best of all, you only have to train your Macintosh one time.

Start Me Up

Bob LeVitus, of *Dr. Macintosh* fame, once told me he had his most frequently used files and applications assigned to the function keys of his extended keyboard. When he wanted to load Word, for example, he pressed ⌘-F1 . This may seem a rather trivial use for a program as powerful as QuicKeys, but you shouldn't overlook the immediate benefits. If you have a large hard drive filled to overflowing with applications and files, navigating the folders can take time and be frustrating. While programs like On Cue, HandOff II, and MultiMaster allow you to launch applications and open files, they can make heavy demands on your Macintosh's memory (and your patience whenever extensions, or INITs, step on or collide with one another). If you are trying to run a lean and clean system, macros can provide some of the

same functionality without the additional overhead. As a matter of fact, if you ran Instant QuicKeys after you installed either the QuicKeys demo from the companion disk or from your shipping copy of QuicKeys 2.1.2, it offered to search your hard drive and create a list of applications to which you could assign keystrokes.

Let's go a step beyond Bob's trick and what Instant QuicKeys does. Let's create a macro that opens a program and loads all of your current files. When you want to begin work, you press a single keystroke to get started.

Before we begin: Open QuicKeys and change to the Universal or Finder keyset. If you plan to use the macro only from the Finder, select that keyset. Select the Universal keyset if you think you may want to use the macro while you are in another program (you will need to be running System 7 to do this). If you have a large number of macros, you should make the StartMeUp macro Finder-specific. Doing so will give you more free shortcut keys.

Create the macro:

1. Open the QuicKeys editor if you closed it.
2. Select Sequence from the Define menu.
3. When the Sequence editor opens, select Finder Events from the Extensions menu.
4. When the Finder Events extension opens, select Open from the pop-up menu. A dialog box appears where you can select files or programs that you want to open (Fig. 3-30).
5. Select the files and click the Add button. When you are through adding files, click the Done button. The list of files appears in the Finder Events editor (Fig. 3-31). The type and creator of the files appear below their names. If you select documents, the program is automatically loaded.
6. Name the extension. You don't have to do this, but it saves you the trouble of opening the extension to see which files you have chosen if you give it a general name.
7. Click the OK button or press Return to close the editor. If you want to add more programs to your StartMeUp sequence, you

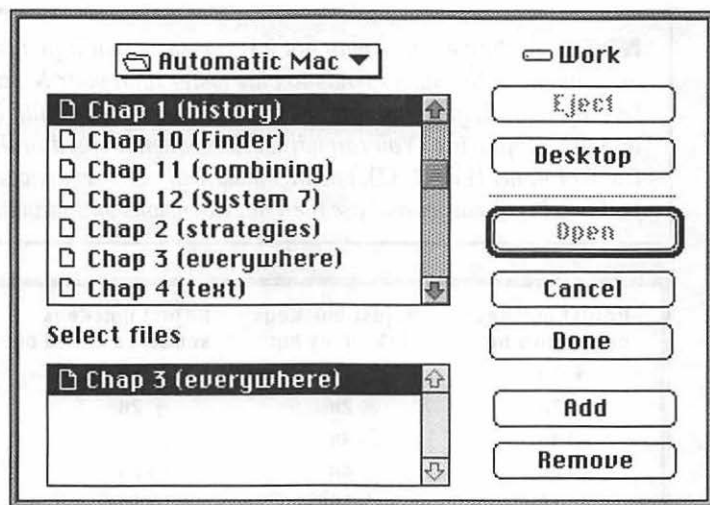


Figure 3-30: The Finder Events open dialog where you can add or remove files and programs the extension opens.

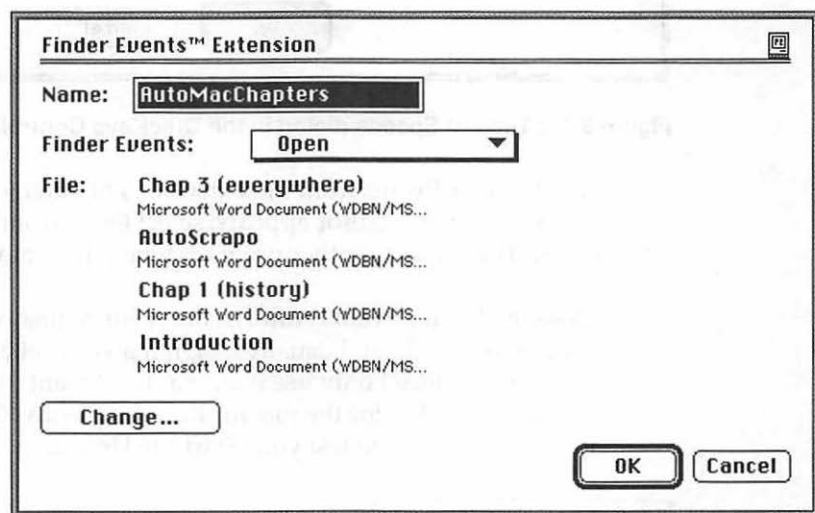


Figure 3-31: The Finder Events extension with Word files selected.

could add them into the same Finder Events dialog. However, in the interests of modularity, you may want to create a separate Finder Event for each different program. We'll add another program to this sequence in just a moment.

8. Select Pause from the bottom of the Define menu.

NOTE: Adding a pause may not be necessary; you'll just have to experiment. If the macro tends to play faster than your Macintosh can keep up with it, you'll hear beeps when dialog boxes should be opening or windows appearing. You can adjust the sequence speed in the QuickKeys Control Panel (Fig. 3-32), but this makes all your sequences run slower. I prefer to insert pauses and use the Wait extensions for individual sequences.

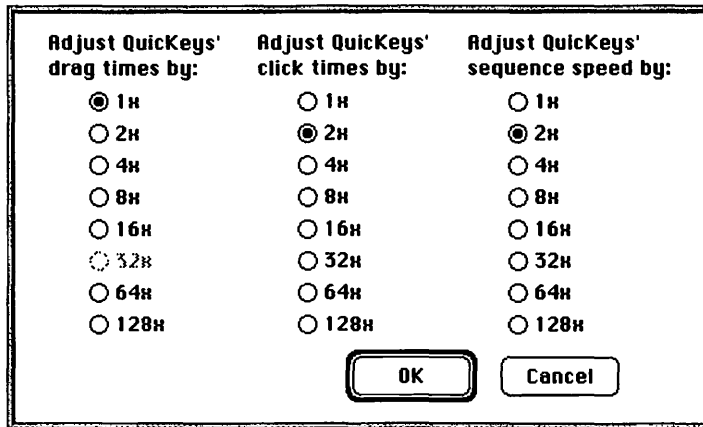


Figure 3-32: The Set Speeds dialog in the QuickKeys Control Panel.

9. Select Finder Events from the Extensions hierarchical menu. When the Finder Events editor appears, select Open from the pop-up menu. Then select another program when the dialog box appears.
10. Close the Finder Events editor, name your sequence, and assign a keystroke (Fig. 3-33). I usually assign it a keystroke I wouldn't use very often—since I only use it once a day, I want to save the simple keystrokes for the macros I use frequently. Close QuickKeys and you are ready to test your Start Me Up macro.

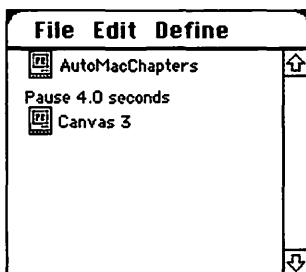


Figure 3-33: The Final StartMeUp sequence.

Variations: Probably the most obvious variation would be to make StartMeUp a timed macro. You could add a time in the QuickKeys window so the macro plays 2 minutes after your Macintosh starts up in the morning (Fig. 3-34).

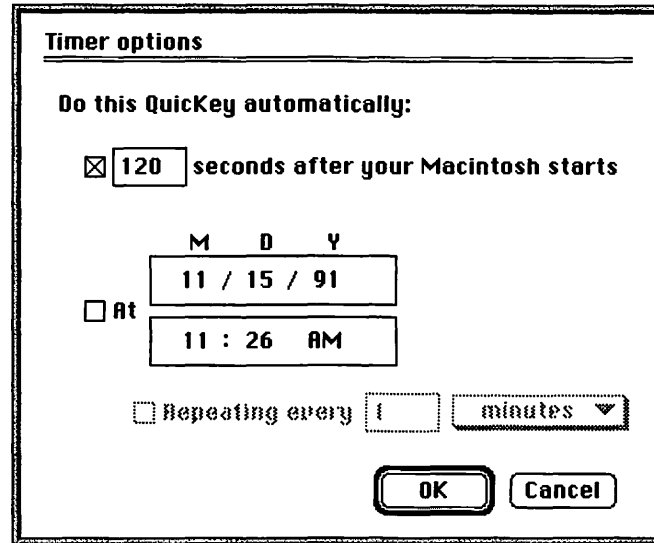


Figure 3-34: The Timer options dialog.

Combine this macro with the PowerKey device to turn your Macintosh on at 7:00 AM, and it would be ready to go when you get out of the shower in the morning.

If you don't have a great deal of RAM in your Macintosh, you may want to limit the number of programs you load each morning. Basically, you should load the program or programs you find yourself opening when you first sit down at your Macintosh. If the program takes a long time to load (like Word 5), it is better to have it ready to go than drum on your keyboard while you wait for it to load. If there is a document or template you open everyday, you should include it as well.

Save All

If you often work with multiple files open, you usually have to go through the process of saving each file separately. Generally, this involves pressing **(Return)** or clicking the OK button as the program asks you if you want to save each document. You can save yourself a few keystrokes by creating a macro that cycles through all of the open

documents and saves each one. While your Macintosh is saving your work, you can clean the top of your desk.

Before we begin: You should have several files open (and already saved once) in the program you want to use. Open QuickKeys and select the keyset for the program for which you want to create the macro. I'll use Word 5.0, but you should be able to use this macro for any program that allows you to have multiple open files.

Create the macro:

1. Open QuickKeys and select Sequence from the Define menu.
2. When the Sequence editor opens, select Repeat from the Extensions hierarchical menu. We want to create a loop that will cycle through each document window and save it, so we use Repeat to start and stop the loop.
3. Select the Begin Repeat button for the function and select the Put up dialog checkbox under Additional values (Fig. 3-35). It opens with a default number of times to repeat. When the sequence runs, the Repeat extension asks you how many times it should repeat the sequence. You would enter the number of documents you have open (Fig. 3-36). While this seems like a bother, it takes fewer keystrokes to enter the number of documents and press **Return** than it does to press **Return** for each document.

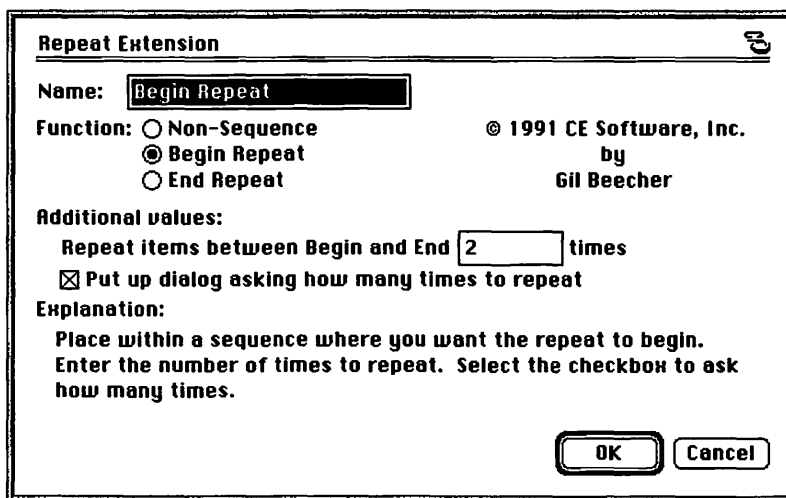


Figure 3-35: The settings for the first Repeat extension in the sequence.

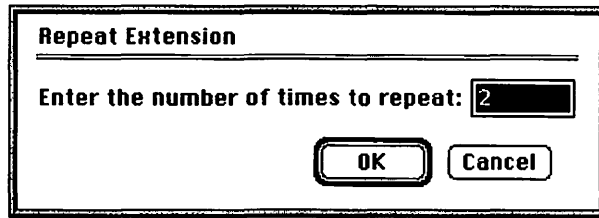


Figure 3-36: The Repeat dialog box you see when the sequence runs.

4. Click OK or press **Return** to close the extension and save your changes.
5. Select the Select rear window QuickKey from the Specials menu and then click OK. This cycles through all open documents.
6. Click the Literal button and then press **⌘-S**. This literal QuickKey saves the document. Click OK to close the editor and save your changes.
7. Select Repeat from the Extensions menu again. When the extension appears, select End Repeat for the Function setting. This marks the end of the repeat. Press **Return** or click OK to close the editor and save your changes.
8. Name your sequence and assign a keystroke. You should assign a keystroke that is mnemonically close to the program's Save command (e.g., **⌘-Option-S**). Your final sequence should look like Figure 3-37.

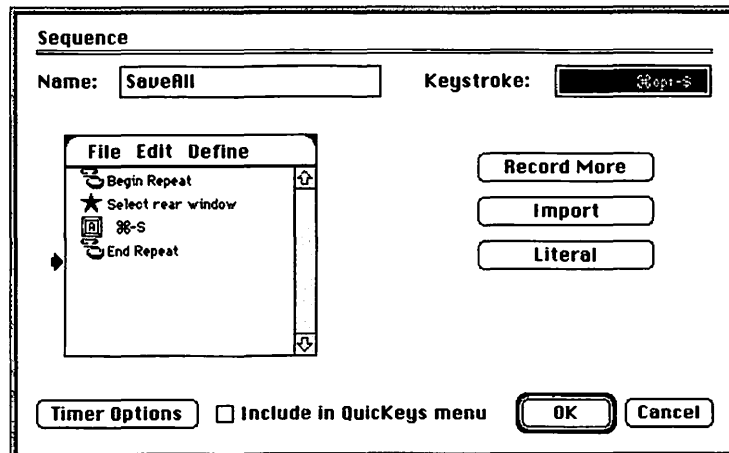


Figure 3-37: The final SaveAll macro.

Variations: You may want to expand this sequence to save your changes and then quit the program. To do this, you would insert a Close window mouseie after the ⌘-S and a ⌘-Q after the End Repeat extension to quit the program. You could expand this macro even further to save your documents and quit all of the open programs and shut down your Macintosh at the end of the day.

Timed Archives

No matter how well-behaved the program, you may eventually crash it or have trouble opening a file. Some programs, like Aldus PageMaker, compress the document whenever you save a copy, so the copy is smaller than the original. For your sanity's sake, it is a good idea to always make a backup copy of your files. Not all programs make backup copies for you, and you can make a macro for programs that don't have this feature. The macro ensures that you always have more than one copy of what you are doing on your hard drive. If the first QuickKey in the sequence is the MessageAction extension, you can always choose not to create the copy.

Before we begin: You will need to open the program for which you want to create the macro. I'm going to use PageMaker for this one. Open QuickKeys and select PageMaker from the pop-up menu. If you are going to make a timed reminder, you should save it in PageMaker's keyset. Otherwise, the macro will run no matter what program you are in, and you might have some unexpected results. By saving the timed sequence in PageMaker's keyset, it will only run when PageMaker is open.

Create the macro:

1. Select Sequence from the Define menu.
2. When the Sequence editor opens, select Message from the Extensions hierarchical menu.
3. Enter a message in the Message box. Select the checkboxes for the Show OK and Show Cancel buttons (Fig. 3-38). You can also select the Continue automatically option and enter an amount of time for the pause if you want the QuickKeys to save your work when you don't respond to the message dialog box. Click OK to close the editor and save your changes.

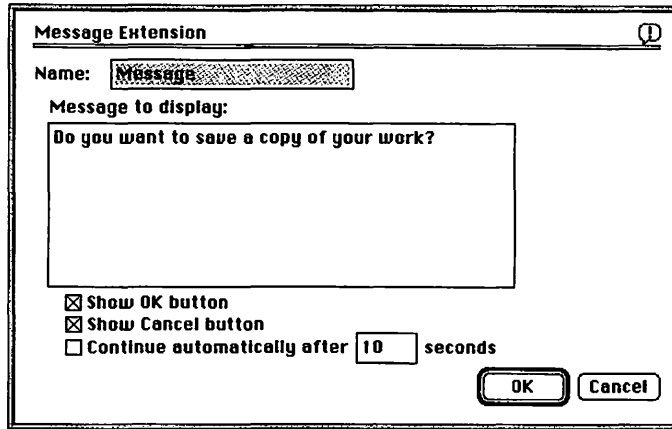


Figure 3-38: The Message extension for the SaveCopy macro.

4. Select Menu/DA... from the Define menu and then select the Save As... command from the File menu. Click OK to close the editor.
5. Select WindowAction from the Extensions menu. When the extension opens, uncheck the Window Name checkbox, select the Window Type checkbox, select Dialog from the pop-up menu (Fig. 3-39), then select the Wait... button, and click OK. This pauses the sequence until the Save As dialog box opens.

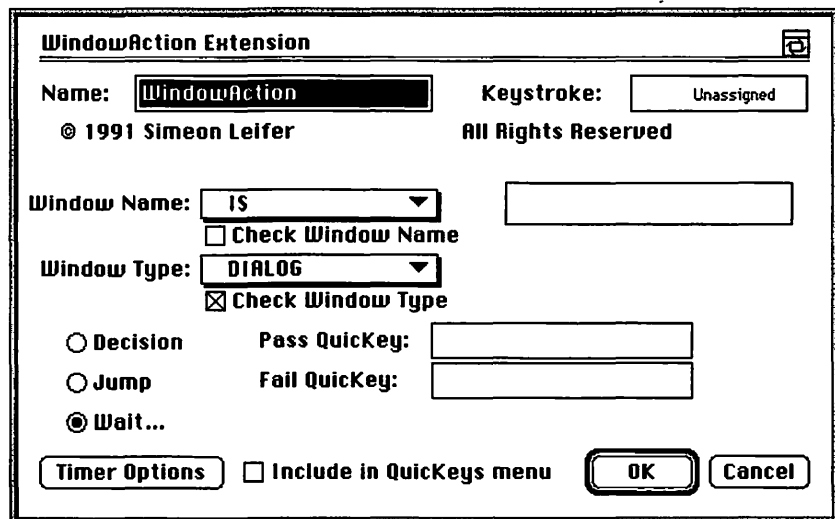





Figure 3-39: The WindowAction extension for the SaveCopy macro.

6. Click the Literal button and press the . This places the insertion point after the current name of the file in the Save As dialog.
7. Select Text from the Define menu. When the Text editor opens, type a space followed by copy or backup. When the sequence is playing, this places a space after the filename and then adds the word copy or backup to the filename.
8. Click the Literal button and press .
9. Select CursorWait from the Extensions hierarchical menu. When the extension opens, select Is and the Watch cursor from the pop-up menus. Select the Check Cursor Mask checkbox. Then click OK or press  to close the extension and save your changes. This extension pauses the sequence until the watch cursor is replaced by the arrow, that is, until PageMaker finishes saving the file. If you don't insert it, the sequence tries to play before PageMaker is ready.

TIP: Whenever you are creating long sequences, you should name the individual extensions something descriptive. The name makes it easier to tell what the sequence is doing without opening all of the extensions to look at the settings. In this instance, you could name the extension *Cursor(Watch)* since it is waiting until the cursor changes to a watch.

10. Select Menu/DA... from the Define menu and then select the Close command from the File menu. Now click OK to close the Menu editor. Since your current document is now the backup copy, you need to close it and reopen your original.
11. Select Menu/DA... from the Define menu and then select the Open... command from the File menu. Click OK to close the Menu editor. This displays the Open dialog so you can reopen your original document.
12. Name and assign a keystroke to your sequence. The completed sequence should look something like Figure 3-40.
13. To make it a timed sequence, click the Timer Options button. When the dialog box opens, enter 3600 in the seconds after start box (Fig. 3-41). Click OK to save your changes. Now the sequence will play 60 minutes after you open PageMaker.

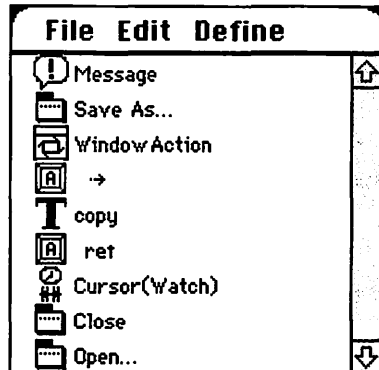


Figure 3-40: The completed SaveACopy macro.

NOTE: If you have already saved a copy of your document on your hard disk, then you will see another dialog box asking whether or not you want to replace the existing copy. To avoid this, you must insert another step to anticipate the dialog box or you may want to add a Date/Time QuickKey instead of the Text QuickKey. See the variation suggestions below.

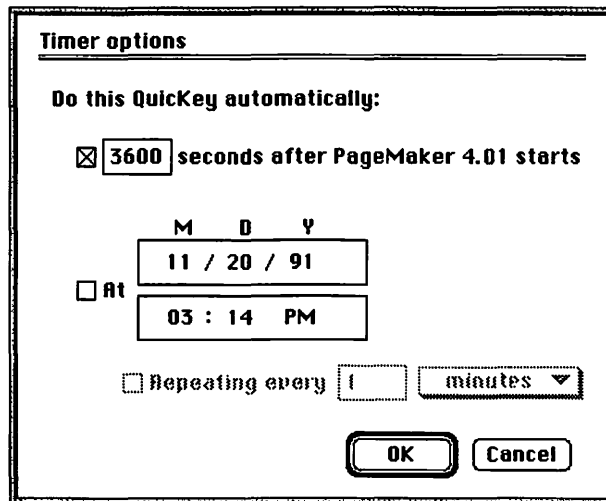


Figure 3-41: Setting the frequency the macro runs in the Timer Options dialog box.

Variations: Instead of saving a copy every hour, you may prefer to save a copy whenever you close your document. If you make this change, the sequence would not interrupt your work. You would add a Save Menu

QuickKey to the beginning of the sequence and follow it with a CursorWait QuickKey to pause the sequence until the program saves your current edits. At the end of the sequence, you would replace the Close and Open Menu QuickKeys with a single Quit Menu QuickKey (Fig. 3-42). Now the sequence will pause until the program has saved the copy of the file and then quit the program.

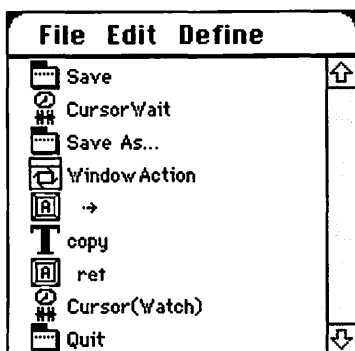


Figure 3-42: The Final SaveACopy&Quit sequence.

You could also vary the sequence by adding the date to the file name. You would do this by inserting a Date/Time QuickKey in place of the Text QuickKey. Now you would have multiple versions of the document saved with the date you worked on them appended to their names.

Time Logger

If you do any contract work, you can create a macro that logs your time. Every hour or half hour, a message pops up and asks you if you would like to record what you've been doing. If you click yes, the document that you keep the log in comes to the front of the screen, and you can enter a brief note. If you click cancel, the action is delayed until the next time the logger macro is activated. I'll be using TeachText for this macro.

Before we begin: You will need to create a TimeLog file with TeachText. Open QuickKeys and select Universal from the pop-up menu.

Create the macro:

1. Open QuickKeys and select Sequence from the Define menu.
2. Select Message from the Extensions hierarchical menu.

3. When the extension opens, enter your message. Select the checkboxes for the Show OK and Show Cancel buttons (Fig. 3-43). Click OK to close the extension.

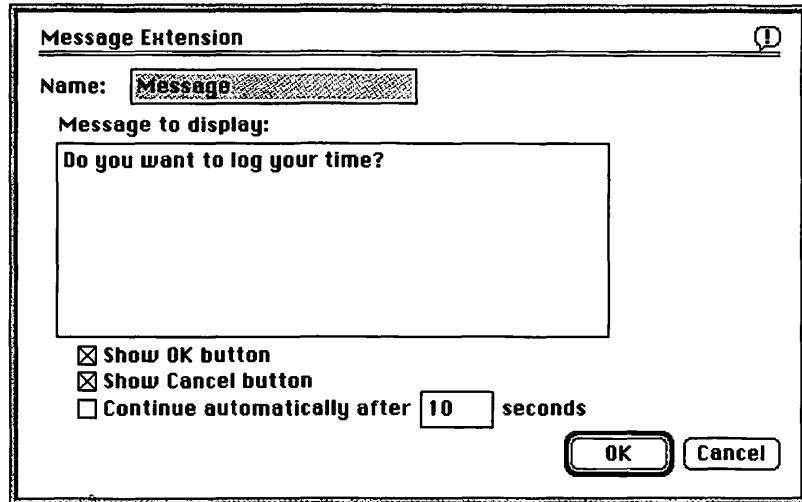


Figure 3-43: The Message extension for the TimeLogger macro.

4. Select Finder Events from the Extensions hierarchical menu if you are running System 7. If you are running System 6 and MultiFinder, you will need to select File... from the Define menu. When the Finder Events editor opens, select Open from the pop-up menu. A dialog box opens where you can select the file for the Finder Event to open. Select the TimeLog file and then close the dialog box. Click OK to close the Finder Events editor and save your changes.
5. Select WindowAction from the Extensions hierarchical menu. When the extension opens, enter the name of the window in the Window Name box and select Is from the pop-up menu. Inserting this extension causes the sequence to wait until the desk accessory window comes to the front. Otherwise, part of the time-stamp might be entered in the window in which you are working. Click OK to close the extension.
6. Select a date format from the Date/Time menu. Click OK.
7. Click the Literal button and press **Return**.
8. Select a time format from the Date/Time menu. Click OK.

9. Click the Literal button and press **[Return]** again.
10. Name your sequence. Your finished sequence should look like Figure 3-44. Now click the Timer Options button to set the frequency with which the sequence should run. After the macro runs and inserts the time-stamp in your TimeLog file, you can enter a brief note.

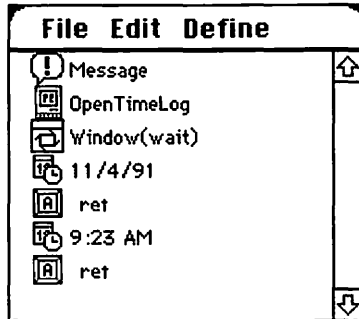


Figure 3-44: The completed TimeLogger macro.

Because TeachText doesn't support the **[J]** key to move you to the end of the document, your notes will run in reverse order—from most recent to the earliest. You may want to move the cursor to the beginning of the TimeLog file and enter a couple of blank lines so you will always have blank lines between each note.

Variations: If you only want to log your time when you are working in a certain program, you can save the TimeLogger macro to that program's keyset. If you don't want to be bothered by the message, you can change the macro so it is activated by a keystroke. Another variant of this macro would just display a message reminding you to log what you are doing. You can use a word processor or different text editor to log your time. If you choose one that supports the **[J]** key, you can insert an End mousie after the WindowAction extension to move the insertion point to the end of the document. You can also use a desk accessory for the macro. For example, FlashWrite II, a shareware program by Andrew Welch, is similar to the Note Pad desk accessory, but it has the advantage of always opening to the same page with the cursor in its last location. This simplifies the macro because we don't have to look for the correct page, open a different file, or look for the last line in the file. The desk accessory does all of that for you.

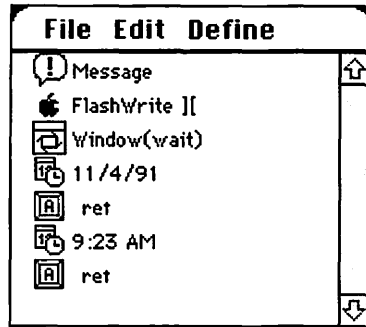


Figure 3-45: The TimeLogger sequence for FlashWrite II.

You will find a sequence for FlashWrite II in the Universal sequence on the companion disk.

Change Creator

Generally, you can open a text file in any text editor or word processor. However, you usually have to open the text editor before you can open the other program's text file, that is, you cannot double-click on the file in the Finder and open your favorite text editor. This can be a problem if you don't have the program that created the text file. When you double-click on the file, the Finder will tell you that it can't open the file because the application that created it cannot be found. For example, you may want to change all of your MacWrite files to have the same creator as MacWrite II files so that you can open MacWrite II by double-clicking on your MacWrite files. The simplest way to do this is to change the creator of the file so the Finder thinks it was created by your text editor. It is difficult to change the type or creator of a file if you don't have a certain amount of savvy. But even if you know how to change the type or creator, it is next to impossible to change a large number of files at once.

You can automate the process of changing either the type or creator for a group of files using one of the file manipulation programs: DiskTop, MasterFinder, or DiskTools II (refer to Appendix A, "Where Do I Find This Stuff?"). It is easy to do with DiskTop since this program displays a list of files in a window. It is harder to do in MasterFinder and DiskTools II because you can't get to the Get Info window directly from the list of files. Let's create a macro using DiskTop to change the creator of a group of files.

Before we begin: Open DiskTop and then open QuickKeys. If you are running System 6 and MultiFinder, select DA Handler from the pop-up

menu. If you are running System 7, select DiskTop from the pop-up menu. Now the macro only plays when you are in DiskTop or the DA Handler. If you are using System 6 without MultiFinder, you will need to save the sequence in the Universal keyset. The files you want to change should be displayed in the DiskTop window. I am going to change the creator of my America Online text files so I can open them directly with my favorite text editor.

Create the macro:

1. Open QuickKeys and select Sequence from the Define menu.
2. When the Sequence editor opens, select Repeat from the Extensions hierarchical menu.

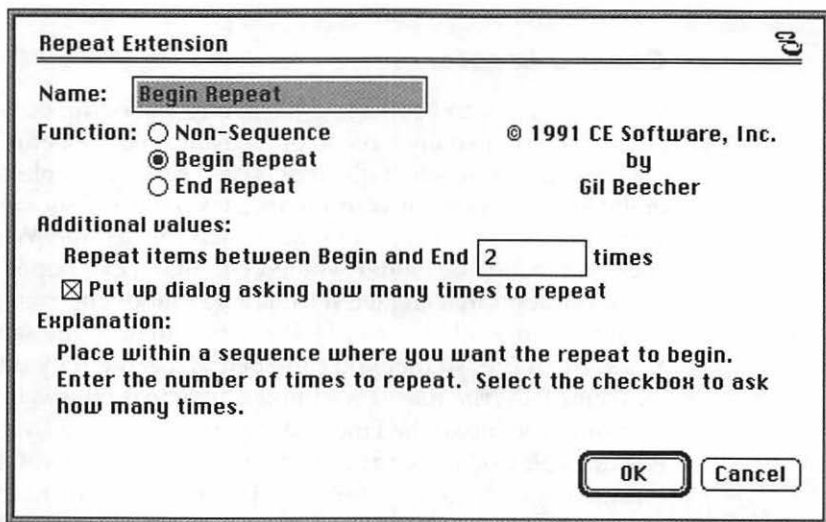


Figure 3-46: The Begin Repeat settings for the ChangeCreator macro.

3. Select the Begin Repeat button and select the Put up dialog checkbox (Fig. 3-46). When the macro runs, this extension displays a dialog box and asks you how many times it should repeat. Click OK.
4. Click the Literal button and press \downarrow . This selects the first file in the DiskTop window.
5. Click the Literal button and press ⌘-I . This opens DiskTop's Get Info dialog box (Fig. 3-47).

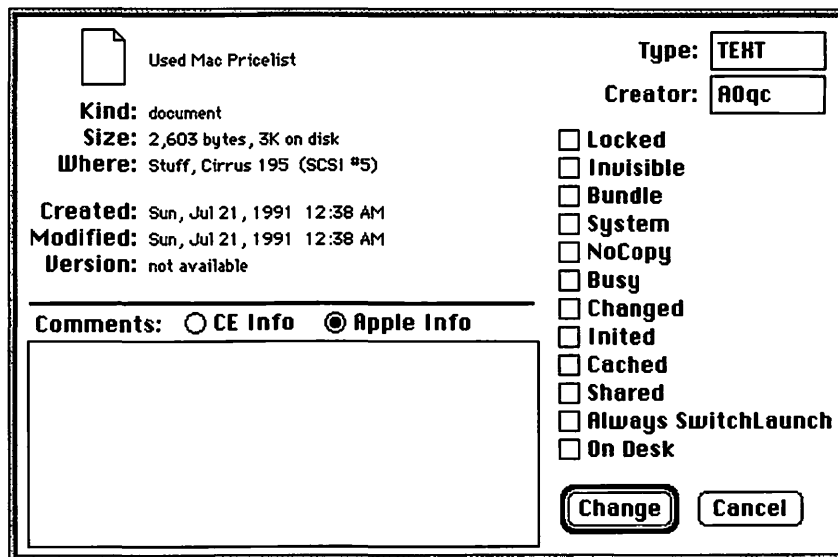


Figure 3-47: DiskTop's Get Info window.

6. Click the Literal button and press **Tab**. This selects the Type box.
7. Click the Literal button and press **Tab**. This selects the Creator box.
8. Select Text... from the Define menu. When the Text dialog box opens, enter the Creator abbreviation for the program you want to open the files. I'll enter "McSK" for the Vantage and McSink text editors. Click OK.
9. Click the Literal button and press **Enter**. This selects the Change button in the Get Info window and saves your changes. If your keyboard does not have an **Enter** key, you will need to add a Click for the Change button instead.
10. Select Repeat from the Extensions menu and select the End Repeat option to stop repeating the sequence.
11. Name your sequence and assign a keystroke. Your finished sequence should look like Figure 3-48.

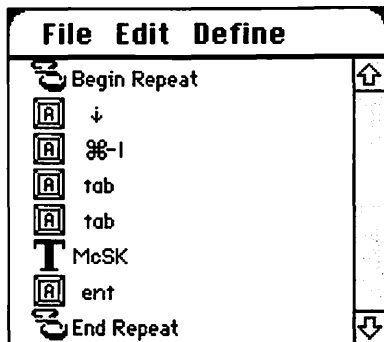


Figure 3-48: The completed ChangeCreator sequence.

Variations: If you don't have one of the desk accessories mentioned above, you could modify the sequence to work with ResEdit's Get Info command. However, the Macintosh Open dialog box doesn't remember the last file you selected, so you would also need to have Rebound, Boomerang, SuperBoomerang, or Shortcut installed to bestow some additional intelligence to the Open dialog (Rebound is included on the companion disk; refer to Appendix A, "Where Do I Find This Stuff?" for information about the other programs). Any one of these INITs helps the Macintosh remember the last file you selected and causes the ⌵ Alias to select the next file. Since you are working within the confines of the Open dialog, you would need to run the sequence once before beginning the repeat. Doing so lets you change the creator of the first file in the Open dialog box. Once you change the first file, you can begin the repeat so the macro steps through each file. Select the Get File/Folder Info command immediately after beginning the repeat (see Fig. 3-49).

Then enter a ⌵ to select the next file in the dialog box. You may need to enter a pause between the Menu/DA and the ⌵ Alias. The pause causes QuickKeys to wait until the dialog appears completely on screen. If you do not, Rebound causes the ⌵ to jump back and select the first file in the dialog box; this action is exactly what you are trying to avoid. You might want to add a Location QuickKey at the very beginning to set the Open dialog box to the appropriate folder before the sequence begins to run.

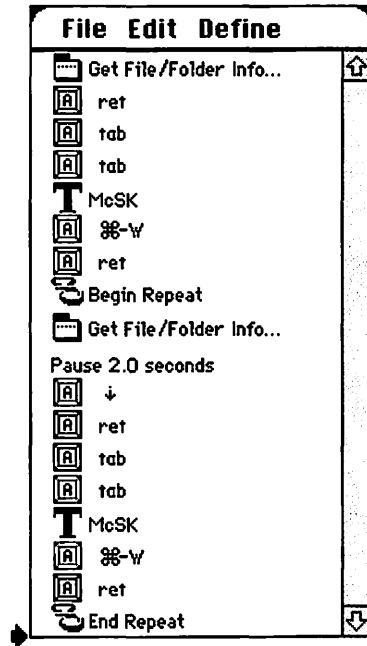


Figure 3-49: The ChangeCreator sequence modified to work with ResEdit's Get File/Folder Info command.

Extended Find

Of course, the files that you plan to change the creator of are probably scattered all over your hard disk. You can build upon the previous sequence in a rather natural way by invoking DiskTop's search command before running the repeat portion of the sequence. DiskTop can locate all of the files with a specific creator and retain them in a separate list window. Then you are ready to run your ChangeCreator sequence.

Before we begin: You should set up DiskTop and QuicKeys as described in the ChangeCreator sequence above. You should also have DiskTop.Extras and CE Toolbox installed so the DiskTop Find command is available from the Apple menu. These two programs are available on DiskTop's shipping disk. If you are running System 6, they need to be installed in your System folder. If you are running System 7, they should be installed in the Extensions folder inside your System folder. You should configure the CE Toolbox Menu Configuration with menus either Grouped or Sorted with DAs (Fig. 3-50).

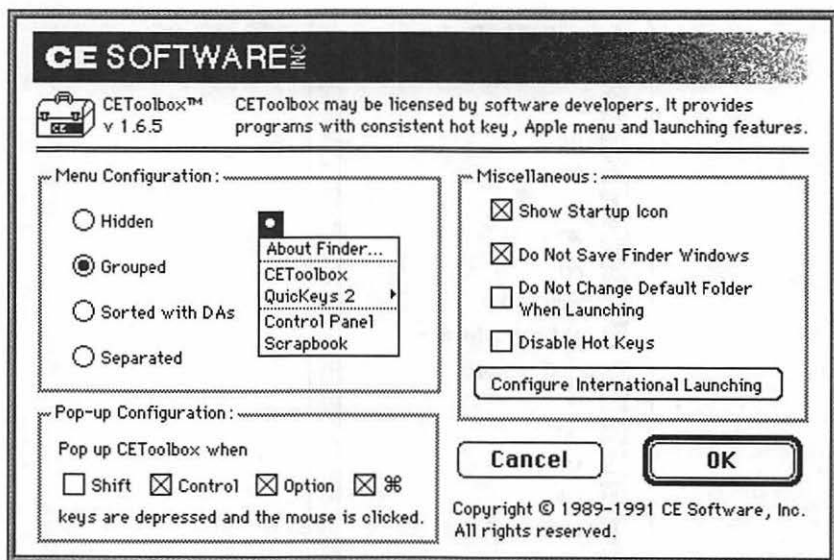


Figure 3-50: The CEToolbox dialog. Notice that menus are configured as **Grouped**.

Create the macro:

1. Open DiskTop and then open QuickKeys.
2. Find the ChangeCreator sequence and open it. If you don't have DiskTop open and you are running System 7, you will need to open the DiskTop keyset using the Open command in the File menu. If you are running System 6, you will open the DA Handler (with MultiFinder) or Finder keyset. Select all of the QuickKeys in the sequence, copy them, and close the sequence.
3. Now select Sequence from the Define menu. When the editor opens, paste the ChangeCreator QuickKeys.
4. Scroll to the top of the sequence and then move the insertion arrow before the Begin Repeat QuickKey.
5. Select Message from the Extensions menu. Enter your message. Select the checkboxes for the Show OK and Show Cancel buttons (Fig. 3-51).
6. Select Menu/DA from the Define menu and then select DiskTop Find from the Apple menu. Click OK to close the Menu editor.

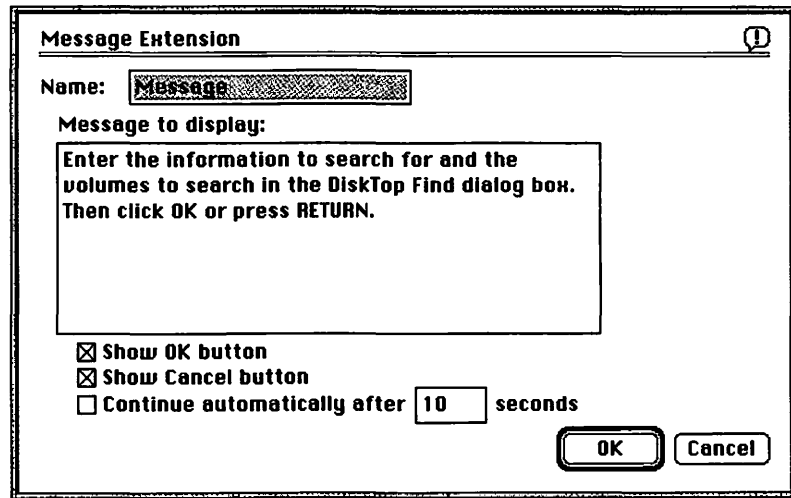


Figure 3-51: The Message extension for the ExtendedFind sequence.

7. Select Wait from the Extensions hierarchical menu. When the extension opens, select Dialog in front from the pop-up menu. When you play back the sequence, this extension pauses the sequence until Find Criteria dialog comes to the front. Because the Wait extension allows user actions, you will be able to search for files and retain them to the DiskTop window.
8. Select Wait from the Extensions menu a second time. When the extension opens, select the Not dialog from the pop-up menu. When you play back the sequence, this extension pauses the sequence until the Find Criteria dialog closes and the DiskTop

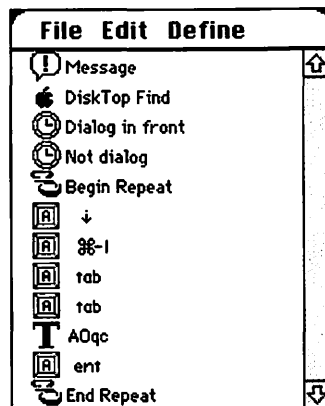


Figure 3-52: The completed ExtendedFind macro.

window comes to the front. Once you click the Retain button after you complete the search, the sequence automatically begins to run again.

Variations: Unfortunately, ResEdit does not have a built-in Find function, so you can't create a ResEdit variation on this sequence.

And Now ...

Whew! If you've read through or created each of the macros in this chapter, you've absorbed some important information and learned a trick or two. No doubt you've already thought of ways to modify the macros to work with the programs you use everyday. You may already have modified the macros to perform different tasks. In either case, you're on your way to becoming a macroing wizard.

In the next chapter, we look at some macros that you can use with your favorite word processor or text editor. If you don't use a word processor very often (is that possible?), you may want to refer to the table of contents and read a chapter about the software you use everyday.

4

CHAPTER

Text Tricks

The macros in this chapter are designed to make manipulating and managing text more automatic and less painful. They address formatting problems I've encountered and add features I've always wanted to program. This chapter begins with simple tricks you can use in any word processor or text editor. It moves on to macros that make your formatting commands more powerful and ends with macros that produce templates and boilerplate at a keystroke. Though the general focus of this chapter is on macros for word processors and text editors, you may find that some of these macros also apply to spreadsheets and desktop publishing programs. I should also point out that the majority of the macros in this chapter are built in Microsoft Word, but you should be able to create them in your word processor of choice.

General Macros

The first macros we create in this chapter add features to word processors or text editors that the developers often left out. In the case of the FindMark macro, I use QuicKeys to add a feature to Word that should have been included years ago. By the time you reach the end of this first section, you are creating some very complicated sequences. They may look intimidating to novice QuicKey users, but they are only the repetitious tasks you perform every day. When you read through the recipe for each macro, you'll see how easy they are to make.

Finding Two Spaces

As desktop publishing becomes more and more entrenched, professional and in-house publications are beginning to follow the guidelines

of typography. One of those is using only one space between sentences. If you are having a hard time retraining yourself to use only one space while you are typing, you can create a macro that checks for occurrences of two spaces in a document and changes them to one space.

Before we begin: You should be in your word processor or text editor. You would also find this macro useful in any desktop publishing program. Open QuickKeys, select the program's name from the pop-up menu, and close QuickKeys.

Create the macro:

1. Select Record Sequence from the QuickKeys menu or press the keystroke you have assigned to the Recorder On/Off special.
2. Press **⌘-Home**. This moves the insertion point back to the beginning of the document.
3. Press **⌘-H** to open Word's Replace window.
4. When the window opens, the insertion point is in the first text box. Type a period followed by two spaces. Press **Tab** and then type a period followed by one space. This causes Word to look for every occurrence of two spaces between sentences and replace it with one space. The period ensures that Word only finds and replaces the two spaces between sentences.
5. Press **⌘-R** to click the Replace All button.
6. When the dialog box appears telling you that Word has reached the end of the document (Fig. 4-1), press **Return**.
7. Press **⌘-W** to close the Replace window.

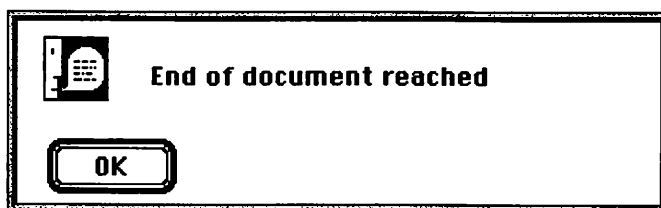


Figure 4-1: Word's End of document dialog.

8. Now select Stop Recording... from the QuicKeys menu or press the key you have assigned to the stop recording command. QuicKeys will open.
9. When the Sequence editor opens, move the insertion arrow between the `⌘-R` and the `Return` literal.
10. Select CursorWait from the Extensions menu. When the CursorWait extension opens, select Is and Arrow from the pop-up menus and then select the Check Cursor Mask checkbox (Fig. 4-2).

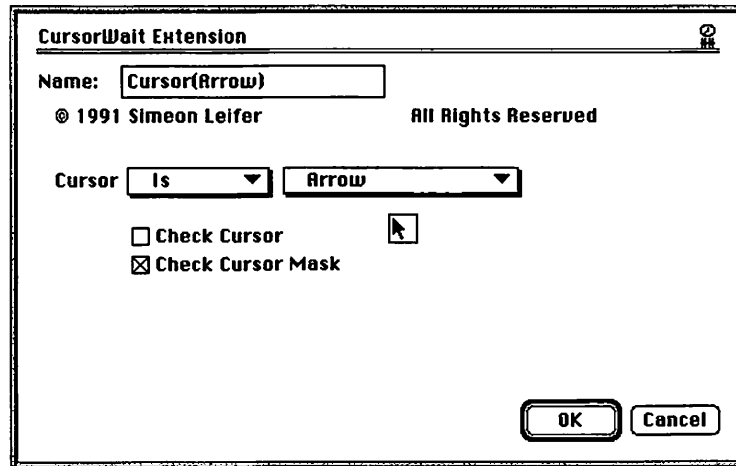


Figure 4-2: The CursorWait settings for the 2SpacesTo1 macro.

It is necessary to add this pause in the sequence; no matter how fast your Macintosh is, it will still enter the `Return` before Word finishes changing all of the spaces. CursorWait causes QuicKeys to wait until the cursor returns to an arrow. If you watch closely, you'll see this happens just before the End of document reached dialog appears.

11. Now close the CursorWait editor.
12. Name your macro and assign a keystroke. Your macro should look like Figure 4-3. When you have done this, click OK to close the Sequence editor and save your macro.

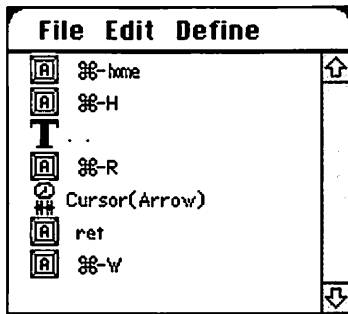


Figure 4-3: The final 2SpacesTo1 macro.

Variations: If you've ever noticed after you have printed your document that you have extra spaces between words, you can modify this macro to search for and remove those. You will need to open the text QuickKey after the **⌘-H** alias. When the Text editor opens, you'll see that it contains your settings for Word's Replace window (Fig. 4-4).

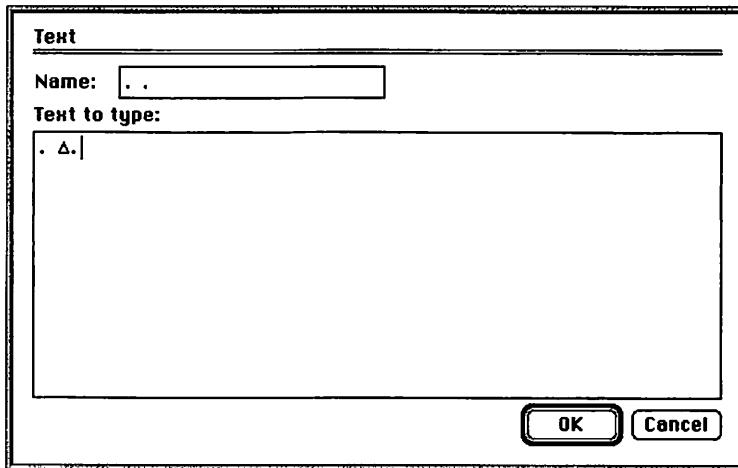


Figure 4-4: The Replace dialog settings for the 2SpacesTo1 macro.


The period followed by two spaces is the information you entered in the Find What text box. You entered the tab (the triangle symbol) to move the insertion point to the Replace With text box. To search for extra spaces, you could remove the periods leaving just the two spaces that the macro enters in the Find What text box and the single space that it enters in the Replace With text box.

NOTE: Normally, the tab is recorded as an alias, but the tab is recorded in the text QuickKey because you were entering text before you pressed it. You can remove it, create an alias for it, and create another text QuickKey for the period and space that follow, so you can see the individual steps of your macro. However, it doesn't hurt to leave the text QuickKey as it is.

Now that you see how easy it is to enter information in a program's dialog boxes or windows, let's use these techniques to add a missing feature to Microsoft Word.





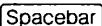

Inserting a Mark

A number of text editors and programming editors have a Set Mark feature that lets you set a marker anywhere in a document. Markers are like electronic bookmarks. They allow you to mark places in your text to which you may later want to refer. Or, you can mark two places in your document that you want to compare. Then, with a keystroke, you can jump to the marker. If, for example, you are dissatisfied with the wording of a certain paragraph and you want to come back to it later, you could place a marker in that paragraph. When you search for the marker, it takes you back to that paragraph and saves you the time it would take to search for it.

Sadly, this feature has not found its way into very many Macintosh word processors (currently, only FullWrite Professional and Nisus). You can create a macro that lets you add this function to any word processor that has a Find command. Markers should be characters that do not normally appear in your text. You could use any of the Macintosh's special characters, like  I generally combine the @ symbol with either numbers or words for my markers. This technique makes it easy to remember markers. The macro you are about to create inserts a marker in your document using a standard character and formats the marker so it is hidden. After we create this macro, we'll create one to find the marker.

Before we begin: This macro is primarily for Word since it does not have a Set Mark feature and it can format your markers as hidden text so they are not accidentally printed. I'm also using Word because I've always wanted a Set Mark feature in it. Open QuickKeys and select the program's name from the pop-up menu and close QuickKeys.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Press --X. In Word, this turns on the Hidden character format. I'm using this format so I can leave my markers in the text. If I don't want them printed, I can leave the Print Hidden Text checkbox unselected.
3. Type the @ symbol followed by two spaces. You can insert any character that is not normally included in words instead of the @ symbol. The spaces give us room to insert a word.
4. Press --X again. This turns off the Hidden character format.
5. Press  once to insert a blank space. The space allows us to treat the marker as a word. Because we've formatted the space around the marker and the word we are going to enter as hidden text, Word will not print any extra spaces.
6. Press  twice to move the cursor back to the marker symbol. Now your cursor is positioned to insert the word or number for the marker.
7. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, name your macro and assign a keystroke. It should look like the macro in Figure 4-5.
8. Click OK to close the Sequence editor and save your changes.

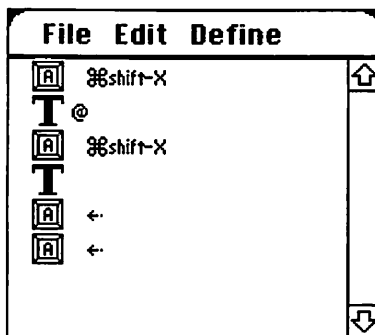


Figure 4-5: The final InsertMarker macro.

Use and variations: Since the macro inserts a marker with a space after it, you can put the marker in the middle of a paragraph without disturbing any of the formatting around it. If you don't use Word, you probably won't be able to format the marker as Hidden. In that case, you should record the macro without the two **⌘-Shift-X** aliases and the extra space after the marker. Whether or not you use Hidden formatting, you can make the marker stand out by formatting it with color or boldface. You will also need to remember to delete your markers before you print your document.

Finding a Marker

Now that you've created a macro to insert a marker, let's create a macro that finds the markers you insert. Like the one above, this macro should work in any word processor.

Before we begin: You should have the word processor or text editor you want to use open. Open QuicKeys, select the program's name from the pop-up menu, and then close QuicKeys.

While you could create a macro that quickly looks for one particular marker, you would have to create a macro for each marker that you use. Also, you would have to change the macro every time you change your markers. A more flexible solution is to create a macro that pauses at the Find dialog box and lets you enter the marker to which you want to go. If you use a standard symbol combined with a number or words, you can even have the macro insert the symbol for you. If you have your program ready, let's build the FindMarker macro.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Select Find from Word's Edit menu or press **⌘-F**.
3. Type the symbol that you used for your marker and add a space.
4. Steps 4-8 are added to accommodate Word's Find command which has an option that lets you set the search direction (Up, Down, All, or Selection). With the Find dialog box still open, press **⌘-Tab**. You'll notice a blinking gray line under the Match Whole Word Only checkbox.

5. Press **⌘-Tab** again. This time you'll see the blinking gray line under the Match Case checkbox.
6. Press **⌘-Tab** again. Now the blinking gray line is under the Search drop-down menu.
7. Press **⌘-Spacebar**. This extends the menu (see Figure 4-6).

NOTE: *SCSIProbe*, a freeware program by Robert Polic, uses **⌘-Spacebar** as the default keyboard shortcut for mounting a hard drive. If *SCSIProbe* is on your hard drive and **⌘-Spacebar** doesn't extend the menu in Word's Find window, you should change the default mounting key in *SCSIProbe*.

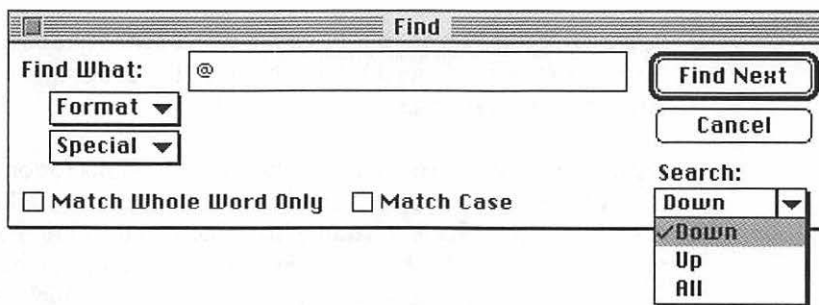


Figure 4-6: Word's extended Search menu.

8. Type an "a" to choose the All option and press **Return** to select it.

NOTE: *For your macro to work successfully every time, you must incorporate steps 4-8 in the macro. If you do not and your insertion point is below the marker, Word will not find the marker and then will display a dialog box asking if you want to search from the beginning of the document. Since the macro responds to all of the dialogs, the dialog asking if you want to continue searching is unexpected and causes the macro to fail. You could try to select the All option from the menu using a click and drag, but you will discover that the keyboard shortcuts to select the option are more reliable since they aren't dependent on screen location.*

9. Select Stop Recording... from the QuickKeys menu. Now that you have recorded most of the macro, let's tweak it just a bit.

10. When the Sequence editor opens, click the Literal button and press **[Return]**. When you play the macro, this return clicks the Find Next button in the Find dialog box and searches for the marker.
11. Select CursorWait from the Extensions menu. When the CursorWait extension opens, select Is and Arrow from the pop-up menus and select the Check Cursor Mask checkbox. Click OK to close the editor.
12. Click the Literal button again and press **[⌘]-W**. This QuicKey will close the Find dialog box after Word finds the marker.
13. Now move the insertion arrow between the text QuicKey for the marker symbol and the first **[⌘]-Tab** and select Pause from the bottom of the Define menu. When the Pause dialog box appears, enter in it the number of seconds you want to allow yourself to type the marker word or number.

You could select the Pause and wait for user button in the Pause QuicKey, but—horror of horrors—that would require you to lift your hand from the keyboard and use the mouse to unpause the macro.

14. Name your macro and assign a keystroke. When you are finished building your macro, it should look like the one in Figure 4-7. Click OK to close the Sequence editor and save your changes.

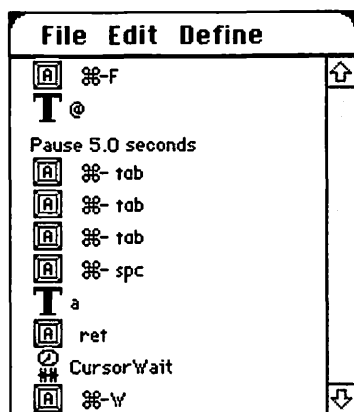






Figure 4-7: The completed FindMarker macro.

Use and variations: When you play this macro, it will open the Find dialog box and enter the @ symbol. Then it will pause for five seconds while you enter the word or number that makes up the rest of the marker. For instance, if you have entered "@ 2" anywhere in your document, you can enter "2" at the pause. When the macro resumes, it will find and highlight the marker and close the Find dialog box.

If you can't remember your markers, you can use Word 5's Find Character command to search for hidden text.

If you are comparing two parts of a Word document, you can use the Go Back command to return to your previous location. In other words, you would use the macro to find the marker and then select Go Back (--Z) to return to your previous location.

You can create a variation of this macro if your word processor's search command can search through all open documents or all documents on disk. You could also use this macro to search through all of the files until you find the marker. In Word 5, you can use the Find File command and enter your marker symbol in the Any Text text box:

1. Select Record Sequence from the QuickKeys menu.
2. Select Find File... from Word's File menu.
3. Press  twice to move to the Any Text text box.
4. Enter the symbol you have used for your marker.
5. Select the drive you want to search from the Drives menu.
6. Select Word Documents from the File Types menu. This will limit Word's search to only Word documents instead of searching all of the readable files on your drive.
7. Press  to click the OK button. Now wait until Word is through searching the document and then select Stop Recording... from the QuickKeys menu. If you have a large number of files on your drive, you might want to cancel the search before it is completed.
8. When the Sequence editor opens, select the CursorWait extension from the Extensions menu. Select Is Not and Watch from the pop-

up menus. Name the extension and click OK to close the extension editor and save your changes.

9. Your final FindMarkInFiles macro should look like Figure 4-8. Name the macro and click OK to close the Sequence editor and save your changes.

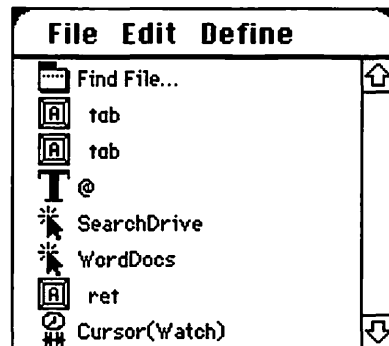


Figure 4-8: The FindMarkInFiles macro.

Now you have a macro that can search all of your documents for your marker. If you do not use Word or a program with a built-in command to search through files, you will need to use a program like the GOfer desk accessory from Microlytics to search for the marker in your word processing files.

Jumping Between Pages

Most word processors have a Go To command that allows you to enter the page number to which you want to jump. While the Go To command is faster than scrolling to the page, it still requires several keystrokes. If you find yourself returning to the same page again and again, you can save yourself those keystrokes by creating a macro that jumps directly to the page. For instance, **[Ctrl]-1** takes you to page 3 of your document, and **[Ctrl]-2** takes you to page 48. The keystrokes here are completely arbitrary, but you don't want to make them too complicated or they defeat the simplicity of the macro. This is useful if you are creating a table of contents or if you are moving information from one page to another.

Before we begin: You need to open the word processor and the document for which you want to create the macro. I'll be using Word 5.0. Open QuicKeys, select the program's name from the pop-up menu, and close QuicKeys.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Select Go To from the Edit menu or type **⌘-G**.
3. When the Go To dialog box appears, enter the page number to which you want to jump.
4. Press **Return** to close the dialog box. Word jumps to the page.
5. Select Stop Recording... from the QuicKeys menu.
6. When the Sequence editor opens, name your macro and assign a keystroke. Your completed sequence should look like the one in Figure 4-9.

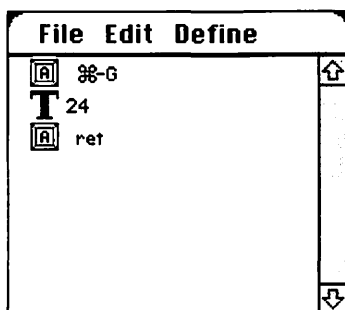


Figure 4-9: The completed GoToPage macro.

Variations: While we've only concerned ourselves with jumping between pages in a long document, you could easily change the macro to go to a page in a different document. If you work in Word, you could add the document to Word's Work menu. You can create the Work menu and add the document by typing **⌘-Option-+** and then clicking the title bar of the document when the cursor turns to a plus sign. Your steps in the macro would then be to select the name of the document from the Work menu, type **⌘-G**, enter the page number to which you want to jump, and press **Return**. By adding the document to the Work menu and selecting it from the Work menu, Word will bring it to the front—whether or not it is already open.

Combining Files

Since most word processors slow down as your files get bigger, it is a good idea to break a file into smaller chunks. For instance, you might want to divide a long report or paper into different sections and then combine the smaller files when you are through editing the document. Or, you may have a database program that generates several short reports that must be combined. Whatever the case, you can create a macro that extracts the contents of smaller files and combines them into one master file.

Before we begin: To make this macro work, you will need a program that adds bounce-back to the Macintosh's standard Open dialog box. The bounce-back feature remembers the last file you opened. When the Open dialog appears again, it scrolls to the last opened file and highlights it.

Andy Hertzfeld, one of the original Macintosh team members, wrote SF Scroll in 1988. He didn't update it when the Macintosh system software changed, so Fred Reed, the author of OnCue and its sequels, wrote Rebound to take its place. Ray Lau incorporated the bounce-back feature into the shareware program SFVollINIT and then into Shortcut. Hiro Yamamoto added the feature to Boomerang (a shareware program) and to its commercial big brother Super Boomerang. Since Apple has not incorporated bounce-back into the standard file package, you are forced to come up with one of these programs. Fred Reed has rewritten Rebound so it is compatible with System 7 and has allowed me to include it with the book. You will find it on the companion disk in the back of this book.

The very first thing you should do is move all of the files you want to combine into a folder. They should also have a number or a letter before their names so they appear in the Open dialog box in the order in which you want them combined. These two preparatory steps ensure that the macro will select your files in the order in which you want them combined and that the bounce-back feature correctly selects the next file in the list. You will also need to set up one of the Open dialog INITs/ extensions discussed above. You'll need to reboot your Macintosh after you drag one of them into the System folder. Open the program for which you want to record the macro and open a new document. This will be your master file. If you are using Word, you will need to save your master file before you begin; otherwise, Word replaces your untitled master file with the first file you open to select the text from.

Also, do not save your master file in the folder with the other files you want to combine—that would really make a mess since it would be included in the group of files the macro opens to select text from. Get the idea? Once you have all of the files you want to combine in a folder, open your word processor, select the open command, and then find the folder that contains the files in the Open dialog box. Now click the Cancel button in the Open dialog box. This sets up the macro so it defaults to the correct folder. We'll add a location QuicKey when we edit the macro so it will always default to your Combine Files folder. Now open QuicKeys and select the program's name from the pop-up menu. Close QuicKeys. We are going to record a portion of the macro that will become the basis for a repeat loop.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Select Open from the File menu of the program or press **⌘-O**. This sets up the repeat by displaying the Open dialog box.
3. Press **⇧-8**, the asterisk. This selects the first file name in the dialog box.
4. Press **⏏** to open the first file that appears in the dialog box.
5. Select the Select All command from the Edit menu or press **⌘-A** to select the contents of the file.
6. Press **⌘-C** to copy the contents of the file.
7. Press **⌘-W** to close the file. This brings your master file window to the front and improves the performance of your program and macro. A number of open windows would consume memory and slow down the program.
8. Press **⌘-V** to paste the Clipboard contents in the master file.
9. Press **⌘-O** again to display the Open dialog box. This one is necessary since the first Open will be outside of our repeat loop.
10. Press **↓** to select the next file in the dialog box. Then type **⌘-C** to close the dialog box.

11. Select Stop Recording... from the QuicKeys menu. When the Sequence editor opens, your sequence should look like the one in Figure 4-10.

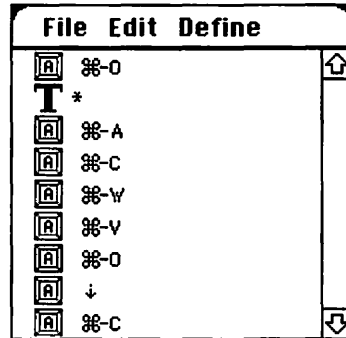


Figure 4-10: The unedited CombineFiles macro.

You'll notice that the **⌘-8** and **Return** are entered as a text QuicKey. To insert the Repeat extension, we need to convert this text QuicKey to two alias QuicKeys.

1. Select the text QuicKey then select Delete from the Edit menu.
2. Move the insertion arrow below the **⌘-O** or the Open Menu QuicKey at the top of the sequence.
3. Select Alias from the Define menu, type **⌘-8**, and then click OK to close the Alias editor.
4. Select Alias from the Define menu and press **Return** while holding down the mouse button. Then click OK to close the Alias editor.
5. We're going to leave the **⌘-C** at the bottom of the sequence and use it later.

Now that we have the sequence for the repeat, let's add a location QuicKey and beginning and ending repeat QuicKeys. We will also add some pauses to the sequence to make it run more reliably.

1. Move the insertion arrow to the top of the sequence and select Location from the Extensions menu. When the Location editor opens, select the folder into which you are putting the files you want to combine. This ensures that Word opens to the correct folder. Click OK to close the Location editor.

2. Now move the insertion arrow between the **⌘-8** and **⌘-Return** aliases.
3. Select Pause from the Define menu and enter 3 seconds. This pauses the sequence just long enough for the Open dialog to appear. Click OK to close Pause dialog box.
4. Select Repeat from the Extensions menu. When the Repeat extension opens, select the Begin Repeat radio button and the Put Up Dialog checkbox. When the macro runs, you will be prompted to enter the number of times you want the sequence to repeat. Click OK to close the Repeat editor.
5. Now move the insertion arrow between the last **⌘-O** or the Open Menu QuickKey and the **⌘-↓** aliases.
6. Select Pause from the Define menu and enter 5 seconds. This may or may not be necessary on your Macintosh. You want to prevent QuickKeys from entering the **⌘-↓** before the dialog box opens. If it does, ShortCut (or the program you are using for the bounce-back feature) loses track of the last file you opened and causes the macro to fail. Click OK to close the Pause dialog box.
7. Move the insertion arrow above the **⌘-C** at the end of the sequence and select Repeat from the Extensions menu. When the Repeat extension opens, select the End Repeat radio button. Click OK to close the Repeat editor.
8. Now we have the **⌘-C** at the end of the sequence from our earlier recording session. This closes the Open dialog box if it happens to be open when the macro finishes running.

NOTE: If you are using Super Boomerang, the **⌘-C** opens its Configure dialog box. You will either need to change the keyboard shortcut in Super Boomerang or select Copy from the Edit menu of your word processor.

9. Name your sequence and assign a keystroke to it. The final macro should look like the one in Figure 4-11.
10. Click OK to close the Sequence editor and save your changes.

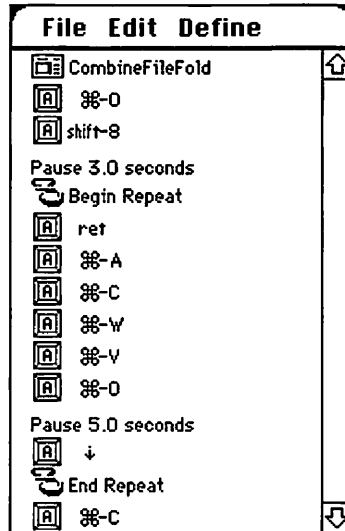


Figure 4-11: The final CombineFiles macro.

Variations: In some ways this macro is a little simplistic. It combines the files, but it doesn't attempt to make any changes to the combined master file. You could embellish the macro by having it make formatting changes or adding headers as it combines the files. For example, if you are combining all of the mail files you received on a given day, you might want it to add a line between each file, date-stamp the final document, and change the font before you send it off to the printer. This is relatively easy. We can add a step in our repeat loop that inserts the line. Then we can add the date-stamping and font changes after the loop has run and all of the files are combined.

To add the line, we will insert a step before the `⌘-V`, which pastes the text from a file into the master file. First let's set up QuickKeys to grab the line and save it as a clip:

1. Open QuickKeys and select Grab Ease from the Extensions menu.
2. When the Grab Ease editor window opens, enter a keystroke. I have assigned my Grab Ease QuickKey to `Ctrl-F3`. This is only a slight variation of the keyboard shortcut `F3`, which is Copy in most programs, and is easy for me to remember. If you do not have an extended keyboard, you should assign a keystroke close to `⌘-C` so it will be easy to remember. Click OK to close the Grab Ease editor.

Now you are ready to grab a line that you can insert as a divider between the individual files you combine. Open your favorite graphics program and draw a line about four inches long.

1. Select the line.
2. Type the keystroke you assigned to Grab Ease. A dialog box appears asking you to name the clip (Fig. 4-12).

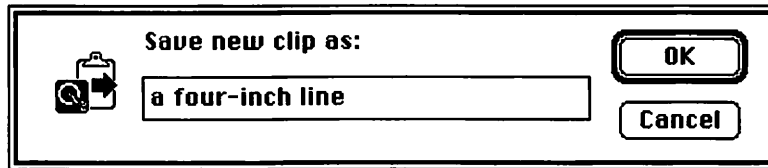


Figure 4-12: The Grab Ease dialog box prompting you to name your clip.

3. Enter a name for the clip and click OK. This saves the line as a clip in the Clipboards folder inside your QuickKeys folder.

Now you can paste the line between files as a single step. Switch back to your word processor. Open your word processor if you have closed it.

1. Open QuickKeys and open the CombineFiles sequence. This may require you to select your word processor's keyset from the pop-up menu.
2. Move the insertion point between the `⌘-V` and the `Return` aliases.
3. Select Paste Ease from the Extensions menu.
4. When the Paste Ease editor opens, select your line clip from the pop-up menu. Then click OK to close the editor (Fig. 4-13). This will place your line between each text file as it is pasted into the master file.
5. Select Alias from the Define menu. When the editor opens, hold down the mouse button and press `Return`. This will place a blank line between the line and the next file that you paste into the master file.
6. Now move the insertion arrow to the end of the sequence.

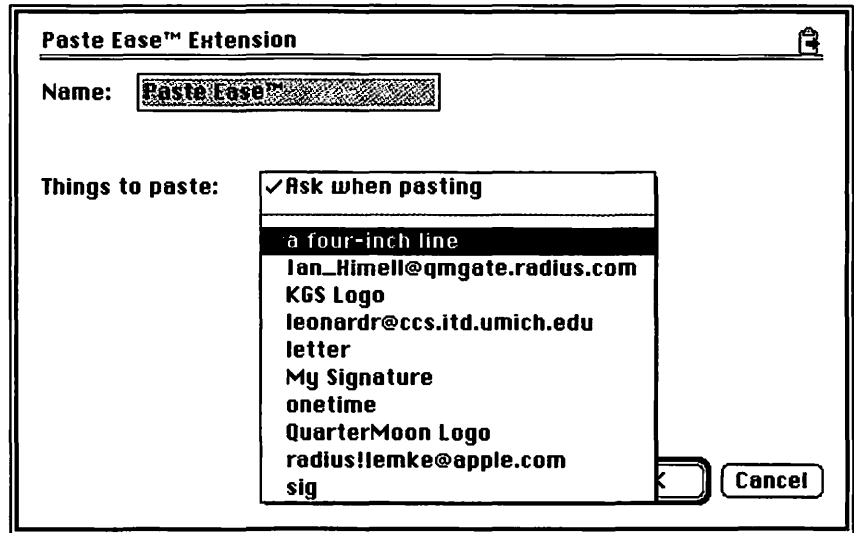


Figure 4-13: The Paste Ease editor displays a dialog box that lets you select the graphic you want to paste.

7. Select Menu/DA... from the Define menu and then select the Select All command from the Edit menu of your word processor. Click OK to close the Menu editor. This selects all of the text in your master file.
8. Select Menu/DA... from the Define menu and then select a font from the Font menu. This changes all of the text in your master file to the same font. Click OK to close the Menu editor.
9. Select Menu/DA... from the Define menu again and then select a font size from the Font menu. Click OK to close the Menu editor.
10. Select Alias from the Define menu and type `⌘-Home` when the Alias editor opens. Click OK to close it. This keystroke moves the insertion point to the beginning of the master file.
11. Select a Date format from the Date/Time menu. Click OK to close the Date/Time editor and save your changes.
12. Select Alias from the Define menu and hold down the mouse button and press `Return` whenever the editor opens. This will place a blank line between the date-stamp and your text. Click OK to close the Alias editor and save your changes. Your new Combine&Format macro should look like Figure 4-14.

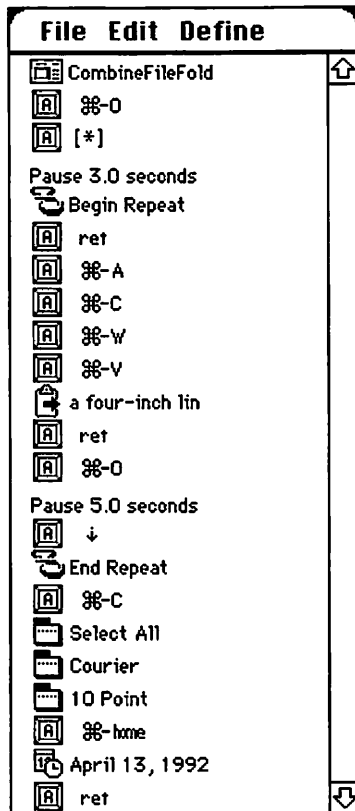


Figure 4-14: The final Combine&Format macro.

13. Click OK to close the Sequence editor and save your changes.

Now your CombineFiles macro not only combines all of the files in a folder, but it also adds a bar between the different files and formats the final master file.

Adding Graphic Markers

Because adding graphics to your word processor document can slow down scrolling and saving, it is a good idea to wait until you are through editing your document before you add the graphics. You can create a macro that inserts the figure information for each graphic and then builds a checklist in another document. The checklist is useful for making sure all of your graphics are included in your final document. The macro enters a figure template, formats the template, pauses while you enter the actual figure information, and then copies the figure information to a checklist file.

Before we begin: You should have a Figure Checklist document opened. We will copy the figure information for each graphic into it to build a checklist of the graphics you should create. We will use this document first. You will also need to open the document in which you are inserting the figure information. You can create this macro for any word processor, but I'll be using Word 5. I want to take advantage of its Windows menu and use its Style command to format the figure information quickly. You'll also need to open QuicKeys, select the program's name from the pop-up menu, and then close QuicKeys.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Type the figure information in your document. For instance, type "Figure 0-00:" as the placeholder. The first zero is the chapter number and the zeros after the dash are for the figure number.
3. If you are recording the macro in Word, press **⌘-Shift-S**. This activates the message box so you can enter the name of the style you are using for figures. I have a style called "Figure." If you are using Word, you could add the menu selections necessary to format your placeholder.

NOTE: **⌘-Shift-S** is also the keyboard shortcut for the QuickSend command in Quick Mail. If you are using Word and Quick Mail together, you will need to change this keyboard shortcut assignment in one of the two programs.

4. Press **Return** to apply the style to the placeholder.
5. Select Pause from the QuicKeys menu. Change the zeros to correspond to chapter and figure numbers. When you are through making the changes, leave the insertion point at the end of the line.
6. Select Pause from the QuicKeys menu again. QuicKeys continues recording the sequence.
7. Press **Shift-[7]** (7 on the keypad). In Word, this selects the text from the cursor to the beginning of the line. So it selects the figure information and gets ready for the macro to copy it. If you don't have a numeric keypad, you can press **⌘-Shift-←** about five times to select figure information.

8. Press **⌘-C** to copy the figure information.
9. Now select the checklist file from the Window menu. This creates a Menu/DA QuickKey to bring the checklist window to the front. You can do this with a Select Second Window or Select Second Window special, but using the Window menu is more reliable when you have a number of windows open.
10. Press **⌘-End** to move the cursor to the bottom of the document. This step adds the figures in order.
11. Press **⌘-V** to paste the figure information from the clipboard.
12. Press **Return** twice to insert a line between each figure's information.
13. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, move the insertion arrow between the text QuickKey and the **Shift-[7]** or **⌘-Shift-←**.
14. Select Pause from the Define menu. Enter the amount of time you think it will take you to change the figure information or select the Pause and wait for user button. Click OK to close the Pause dialog box.
15. Name your sequence and assign it a keystroke. Your macro should look like the one in Figure 4-15.

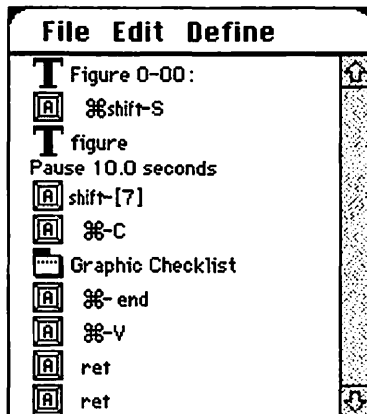


Figure 4-15: The completed GraphicList macro.

Variations: You could insert another menu QuickKey or special to bring you back to your main document after inserting the figure information in the checklist document. Depending on the way you set up your figure markers, you could also modify the macro to enter more information about your figures. If you are not using Word, you will want to add the formatting changes with the macro.

Inserting Graphics

If you walked through the macro above, you now have a document filled with figure markers showing where all your graphics need to go. This can save you a great deal of time while you are editing since it can reduce the amount of time it takes you to scroll through and save your document. However, at some point, you will need to copy the graphics into your document. Remember the document you created that is a checklist of your figures? You can use it as a poor man's graphic database and paste each graphic into it above the figure information. When the time comes to move the graphics into your document, you can use a macro that searches for the figure markers in your document, finds the appropriate graphic in the database, and then moves it from the database into the document. While this may not be the most elegant solution in the world, it can save you a great deal of time cutting and pasting. Best of all, it's automatic; the macro can run while you walk the dog.

Before we begin: If your program does not support pasting into the Find dialog box, you will need to install the Escapade control panel and reboot your Macintosh before you record this macro. Escapade is a shareware program written by Chris Wysocki and is on the disk included with this book. In addition to allowing you to paste text in dialog boxes, it also allows you to select the buttons and checkboxes in dialog boxes using your keyboard. You should have only the document and the graphics checklist open from the previous example. I have set up this macro to use the checklist as the basis for searching for the figure markers in the document. My graphic checklist document alternates between graphics and figure markers without any blank lines between them (Fig. 4-16). Notice the graphic appears above the figure information.

Because it is a regular pattern, it is easy to build a macro that works every time. You will also need two or three blank lines above the figure marker in your main document so there will be enough room for the graphic. If you have your program running and the two documents open, you are ready to record the macro. Open QuickKeys, select the program's name from the pop-up menu, and close QuickKeys. We'll start our recording in the checklist document.

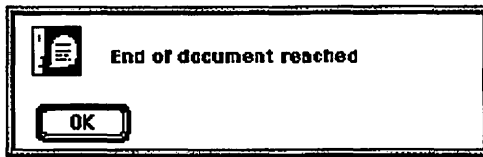


Figure 4-1:

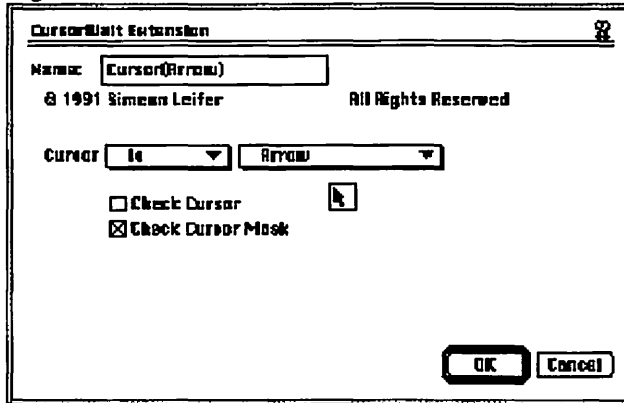


Figure 4-2:

Figure 4-16: The format of the graphics checklist with figures inserted.


Create the macro:





1. Place the insertion point at the beginning of the two documents and make certain that the checklist is the top window.
2. Move the insertion point after the colons in the first figure of your checklist.
3. Select Record Sequence from the QuickKeys menu.
4. Press **Shift-[7]** (7 on the keypad) to select to the beginning of the line. This is Word's command for copying to the beginning of a line. You will need to use your word processor's command to select to the beginning of the line. If you copy from the beginning of the line instead of the end, Word copies the paragraph marker, and that causes the Find command in step 7 to fail.
5. Press **⌘-C** to copy the figure information. This places something like "Figure 4-1:" on the clipboard.
6. Press **⌘-Option-W** to bring the main document window to the front.

7. Press **⌘-F** to open the Find window.
8. Press **⌘-V** to paste the figure information into the Find window.
9. Press **Return** to search for the figure information.
10. When Word finds the figure information, press **⌘-W** to close the Find window. You'll notice that the first figure marker in your document is highlighted.
11. Press **[7]** three times to move the insertion point two lines above the figure marker.
12. Press **⌘-Option-W** to bring the graphics checklist to the front. Notice that the first figure marker is still highlighted.
13. Press **[7]** to move the insertion point to the beginning of the line containing the figure information.
14. Press **⌘-[7]** to move to the end of the previous line.
15. Press **⌘-Shift-[7]** to select the graphic.
16. Press **⌘-C** to copy to the graphic.
17. Press **⌘-Option-W** to switch back to the main document.
18. Press **⌘-V** to paste the graphic. Congratulations, you have just automated a very repetitive task.
19. Press **⌘-Option-W** to switch back to the checklist. This sets us up to get the next figure marker and start the process all over again.
20. Notice that the first graphic in the graphic checklist is still selected from our previous copy. We need to move the insertion point after the colon of the next figure marker. Press **[1]** to deselect the graphic and move the insertion point to the end of the line.
21. Press **[1]** to move the insertion point after the colon of the first figure marker.
22. Press **[1]** to move to the insertion point after the next graphic.
23. Press **[1]** to move the insertion point after the colon of the second figure marker.

24. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, we'll need to make a few additions.

We're going to add the repeat information to the macro without testing it first. Normally, you should practice building and testing a macro as modules (refer to Chapter 2). This makes it easier to find problems in the macro before you add the next layer of complexity. While the Repeat extension alone is not another module, it does add another layer of complexity to the macro. By testing the macro first, you don't have to worry about interrupting it when it misbehaves.

TIP: Remember, when a macro is playing, you can interrupt it any time by clicking the mouse button or typing - (period).

25. Move the insertion arrow to the top of the sequence and select Repeat from the Extensions menu again. Click the Begin Repeat radio button and the Put Up Dialog checkbox. Click OK.
26. After the first -Option-W in the sequence, add a Pause of about 3 seconds. This pauses the macro until the main document window comes to the front.
27. Add a CursorWait three steps later—between the  and the -W. Select Is and the Arrow from the pop-up menus and then select the Check Cursor Mask checkbox. This pauses the sequence while Word searches for the figure marker. If you don't insert this CursorWait, the sequence may continue to run before Word completes the search.
28. After the second -Option-W in the sequence, add another Pause of 3 or 5 seconds. Again, this pauses the macro while the windows change.
29. Now move the insertion point to the bottom of the sequence, select Repeat from the Extensions menu, and click the End Repeat button. Your final macro should look like the one in Figure 4-17.
30. Name the sequence and assign it a keystroke. Click OK to close the Sequence editor and then click OK again to close QuickKeys and save your changes.

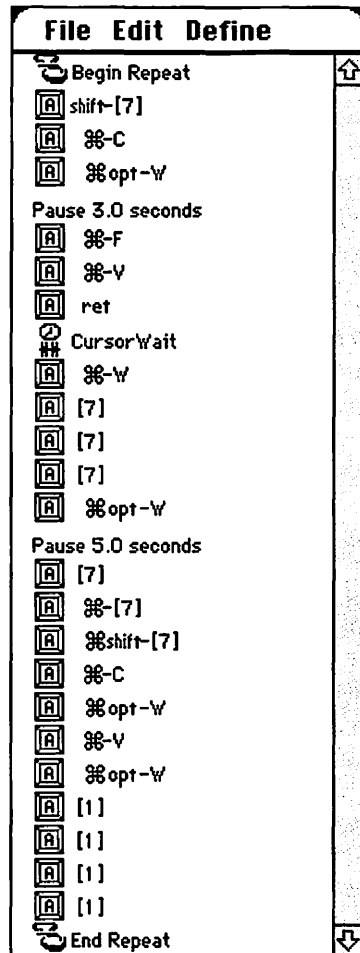


Figure 4-17: The completed InsertGraphic macro.

Variations: While this macro is recorded in Word, you'll discover that it runs smoother in MacWrite II. MacWrite II automatically wraps the search back to the beginning whenever it reaches the end of a document. Since Word asks you if you want to begin searching from the beginning of the document when it comes to the end, you must begin the macro with the insertion point at the top of the main document. You can sidestep this problem by selecting All from the Search menu the first time you open the Find dialog during the macro. You can take a cue from the FindMark macro we created earlier. You can press **⌘-Tab** or **⌘-Shift-Tab** to move through the menus and checkboxes in the dialog box until you reach the search menu, type **⌘-Spacebar** to drop the menu, and then press an "A" and **Return** to select All.

More Powerful Formatting Commands

Most word processors have shortcuts that let you change the formatting with a single keystroke. Word, Nisus, and FullWrite Professional have a Style feature that allows you to make a number of formatting changes to a paragraph with one selection. While the feature is very powerful, styles only apply to paragraphs; often you cannot format a line or a character with a style. You can overcome this limitation with QuickKeys by creating macros that select and format units of text smaller than a paragraph. For word processors that do not support styles or a similar feature, QuickKeys lets you combine several menu selections and keyboard shortcuts to format a character, line, paragraph, or document automatically.

General Formatting

At the bare minimum, a word processor has keyboard shortcuts to change the style of the font you are using. Word, for instance, lets you format selected text to boldface by typing **⌘-B** or to italic by typing **⌘-I**. You can also set justification by typing **⌘-Shift-C** to center the text or by typing **⌘-Shift-L** to left-justify text. More often than not, you can make several formatting changes at once. If you find yourself using two or three of these commands at the same time, you can make your word processor's formatting more powerful by creating a macro to make the changes for you in a single keystroke. If, for example, you created a glossary for a handbook, you might use bold and italic to make the term you are defining stand out from the definition. The several steps necessary to select the term and apply boldface and italic formatting can be reduced to a single keystroke.

Before we begin: Open a word processor (guess which one I'm going to use). Type in a word and a brief definition and separate them with a period and a space, like so: "bagatelle. a trifle." Press **Return** and add another word and definition below the first one. Open QuickKeys, check to see that you are recording the macro in the word processor's keyset, and then close QuickKeys.

Create the macro:

1. Move the cursor to the beginning of the line.
2. Select Record Sequence from the QuickKeys menu.

3. Press **⌘-Shift-[1]**. If you are using Word, this selects from the beginning of the line through the period. You could just select the first word on the line, but the macro won't work properly if you have a term that is actually two words (disk drive, for instance).
4. Press **⌘-Shift-B**. This sets the selection's style to boldface.
5. Press **⌘-Shift-I**. This sets the selection's style to italic.
6. Press **⌘-[2]**. This moves the insertion point to the next paragraph. In the case of a glossary, the next paragraph is generally the next term. So the insertion point is positioned for you to play the macro again.
7. Select Stop Recording... from QuickKeys menu. When the Sequence editor opens, your macro should look like the one in Figure 4-18. If you needed to format a whole glossary, all you would need to do is add a Repeat Begin at the beginning of the sequence and a Repeat End at the end of the sequence. Name your macro and assign a keystroke.

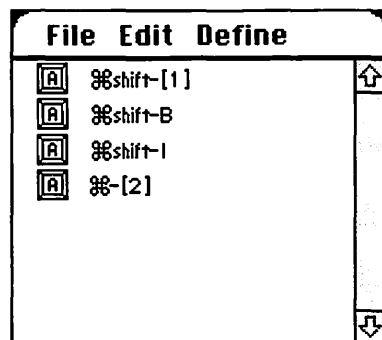


Figure 4-18: The FormatAWord macro.

Variations: The variations on this type of macro are endless. If you needed to add a Zapf Dingbats bullet, you could create a macro that changes the font to Zapf Dingbats, types the bullet character, and then resets the font. Whenever you make two or more formatting changes that require several keystrokes, menu choices, or button clicks, you should make a macro. If you really had to format a glossary that was 100 pages long, the two minutes or so it takes you to create the macro can save you an hour or more of tedium. And, if you make the macro repeat, you can go out for an espresso.

Boldfacing a Paragraph's First Line

As we discussed above, some programs allow you to create styles to format paragraphs. However, these programs cannot always apply the styles on a line-by-line, word-by-word, or character-by-character basis. In the previous macro, we changed the style of a single word, but you can extend this to larger units. You might want to bold the first line of your paragraph for emphasis.

Before we begin: Open your word processor and a document. I'll be using Word 5.0 so I can easily select the text to the end of the line. Open QuicKeys, select the word processor's name from the pop-up menu, then close QuicKeys. Now move the insertion point to the beginning of the first line of a paragraph.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Press **[Shift]-[1]** (1 on the keypad) to select the text to the end of the line.
3. Press **[⌘]-[Shift]-B** if you are using a LaserWriter or TrueType font. If you are using an Adobe font, you will need to select the font from the Font menu.

NOTE: *If you are using an Adobe font and press **[⌘]-[Shift]-B**, Word will not download the bold typeface (Utopia Bold, for instance) to the LaserWriter. Instead, it will use the Adobe font and generate a bold typeface from it—a very subtle difference.*

4. Select Stop Recording... from the QuicKeys menu.
5. When the Sequence editor opens, name your sequence and assign a keystroke (Fig. 4-19).

Of course, not all word processors have the capability of applying several formatting commands to a paragraph with one command. QuicKeys lets you create styles in those word processors.

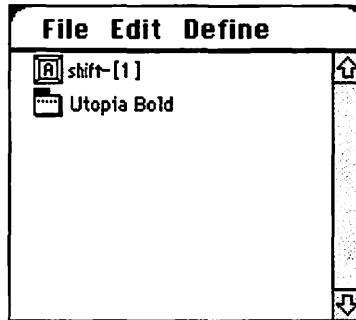


Figure 4-19: The Bold1stLine macro for Microsoft Word.

Creating Styles

Few other computer-related tasks can benefit as greatly from automation as text formatting. Some programs are already geared for automating tasks. Spreadsheets, for instance, have Fill Down and Fill Right commands to automate the process of entering the same formulas across cells. Style commands move word processors in that direction, but even the Styles command can be automated. For word processors that do not have a Style feature, you can create macros that automatically format the selected text.

Before we begin: Open the word processor for which you want to create the macro. I'll be using WriteNow 2.2 since it doesn't have a Style feature, though you could use MacWrite or any word processor. In some cases, you might even find it easier and more convenient to create styles using QuickKeys macros than using a word processor's built-in Style feature. Open QuickKeys, select the program's name from the pop-up menu, and close QuickKeys.

Create the macro:

1. Select the text that you want to format.
2. Select Record Sequence from the QuickKeys menu.
3. Select Bold from Style menu or press **⌘-B**.
4. Select 14 Point from the Size menu.
5. Select a font from the Font menu.
6. Select Stop Recording... from the QuickKeys menu.

7. When the Sequence editor opens, name the macro and assign a keystroke. You may want to give it the name of the style it creates. Your macro should look like Figure 4-20.
8. Click OK to close the Sequence editor and save your changes.

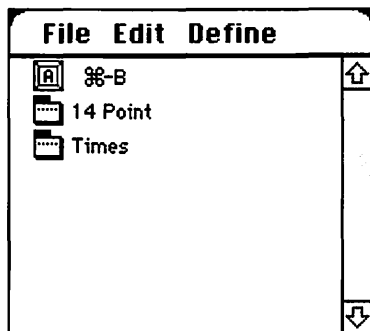


Figure 4-20: The completed WriteNow style macro.

Variations: While Word has a Styles feature and you can apply a style to your text by typing the style's name in the message box, you cannot apply them with a *single* keystroke. For instance, you can press **⌘-Shift-S** and type the name of the style in the message box or you can press **⌘-T** and select the style from the Style window with the **↑** and **↓** keys. You can also select the style from the pop-up menu in the Ruler or you can add the style to the Work menu and select it from there. However, you can be much more efficient if you create a macro to apply the style using one of the methods above. The best method to use is typing the name of the style in the message box. It isn't a good idea to create a macro for selecting the style from the ruler's pop-up menu since you might not have the ruler displayed. A macro to select the style from the Style window using the **↑** and **↓** keys might also fail when you add new styles. Perform the following steps to create a macro to type the styles name in Word's message box:

1. Select Record Sequence from the QuickKeys menu.
2. Press **⌘-Shift-S** to activate Word's message box at the bottom of the document window (Fig. 4-21).
3. Type the name of the style. I have a style called "Heading1" that I'm using in my example.
4. Press **Return** to apply it to the paragraph.



Figure 4-21: Word's message box activated so you can enter the name of the style. Note that the name of the current style appears next to it.

5. Select Stop Recording... from the QuickKeys menu.
6. When the Sequence editor opens, name your macro and assign a keystroke. Your macro should look like Figure 4-22.
7. Click OK to close the Sequence editor and save your changes.

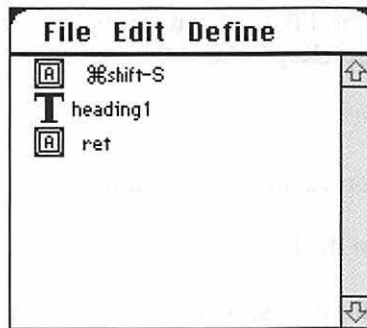


Figure 4-22: The completed macro to select one of Word's styles.

Now you can change styles in Word with a single keystroke instead of making a menu selection. You can create macros for every style and save yourself the time it takes to enter the name of the style in the message box.

Creating Templates

Several Macintosh word processors include macros that are actually very complete templates. That is, the macros create a document with a predefined set of contents designed for some specific purpose or task—like a business report or letter. WordPerfect has macros to open a new document that already has your inside address, the date, a slot for the recipient's address, a greeting, a slot for the body, and a closing. This template is properly formatted and the recipient's address may even be formatted for mail merge. If you find yourself setting up a document with the same margins, the same contents, and the same formatting, you should create a macro to set up the document at a keystroke.

Letter Template

It is very easy to create a macro that automatically inserts a business letter template with a keystroke. At its most basic, it is a series of text QuickKeys and `[Return]` keys. You can insert your address, the current date, the recipient's address, a salutation, a placeholder for the body of your letter, and a closing. If you have scanned your letterhead, you can even paste it in. Your macro can also format the letter with the appropriate fonts and styles.

Before we begin: You should be in the program you want to use. Open QuickKeys and select your word processor from the pop-up menu. In general, you should decide on the contents of the template in advance. You should know where you are going to add formatting and how you are positioning your paragraphs. So you can see how easy it is to build a macro in the Sequence editor, I'll walk you through creating the sequence instead of letting QuickKeys record the sequence.

Create the macro:

1. Open QuickKeys if you've already closed it.
2. Select Sequence from the Define menu.
3. When Sequence editor opens, click the Literal button and press `[Return]`. A `[Return]` literal is added to the Sequence dialog. Enter several of these to add space to the top of your letter.
4. Now select Text from the Define menu. When the Text editor opens, enter your address followed by a `[Return]`. Close the Text editor and add another `[Return]` literal.
5. Select Date/Time from the Define menu. When the Date/Time editor opens, select a Date format from the pop-up menu. Close the Date/Time editor and add two `[Return]` literals.
6. Select Text from the Define menu. Enter "Business address" in the window. This serves as a place marker for the actual address that you will enter from the keyboard or paste from the clipboard. Close the Text editor and add a `[Return]` literal.
7. Select Text from the Define menu again. Enter "Dear name:" in the window followed by a `[Return]`. As in the previous step, the word "name" only serves as a place marker. You will enter the

person's name and change the colon to something less formal—if necessary—when you begin writing your letter. Close the Text editor and add a **Return** literal.

8. Select Text from the Define menu and enter "letter body" in the window followed by a **Return**. This is the place marker for the body of the letter. Close the Text editor and add a **Return** literal.
9. Select Text from the Define menu and enter "Sincerely," in the window followed by a **Return**. Close the Text editor and add another **Return** literal.
10. Select Text from the Define menu and enter your name followed by a **Return**. Close the editor, enter LetterTemplate for the name of the macro, and assign a keystroke. Your completed macro should look similar to the variation in Figure 4-23.

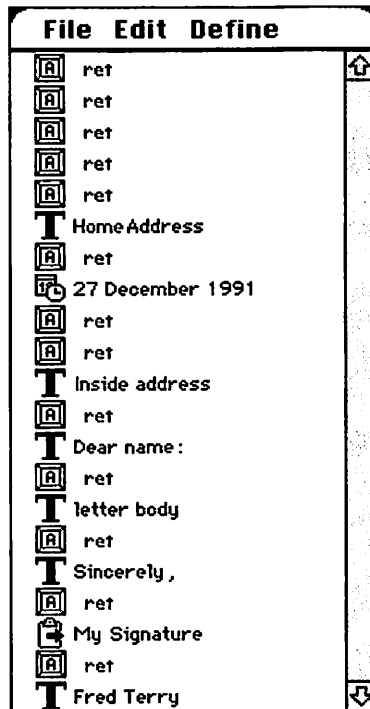


Figure 4-23: The completed LetterTemplate macro.

Variations: I fax a number of letters and have scanned my signature just for this purpose. Some desktop publishing services will do this for you if you don't own a scanner. Global Village, the company that sells the Teleport Fax Modem, will scan your signature at no charge when you buy its modem. I have saved my signature as a Clip object so that I can insert it whenever I need to use the Paste Ease extension. Thus, I can add a step (between steps 9 and 10 above) that automatically inserts my signature. If you have created your company's logo in a graphics program or you have a scanned version, the template could also insert it when setting up the letter template.

You might want to create several macros with each one setting up a different portion of your template. For instance, you can break up your letter template so that the header information (inside address and date) is inserted with one macro, and the business address is inserted with another. This will make it possible for you to have two different versions of the business address—one that lets you paste the address from the Clipboard, and another that is formatted for mail merge.

In addition to creating templates for every possible business form and boilerplate, you can make any number of variations on this macro for every conceivable task of every day life. Templates are the most helpful when you enter the same text repeatedly. Besides saving you time by typing the text for you, template macros eliminate typos. If you are a programmer, you can make templates for your text or program editor. For instance, the macro can insert a main program template and correctly position the insertion point for you to begin entering your code. Depending upon the capabilities of the editor, you may even be able to add the appropriate tabs and spaces for the template.

Creating a Bibliography Template

Another variation on the template is the bibliography. You can build underlining, indenting, and other formatting into the macro so you don't have to remember the bibliography style. When you play the macro, it sets up the format and inserts a sample bibliography entry. You only have to double-click the different parts of the template and type your reference information.

Before we begin: You should have your word processor open. I'll be using Word 5. Open QuickKeys, select Microsoft Word from the pop-up menu, and then close QuickKeys.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Let's begin by entering the author's name. Type "lastname, firstname." Type two spaces.
3. Press **⌘-Shift-U**. This toggles underlining on.
4. Enter the book information. Type "title."
5. Press **⌘-Shift-U** again. This toggles underlining off. Type two spaces.
6. Now we'll enter the city and publisher information. Type "city: publisher, year."
7. Press **[7]** (7 on the keypad) to move the cursor back to the beginning of the line.
8. Press **⌘-Shift-↓** to select "lastname" in the line.
9. Press **⌘-M** to open Word's Paragraph dialog box (Fig. 4-24). Now you are ready to set up the bibliography template's hanging indent format.

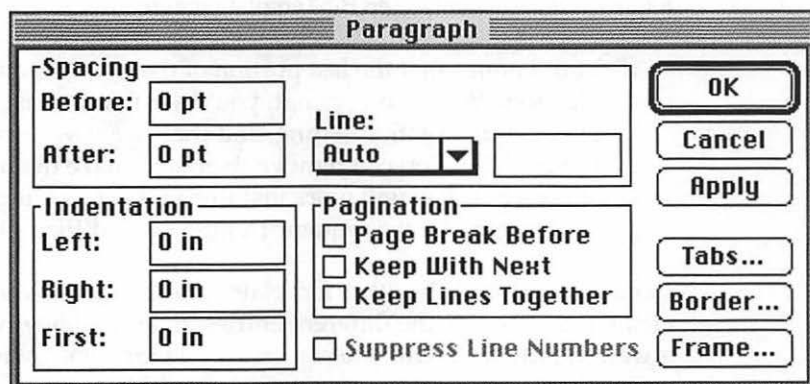


Figure 4-24: Word's Paragraph settings dialog box.

10. Press **Tab** three times. This moves the cursor to the Left Indentation text box.

11. Type “.25” and press the **Tab** twice to move to the First Indentation text box.
12. Type “-.25” in it.
13. Press **Return** to close the Paragraph dialog.
14. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, your macro should look like the one in Figure 4-25.

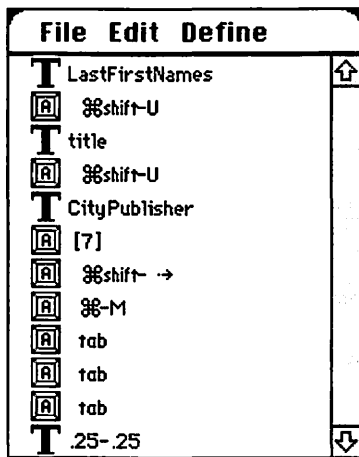


Figure 4-25: The completed BibTemplate macro.

15. You'll notice that the last portion of the macro is only a text QuickKey. When you open it, you'll see that the last two **Tab** keys, the First Indentation setting, and the **Return** were incorporated in it (Fig. 4-26). You can remove them and make them separate aliases, but they will work just fine in one text QuickKey. They do, however, make the sequence a little more difficult to read.

Variations: Some of the other templates in this vein are obvious. You can create templates for the different entries in your bibliography. Since they are all slightly different, you can create one for a book with more than one author, a collection of essays, a book with more than one volume, a book with more than one editor, or an article in a journal or magazine. While the template we created above is according to the Modern Language Association, you can create templates for different associations. You can create bibliography templates for medical or scientific journals.

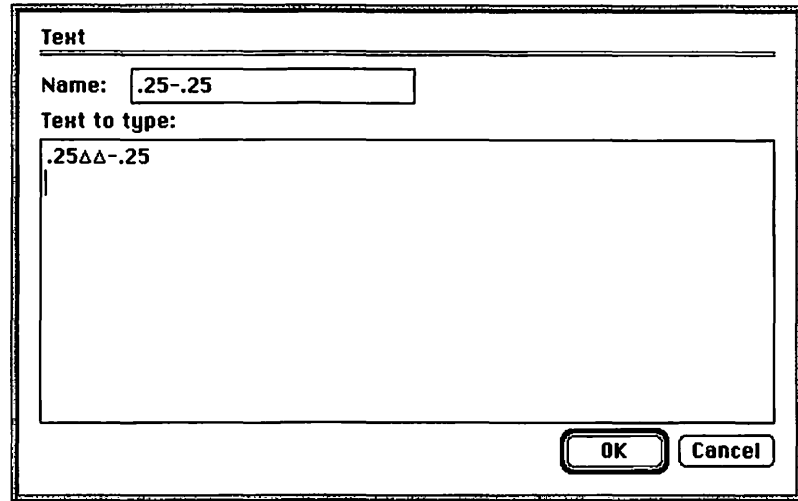


Figure 4-26: The text QuickKey at the end of the BibTemplate macro.

Print Letter and Envelope

If you are lucky enough to have one of the personal laser printers or have access to a high-end laser printer in your office, you can save yourself some time addressing envelopes by hand. This macro prints your letter then sets up a document in your word processor, pastes in the address, and prints the envelope. The real benefit of this macro is that you don't have to buy one of those fancy envelope printing programs while still getting good-looking, laser-printed envelopes.

Since this macro depends heavily on the names of menu items and the layout of dialog boxes, you will have to change those parts of the macro if you do not use Microsoft Word. You'll find a MacWrite II version of the macro on the companion disk. Since you are using a word processor to place the envelope information properly, you will also find different versions of this macro to support different printers. One version supports printers with center-feed guides, like the LaserWriter II series and the GCC PLP and BLP printers. The other version supports printers with side-feed guides, like the LaserWriter and LaserWriter Plus printers.

Before we begin: We'll create the macro for the LaserWriter II series printers. You should be in your word processor; I'll be using Word 5.0. If you are also using Word, be certain that you are in Normal view and not Page Layout view when you begin recording this macro and that the preferences are not set to open Word windows in Page Layout view.

You should also copy an address to the Clipboard so it will be ready to paste. Open QuickKeys, select the program's name from the pop-up menu, and close QuickKeys.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Select New from the File menu or press **⌘-N**.
3. Select Paste from the Edit menu or press **⌘-V**.
4. Select All from the Edit menu or press **⌘-A**.
5. Select Paragraph from the Format menu. When the dialog box opens, press **Tab** three times to move to the Left indentation text box. Type 3.5 in it and press **Return** to close the dialog box and save your changes.
6. Select Document from the Format menu. When the dialog box opens, type .50 in the Left text box, .25 in the Right text box, 4.50 in the Top text box, and .25 in the Bottom text box. The macro will be more accurate if you tab to each of the boxes. Press **Return** to close the dialog box.
7. Select Section from the Format menu. When the dialog box opens, press **Tab** twice to get to the From Top text box and type 2.5. Then press **Tab** to move to the From Bottom text box and type .25. In the next two steps, we'll set up the return address in a header. These two measurements position the header on the envelope.
8. Select Header from the Insert menu. When the header window opens, type your home address.
9. Click the close box of the header window or press **⌘-W**.
10. Select Page Setup from the File menu. When the dialog box opens, click the Landscape orientation icon and press **Return**.
11. Select Page Layout from the View menu. This adds a visual check to the macro so we can see before the envelope prints if it is set up correctly.
12. Select Stop Recording from the QuickKeys menu.

We aren't through with the macro yet. We are going to add some flourishes. When the Sequence editor opens, you'll notice that the insertion arrow is already at the bottom of the sequence.

1. Select Sound from the Extensions menu. When the Sound extension opens, select a sound that will get your attention. I have a buzzer sound that I'll insert here.
2. Click OK or press **Return** to close the extension.
3. Select Message from the Extensions menu. When the Message extension opens, enter a message for yourself in the text box (Fig. 4-27). Select the Show OK and Show Cancel button checkboxes. Click OK or press **Return** to close the extension dialog.

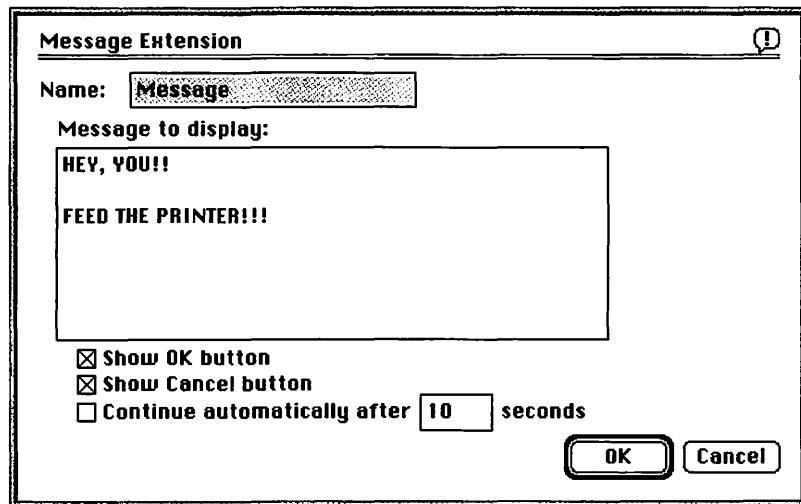


Figure 4-27: The Message extension settings for II/NTEnvelope macro.

4. Select Menu/DA... from the Define menu. When you are prompted to select a menu, choose Print from the File menu of your word processor. Click OK to close the Menu editor.
5. Select Alias from the Define menu. When the Alias editor opens, press **Return** while holding down the mouse button. Click OK to close the Alias editor.
6. The final II/NTEnvelope macro should look like Figure 4-28.
7. Click OK to close the Sequence editor and save your changes.

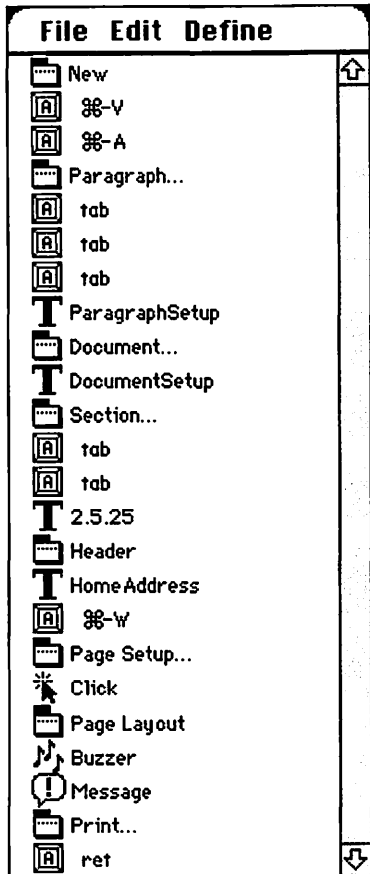


Figure 4-28: The completed II/NTEnvelope macro. You'll also find a LW/+Envelope macro on the disk for the LaserWriter and LaserWriter Plus.

Now you have a pretty sophisticated envelope macro. When you copy the address from your letter and invoke the macro, it opens a new window, pastes in the address, and sets up all of the margins. It even reminds you to insert an envelope in the printer before it selects the print command.

Variations: If you have names and addresses in a database program and it can print envelopes, you may want to change the macro so the database prints the envelopes. If you own Dynodex or a similar product, the macro is much simpler. If you own Dynodex, you must use the program and not the desk accessory for this macro because the desk accessory cannot search for full names. I have also selected the layout for a number 10 envelope (Env 10) prior to recording the macro.

1. Open QuickKeys and select Universal from the pop-up menu, then click OK to close QuickKeys. We'll record this macro as a Universal macro so it will be available from any program.
2. Select Record Sequence from the QuickKeys menu.
3. Press **⌘-C** to copy the address.
4. Open Dynodex. If you are using System 7, you can open Dynodex with a Finder Event. If you have the Dynodex desk accessory installed under System 6.0x or you have added an alias for the Dynodex program under System 7, you can open Dynodex using a menu QuickKey. I've installed a Dynodex alias, so I'll select Dynodex from the Apple menu.
5. Press **⌘-F** to open the Find dialog box.
6. Press **F4** to insert the name in the dialog box. Remember, if you don't have Escapade installed, you won't be able to do this.
7. Press **Return** to search your database for the name.
8. Select Print One from the File menu.
9. Press **Return** when the Page Setup dialog appears.
10. Press **Return** again when the Print dialog appears.
11. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, name the macro and assign a keystroke. If you have recorded the macro for Dynodex, it should look like the one in Figure 4-29.

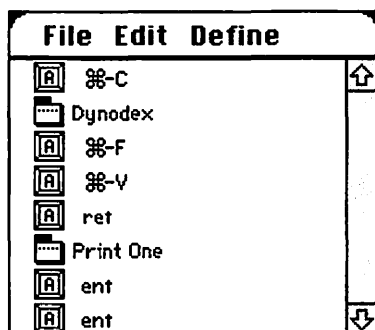


Figure 4-29: The final DynoPrintOne macro.

Now you have a simpler macro than the template above. It opens Dynodex and looks up an address, selects the Print One command from the File menu, and prints the envelope. You could add a print command for Word at the top of the sequence to print your letter before switching to Dynodex to print the envelope. If you do this, don't forget to add either a Pause QuicKey or WindowAction extensions to pause while Word prints the letter.

Moving On

Now that you have a solid collection of macros for your word processor, you are probably ready to create macros for the other programs you use on a regular basis. In the next chapter, we look at some macros for your spreadsheet program that make formatting individual cells and whole worksheets a breeze. You will also find macros for automatically entering equations and generating graphs.

5

CHAPTER

Macros for Spreadsheets

You may ask, “Why do I need QuicKeys for my spreadsheet?” The three popular spreadsheets for the Macintosh—Microsoft Excel, Lotus 1-2-3, and Claris Resolve—all have built-in macroing languages. Those languages, no matter how powerful, only work within the spreadsheet programs and cannot be extended to other programs. And, there are other reasons you might prefer to use QuicKeys over a built-in macroing language.

Some users may find the built-in languages daunting or limiting. QuicKeys allows you to more quickly develop your macros without having to learn a new way of doing things. QuicKeys also allows you to alter the program’s interface by adding or changing keyboard shortcuts; the built-in macroing languages cannot do that. Or, you might want to use QuicKeys to automatically enter data in your document. QuicKeys can also augment the spreadsheet’s macroing language. You can combine the spreadsheet’s built-in language with QuicKeys to make the spreadsheet program an even more powerful pair.

As in the previous chapters, I’ll create the example macros in one spreadsheet program. Where it is appropriate, you will also find versions of the macros for the other two programs on the enclosed companion disk.

Interface Changes

Some of the macros you will want to create are for interface changes similar to the ones we talked about in Chapters 3 and 4. QuicKeys lets you add keyboard shortcuts where a program doesn’t have any, and you can modify the keystrokes already assigned in a program.

Adding Keyboard Shortcuts

Most spreadsheet programs have a large number of keyboard shortcuts for the functions in the programs as well as the menu items. No matter how many keyboard shortcuts are already in a program, I have always found that I don't have a shortcut for the one command I use frequently. For instance, Lotus 1-2-3 has a number of keyboard shortcuts for its commands, but it does not have a shortcut to select the Split... command which lets you split the worksheet window into multiple panes.

Before we begin: I'll be creating this macro in Lotus 1-2-3, but you can create it in any spreadsheet program that contains a command for which you want a keyboard shortcut. Open the spreadsheet program, then open QuicKeys, and select the program's name from the pop-up menu. Click OK to close QuicKeys.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Select Split... from the movable Window menu. The Window Split window appears (Fig. 5-1).
3. Click the Horizontal radio button.

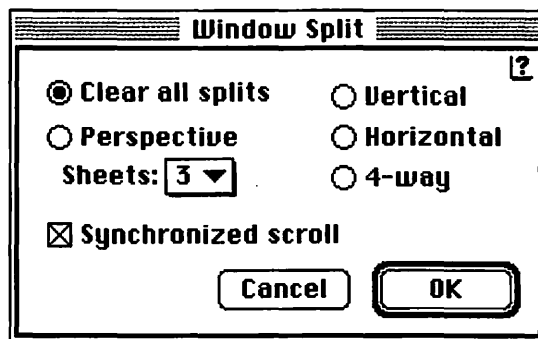


Figure 5-1: The Window Split window in Lotus 1-2-3.

4. Press **Return** to close the window.
5. Select Stop Recording... from the QuicKeys menu. When the Sequence editor opens, your macro should be similar to Figure 5-2.

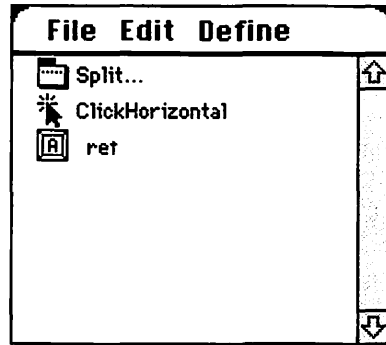


Figure 5-2: The completed SplitHorizontal macro. I've changed the click QuickKey's name to be more readable.

6. Name your macro and assign a keystroke. Click OK to close the Sequence editor and then click OK to close QuickKeys and save your changes.

Variation: You could also create a companion macro that splits the worksheet window vertically. It is essentially the same as the SplitHorizontal macro, but you click the Vertical button in the Split Window dialog box instead. You don't have to create macros for the Clear all splits or the Perspective commands since 1-2-3 already has keyboard shortcuts for them (⌘ -Option-C and ⌘ -Option-P, respectively).

There is no limit to the number of spreadsheet commands for which you could create macros. If you tend to use the keyboard for most of your work, you may want to create a macro every time you find yourself reaching for your mouse. Whenever possible, you should extend the macros to do more than just select a menu item. As in the example above, we could have stopped with a macro that opened the Split Window dialog box, but it is more efficient to add the rest of the settings to the macro.

In addition to creating macros that select menu commands, you could also create macros to activate the commands in the Excel, Resolve, and 1-2-3 tool palettes. If, for instance, you have a button in an Excel 4.0 toolbar that applies formatting commands to the selected cells, you could create a macro that clicks your button. This would save you the trouble of "recreating the wheel."

“Fixing” the Enter Key

I have always expected the **Enter** key to work in exactly the same way the **Return** key works in a spreadsheet program, but it never does. In Excel 3.0 and 4.0, Resolve, and 1-2-3, the **Enter** key enters the number, but it does not advance the insertion point to the next cell. The **Return** key, however, enters the number and advances the insertion point to the next cell. If you are trying to use the keypad on your extended keyboard in the same way you would use a 10-key calculator, this is a hindrance. And, moving your hand from the keypad to press the **Return** key can dramatically slow you down. You can create an alias QuickKey that makes the **Enter** key function the same as the **Return** key.

Before we begin: Open your spreadsheet program.

Create the macro:

1. Open QuickKeys and select your spreadsheet program from the pop-up menu.
2. Select Alias from the Define menu.
3. When the Alias editor opens, click in the Key to type box to select it and then hold down the mouse button and press **Return**. This is the key that QuickKeys sends to the program.
4. Click in the Keystroke field to select it and then hold down the mouse button and press **Enter**. This is the key you press. Your Alias editor should look like Figure 5-3.
5. Click OK to close the Alias editor.

Variations: While entering numbers in your spreadsheet, you may have noticed that holding down the **Shift** key modifies the behavior of the **Return** key. It causes the insertion point to move to the cell immediately above the current cell. You could add an additional alias QuickKey that makes the **Shift-Enter** combination mimic the behavior of the **Shift-Return** combination. It would look like Figure 5-4.

Figure 5-3: The Return alias QuickKey.

Figure 5-4: The Shift-Enter alias QuickKey.

Changing Worksheet Views

If you have ever worked with a very large worksheet, you know how difficult it can be to get the big picture. You can scroll through the worksheet to see all of the information, but the screen-sized bites of information do not give you the same perspective as being able to look at hundreds of cells at once. Fortunately, some spreadsheet programs have added commands that let you reduce your worksheet so you can see more than 20 or 25 rows. In particular, Microsoft Excel 4.0 and Claris's Resolve both have reduce and magnify commands. While the

magnify command may seem superfluous, magnifying a worksheet can make doing detailed work with graphics and charts much easier. For instance, magnifying a chart makes it easier to place labels and saves you the trouble of exporting the chart to a graphics program.

Before we begin: Open Microsoft Excel 4.0 or the spreadsheet program you use. You'll need to confirm that your spreadsheet has commands for showing reduced and magnified views. If you are a Claris Resolve user, open it; I'll talk about creating a version of this macro for Resolve below. Now open QuickKeys, select Microsoft Excel from the pop-up menu, and click OK to close QuickKeys.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Select Zoom... from Excel's Windows menu. A window appears where you can enter the amount by which you want to magnify the spreadsheet (Fig. 5-5).

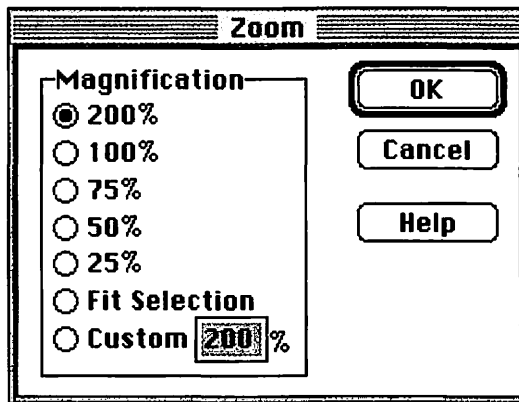


Figure 5-5: The Zoom window in Microsoft Excel 4.0.

3. Click the 200% radio button or enter 200 in the Custom text box.
4. Press **Return** to close the Zoom window and change your view of the worksheet.
5. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, your macro should look like Figure 5-6. If you typed 200 in the Custom text box, your macro will have a text

QuickKey in place of the click and **Return** alias QuickKeys in this figure, and the **Return** will be included in the text QuickKey.

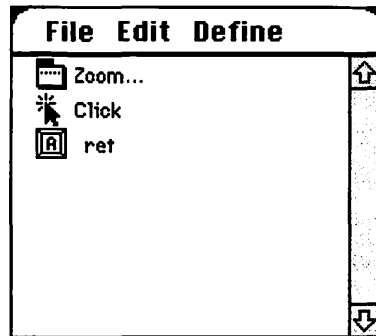


Figure 5-6: The completed Magnify200% sequence.

6. Name your macro and assign a keystroke. Then click OK to close the Sequence editor. Click OK to close QuickKeys. Now let's create a macro that reduces the view of the worksheet.
7. Select Record Sequence from the QuickKeys menu.
8. Select Zoom... from Excel's Windows menu.
9. Click the 50% button or enter 50 in the Custom text box.
10. Press Return to close the Zoom dialog box and change your view of the worksheet.
11. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, your macro should look like the one you created above.
12. Name your macro and assign a keystroke and then click OK to close the Sequence editor. Click OK again to close QuickKeys. Now you have macros that let you magnify and reduce your view of a worksheet.

Variations: Neither Microsoft Excel 3.0 or Lotus 1-2-3 have commands to magnify or reduce the view of your worksheet. Claris' Resolve does, but the reduce and magnify commands are icons in the worksheet window instead of menu items (Fig. 5-7).

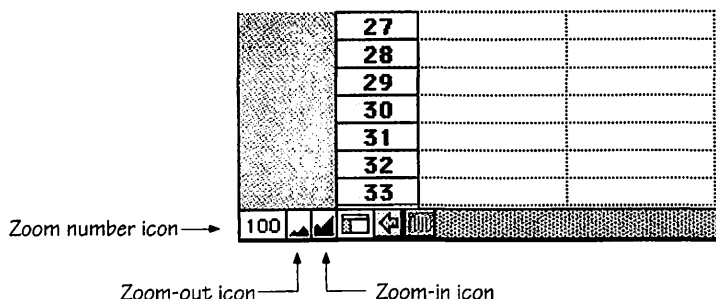


Figure 5-7: A portion of a Resolve worksheet window showing the Zoom icons in the lower-left corner of the window.

Clicking on the Zoom-in icon magnifies your view of the worksheet; clicking on the Zoom-out icon reduces your view of the worksheet. The Zoom number icon displays the current magnification or reduction scale.

Let's create macros to reduce and magnify your worksheet view:

1. Select Record One QuickKey from the QuickKeys menu.
2. Click on the Zoom-out icon. This reduces your view of the worksheet.
3. When the Click editor opens, click the Click button to open the Click Location editor (Fig. 5-8).

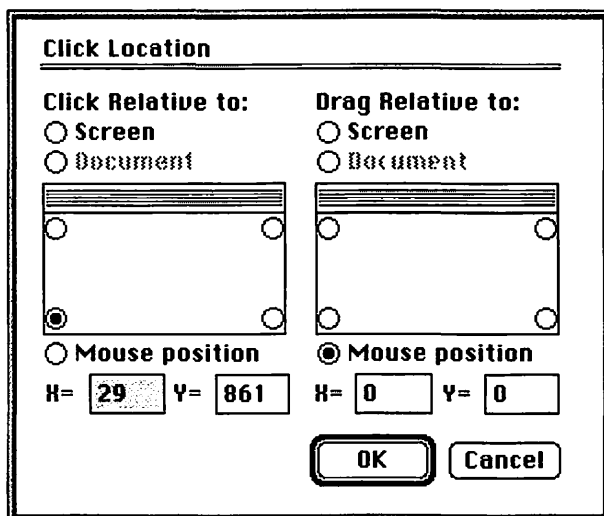


Figure 5-8: The Click Location settings for the Zoom-out icon.

4. Click the button in the lower-left corner of the small document window under Click Relative to. This setting causes QuickKeys to measure the click from the lower-left corner of the window and ensures that the macro always clicks in the same place even when you resize the worksheet window.
5. Click OK to close the Click Location editor. Name your macro and assign a keystroke. Then click OK to close the Click editor.
6. Click OK to close QuickKeys.
7. Click Record One QuickKey from the QuickKeys menu.
8. Click the Zoom-in icon. This magnifies your view of the worksheet.
9. When the Click editor opens, click the Click button to open the Click Location editor and change the settings as you did in step 4.
10. Then click OK to close the Click Location editor, click OK to close the Click editor, and click OK to close the QuickKeys window and save your changes.

You may also want to take a moment to create a click QuickKey for the Zoom number icon. Whenever you click the Zoom number icon, Resolve returns you to a 100% view of your worksheet. If you then click the Zoom number icon a second time, Resolve takes you back to the previous view. A click QuickKey allows you to toggle between two views of your worksheet.

Each successive time you click on the Zoom-out or Zoom-in icons, the magnification or reduction of your view of the worksheet increases. For instance, clicking on the Zoom-out icon four times reduces the worksheet to one-quarter of its size, and clicking on the Zoom-in icon four times increases the worksheet size fourfold. You could press the keyboard shortcut you assigned to the ClickZoomOut macro four times to reduce the worksheet to one-quarter of its size, but it would be more efficient to create a sequence that does it for you:

1. Open QuickKeys.
2. Select the ClickZoomOut macro in the QuickKeys window.
3. Copy the macro.
4. Select Sequence from the Define menu.

5. When the Sequence editor opens, select Paste from the Edit menu. Do this four times until your sequence looks like Figure 5-9.

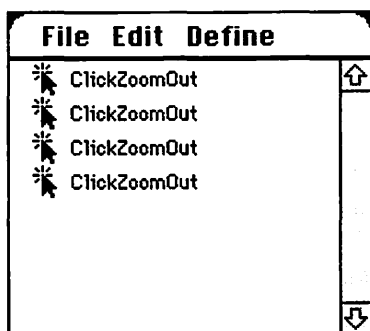


Figure 5-9: The final Reduce25% macro.

6. Name your sequence something informative (e.g., Reduce25%) and assign a keystroke. Then click OK to close the Sequence editor and click OK to close QuickKeys.

First-time Excel users are always surprised when they type $\text{⌘}-X$ to cut the selected item from their worksheet. Unlike most programs, it doesn't immediately remove the selection from the worksheet. Instead, it places a dotted outline around the selection and prompts you to "Select destination and press Enter or choose Paste." Excel assumes that when you use the Cut command, you merely want to move it to another location not completely remove it from the worksheet. Excel has another command, Clear, that behaves like the Cut command in other programs. Most people probably just retrain themselves to press a different keystroke, but you don't have to.

Before we begin: Open Excel and select a row or column of numbers to delete. Open QuickKeys and select Excel from the pop-up menu.

Create the macro:

1. Select Record a Sequence from the QuickKeys menu.
2. Type $\text{⌘}-X$. This copies the selected numbers to Excel's Clipboard. The Cut command in most programs performs this step.
3. Type $\text{⌘}-B$ or select Clear from the Edit menu. This removes the selected numbers from the worksheet.
4. Select Stop Recording... from the QuickKeys menu.

- When the Sequence editor opens, type ⌘-Option-X for the keystroke. Now you have a Cut command in Excel that works like it does in other programs. Your completed macro should look like Figure 5-10.

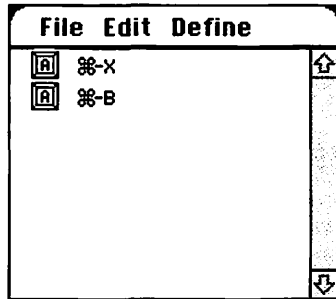


Figure 5-10: The Final ExcelCut macro.

Some keyboard shortcuts have become so ingrained in the Macintosh vernacular that I expect them to work the same way in every program I use. For instance, most of the page layout programs have adopted the same keyboard shortcuts as word processors for changing the style of and justifying text. Some graphics programs, like Canvas, use these same keyboard shortcuts. In most word processors and page layout programs, ⌘-Shift-B makes the selected text bold, and ⌘-Shift-R right-justifies the text.

Excel, Lotus 1-2-3, and Resolve all bow to these ingrained conventions for changing the text style, but they do not all have keyboard shortcuts for the justification commands (called alignment commands in spreadsheet programs).

Before we begin: Open your spreadsheet program. I'll be using Excel 4.0 for this example. If you are using Resolve, you will notice that it has keyboard shortcuts for left, right, and centered alignment. However, those shortcuts are different than the standard shortcuts (⌘-L , ⌘-R , and ⌘-C , respectively). You may want to use alias QuickKeys to redefine them to ⌘-Shift-L , -R , and -C if you do not want to use the commands that Resolve has already assigned them. If you are using 1-2-3, you'll discover that it also has these shortcuts assigned to different commands. Here's what we'll do in Microsoft Excel.

Create the macro:

- Select Record Sequence from the QuickKeys menu.

2. Select Alignment... from the Format menu.
3. When the Alignment window opens (Fig. 5-11), click the Right radio button.
4. Press **Return** to close the window.
5. Select Stop Recording... from the QuickKeys menu.

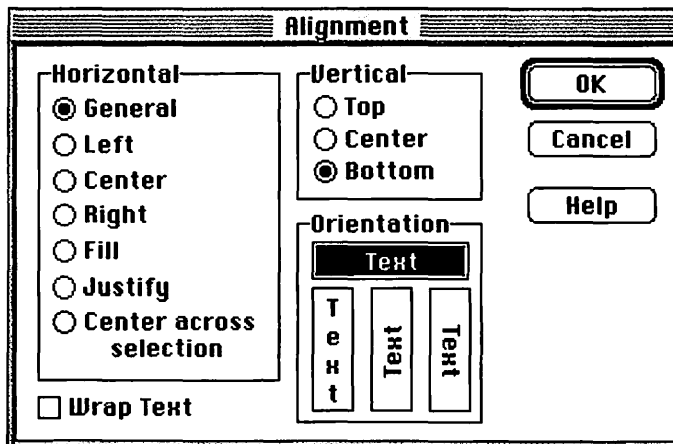


Figure 5-11: The Alignment window in Microsoft Excel 4.0.

6. When the Sequence editor opens, name the macro and assign the **⌘-Shift-R** keystroke—or any keystroke you prefer. You might also want to rename the click QuickKey so it is easier to read your macro. Your macro should look like the one in Figure 5-12.
7. Click OK to close the Sequence editor and then click OK again to close QuickKeys and save your changes.

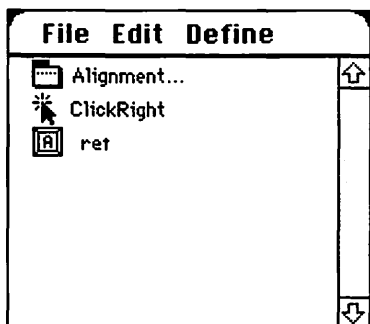


Figure 5-12: The completed RightJustify macro.

Variations: Creating left and center alignment versions of this macro is very simple. You only have to change the click QuickKey so it clicks the Left and Center buttons in the window. You'll also notice that Excel 4.0 has commands for changing the orientation of the selected text. If you routinely change the orientation of text for presentation slides in Excel, you may want to create macros for the orientations you use most.

Excel 4.0 also has commands in the Standard Toolbar for increasing and decreasing the type size of the selected text. If you use Word's keyboard shortcuts to change the size of selected text, you could create macros for Excel that perform the same function:

1. Select Record One QuickKey from the QuickKeys menu.
2. Click the Increase Font Size icon in the Standard Toolbar (Fig. 5-13).



Increase Font Size icon

Decrease Font Size icon

Figure 5-13: The Increase Font Size and Decrease Font Size icons.

3. When the Click editor opens, name the macro something appropriate. Then click the Keystroke field and press ⌘ -Shift-> (which is Word's keyboard shortcut). Notice that it appears in the Keystroke field as ⌘ -Shift-. (period). Your completed macro should look like Figure 5-14.

Click

Name: Keystroke:

Click: From:(314,14) from top-left corner
To:(0,0) from current location

Window:

Control area:

Click time(s)

Hold down: ☐ ⌘ ☐ Shift ☐ Option ☐ Control

Timer Options ☐ Include in QuickKeys menu

Figure 5-14: The final IncreaseSize macro.

4. Click OK to close the Click editor and then click OK again to close QuickKeys.

You can create a DecreaseSize macro that functions like Word's command by clicking on the Decrease Font Size icon and assigning the keystroke **⌘-Shift-←** (appears as **⌘-Shift-←**, in the Click Keystroke field).

Automatic Calculations

All of the spreadsheet programs have a Paste Function or similar feature that lets you select formulas from a pop-up menu or scrolling list. This saves you the trouble of trying to remember a formula or looking it up every time you need to use it. However, this makes entering the formula into the cell a two- or three-step process.

In Claris Resolve, for example, you must open the Paste Function window, select the appropriate group of formulas from the pop-up menu, and then select the formula you want Resolve to place in the cell (Fig. 5-15). You could create a macro that selects the Paste Function... command from the Calculate menu and then makes your selection for you. Such a macro would be rather cumbersome: It would take a number of seconds to run, and you would have to worry about timing the selection of the pop-up menu and then selecting the formula. A much simpler way to create a QuickKey for this action is to paste the formula into a cell in your spreadsheet and then copy it into a text QuickKey.

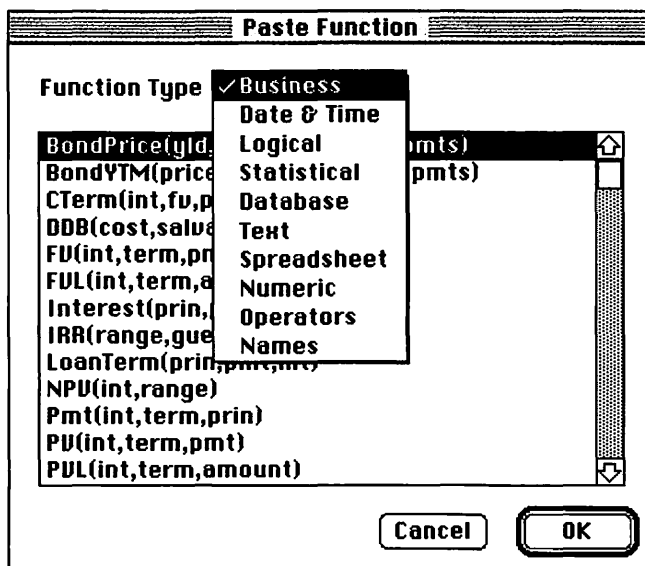


Figure 5-15: The Paste Function window in Claris Resolve.

Sum Macro

Like most spreadsheet users, you probably enter the SUM formula more frequently than any other. If you are a Microsoft Excel user, you can click the AutoSum icon in the Toolbar to enter the formula, but if you are a Resolve or Lotus 1-2-3 user, you have to paste or type the formula whenever you need it. Let's create a macro for Lotus 1-2-3 or Resolve that automatically enters the SUM formula.

Before we begin: I'll use Lotus 1-2-3 for this example. You should have your spreadsheet program open. Open QuickKeys and select the name of the spreadsheet from the pop-up menu.

Create the macro:

1. Select Record a Sequence from the QuickKeys menu.
2. Type "@SUM()" in the cell. If you are using Resolve, you would enter "=SUM()."
3. Press **Option**+**←**. This moves the cursor back between the two parentheses so it is ready to enter the range of cells you select with the mouse.

NOTE: If you press only the **←**, it moves the insertion point to the next cell to the left.

4. Select Stop Recording... from the QuickKeys menu. Your macro should look like the one in Figure 5-16.

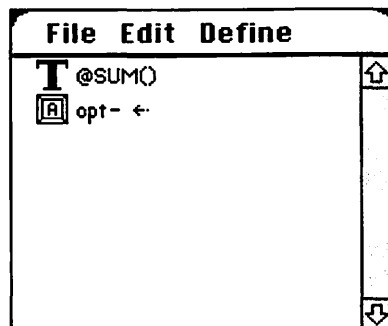


Figure 5-16: The final AutoSUM macro for Lotus 1-2-3.

5. Name your macro and assign a keystroke. Click OK to close the Sequence editor and then click OK to close QuickKeys and save your changes.

Variations: Excel's AutoSum function not only enters the formula, but it checks the cells above and to the left of the cell with the formula and automatically totals either the column or the row. While you can't add this same feature to Lotus 1-2-3 or Resolve, you can mimic it if you have a column or row of cells that is always the same length. To do this, you would enter the range of cells in the text QuickKey along with the formula. You would need to follow the text QuickKey with a **Return** or **Tab** to enter and calculate the formula. The macro might look something like the one in Figure 5-17.

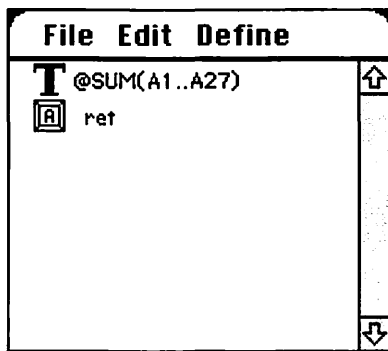


Figure 5-17: The ConfiguredSUM macro.

Equation Solver

Of course, spreadsheets are good for more than just calculating the amortization of a home loan. That may be what you bought your spreadsheet for, but you can also use it to calculate simpler formulas when you're lazy. You can use spreadsheets to solve most equations. Whenever you have standard formulas, you can automate more than just the insertion of the function. In some cases, you can even automate entering most of the formula into the spreadsheet as well as let QuickKeys remember things you always forget.

Whenever I've ever needed to get the radius of a circle or the volume of a cylinder, I've always had to make a trip to a reference book to find the formula before I could even begin the calculation. By creating macros that already contain the formula, I eliminate the trip to the reference book. I can even let QuickKeys solve most of the equations after I've entered the initial numbers. To get the area of a circle, for example, I can

enter the radius, let QuickKeys copy the radius into another cell in the spreadsheet, enter pi, and then have the spreadsheet calculate the area. You can even let QuickKeys prompt you for the information as needed. Let's create a macro that calculates the area of a circle ($\text{radius} \times \text{radius} \times \pi$) so you can see how easy this is.

Before we begin: Open the spreadsheet you are going to use and open a new worksheet. Then open QuickKeys and select the spreadsheet from the pop-up menu.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Enter some number. This is a placeholder for the radius of the circle. When we are through recording most of the steps in the macro, we'll remove this number to make the macro more generic.
3. Press **[Tab]** to move the insertion point to the next cell. This enters the number in the cell so we can come back and copy it.
4. Press **[Shift][Tab]**. This moves the insertion point back to the cell containing the radius. Since we need to multiply the radius by itself, we need to enter it in the next cell.
5. Select Copy from the Edit menu or press **[⌘]-C** to copy the radius.
6. Press **[Tab]** again to move the insertion point back to the second cell.
7. Select Paste from the Edit menu or press **[⌘]-V** to paste the radius.
8. Press **[Tab]** to move to the third cell. This is where we'll enter pi.
9. Enter 3.1415 in the cell.
10. Press **[Tab]** to move to the fourth cell. We are ready to enter the formula.
11. Type `=PRODUCT(A1,B1,C1)`. If you are using Lotus 1-2-3, you will enter `@SUMPRODUCT(A1..C1)`.
12. Select Stop Recording from the QuickKeys menu. We have most of the steps we need for the macro, but we'll need to clean the macro up a bit.

When the Sequence editor opens, you can see that steps 2 through 4 are in a single text QuickKey at the beginning of the sequence and that steps 9 through 11 are in a single text QuickKey at the end of the sequence (Fig. 5-18). We'll need to make these individual QuickKeys, but first let's make the beginning of the sequence more generic.

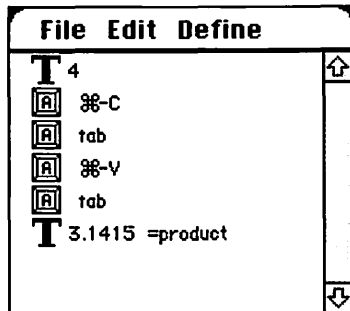


Figure 5-18: The unedited AreaOfaCircle macro.

13. Select the text QuickKey at the top of the sequence and select Cut from the Edit menu. We will need to replace this with a message to prompt you to enter the radius of the circle as well as a pause so you can enter the radius.
14. Select Message from the Extensions menu. When the Message editor opens, enter a message to prompt you to enter the radius (Fig. 5-19). Then select the Show OK button checkbox. Click OK to close the Message editor.

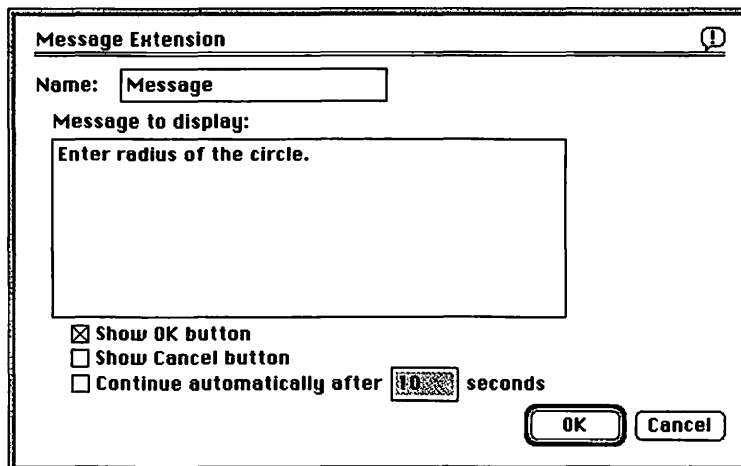


Figure 5-19: The Message extension for the AreaOfaCircle macro.

15. Select Pause from the Define menu of the Sequence editor. When the dialog box appears, enter 3, but don't hit Return. A three-second delay probably allows you enough time to enter the radius.
16. Now open the last text QuickKey in the sequence and select the text `"=PRODUCT(A1,B1,C1)"` in the text box. Don't select the small triangle—it is the **Tab** you pressed to move from the third cell to the fourth cell. Select Cut from the Edit menu. When you have removed the text, press **Backspace** or **Del** to remove the triangle. We will make a literal to replace this **Tab**; a literal will make the macro easier to read since we will be able to see all of the steps. Click OK to close the Text editor.
17. Click the Literal button and press **Tab**.
18. Select Text from the Define menu. Click in the large text area and select Paste from the Edit menu. Now you have your formula complete. Click OK to close the Text editor.
19. Click the Literal button and press **Tab** or **Return**. This causes the program to calculate the equation. Your final AreaOfaCircle macro should look like the one in Figure 5-20.

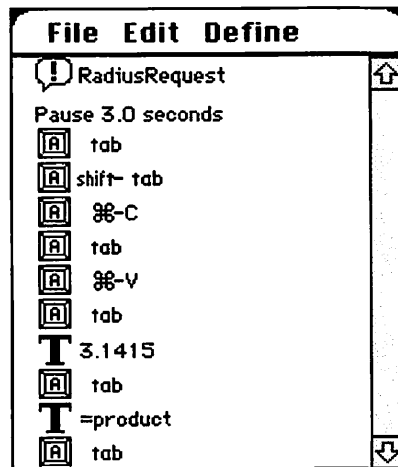


Figure 5-20: The final AreaOfaCircle macro.

20. Click OK to close the Sequence editor and then click OK again to close QuickKeys.

Variations: You can create macros for any equation you use on a regular basis. Most macros can be based on the AreaOfaCircle macro with a slight modification. For example, you can make the AreaOfaCircle calculate the surface area of a cube ($\text{length} \times \text{length} \times 6$ —the number of sides). You enter 6 instead of pi in the third cell. By removing the copy and paste and then adding another message and pause, you can create a macro to calculate the area of a rectangle ($\text{width} \times \text{height}$).

Another variation is to calculate the volume of a cylinder ($\text{radius} \times \text{radius} \times \pi \times \text{height}$). Again, the macro is basically the same as the AreaOfaCircle macro, but you need additional steps prompting for the height of the cylinder. You can copy the AreaOfaCircle macro and make the changes and add a message QuickKey to request the height of the cylinder. You follow that with another three-second pause and a Tab. The final macro looks like Figure 5-21.

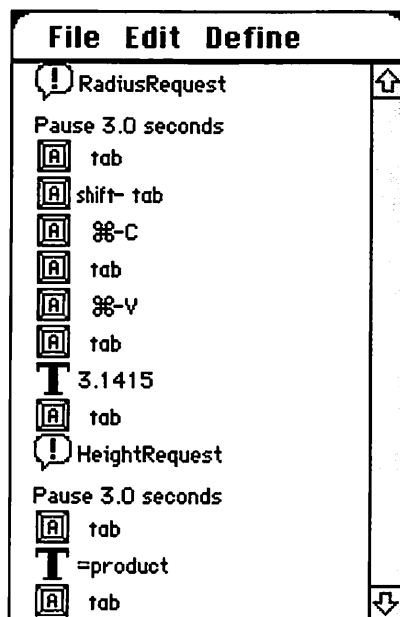


Figure 5-21: The final VolOfaCylinder macro.

These may seem to be trivial examples, but when you substitute any formula that's important to you, it becomes a valuable tool.

Formatting with a Single Keystroke

As we've already seen in the preceding chapters, the real power of macros is most apparent when you can reduce a number of actions or commands to a single keystroke.

Formatting Cells

Even though Excel 3.0 has buttons for formatting cells in the menu bar and Excel 4.0 has a Formatting Toolbar (which is really a windoid) devoted to formatting commands, those buttons are best used when you are experimenting with different formats in your worksheet. As with pressing several keystrokes when you are formatting text in a long document, you should combine several worksheet formatting commands into a single macro and assign them to a keystroke. When you create macros to format your worksheet, you'll find that command key equivalents are more efficient than clicking on a button in a windoid.

If you have a standard format you use for all of your report worksheets, you can create macros that automatically format the different areas of the worksheet. For instance, you could create macros that set the size, style, and font for the column labels or that add a shade and a border for rows and columns.

Before we begin: Open your spreadsheet program. Open an existing worksheet that contains text and numbers that you can format. I'll be using Excel 4.0 for this example. Then open QuickKeys and select the spreadsheet from the pop-up menu.

Create the macro:

1. Select the column labels or text in the worksheet you want to format.
2. Select Record Sequence from the QuickKeys menu.
3. Select Font... from the Format menu.
4. When the Font dialog box opens, press **Tab** to move the insertion point to the text box below the scrolling font list (Fig. 5-22) and type the name of the font you want to use. You could select the font from the scrolling list, but the macro would fail whenever you added or removed fonts.

5. Press **Tab** to move the insertion point to the text box below the scrolling size list and then enter the size of font. Again, you could select the size from the scrolling list, but it is more reliable to type the size in the text box.
6. Press **⌘-B** to select the Bold checkbox.
7. Press **⌘-I** to select the Italic checkbox.
8. Click OK or press **Return** to close the Font window. Now we can center the text. If you created the CenterJustify macro above, you could copy the steps and paste them in here. You could also click the alignment icons in the Standard Toolbar instead of opening the Alignment window.

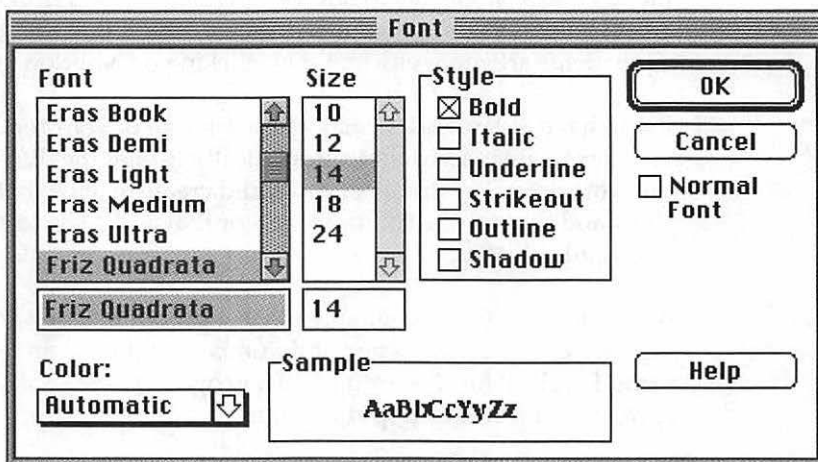


Figure 5-22: Microsoft Excel 4.0's Font window.

9. Select Alignment... from the Format menu.
10. When the Alignment dialog box opens, click the Center button.
11. Click OK or press **Return** to close the window and save your changes.
12. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, you'll see that the font and size information has been entered in a text QuickKey. You can leave the text QuickKey as it is, but it will be more accurate and modular if you move the information into separate text QuickKeys.

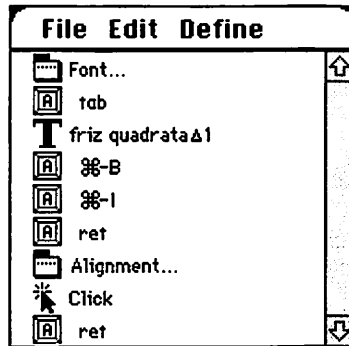


Figure 5-23: The unedited FormatLabels macro.

13. Open the text QuickKey and select the triangle and the font size in the large text box. Then press **⌘-X** to remove the text from the box. Remember that the triangle is the **Tab** you pressed to move from the font text box to the size text box. You will need to add an alias QuickKey to your sequence to take its place. Click OK to close the Text editor.
14. Move the insertion arrow between the text QuickKey and the **⌘-B**, click the Literal button, and then press **Tab**.
15. Select Text from the Define menu. When the Text editor opens, click in the large text box and press **⌘-V** to paste the size information you removed from the text QuickKey. Remember to remove the triangle from the text box. Click OK to close the Text editor. Your final macro should look like Figure 5-24.

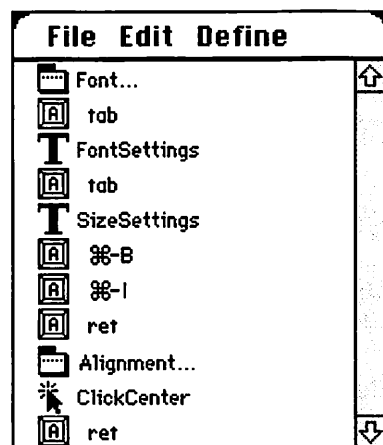


Figure 5-24: The final FormatLabels macro.

16. Click OK to close the Sequence editor and then click OK to close QuickKeys.

Variations: Besides formatting the column labels, you might also want to add a border and shading to the label cells. Both Resolve and Lotus 1-2-3 have pop-up menus containing the colors and patterns you can use for shading; however, QuickKeys has trouble selecting from the Excel 3.0 and 4.0 pop-up menus (which are different). (Perhaps the next full revision of the QuickKeys will be able to select from these pop-up menus.) Though you cannot work around the pop-up menu in Excel 3.0, Excel 4.0 does have a Light Shade button on the Formatting Toolbar. Using this button, you can set the shade for selected cells. To create the Border&Shade macro in Excel 4.0:

1. Select Record Sequence from the QuickKeys menu.
2. Select Toolbars... from the Options menu.
3. When the Toolbars window opens (Fig. 5-25), the Toolbar name text box is selected. Type "Formatting" in the box and press to close the Toolbars window and show the Formatting Toolbar.

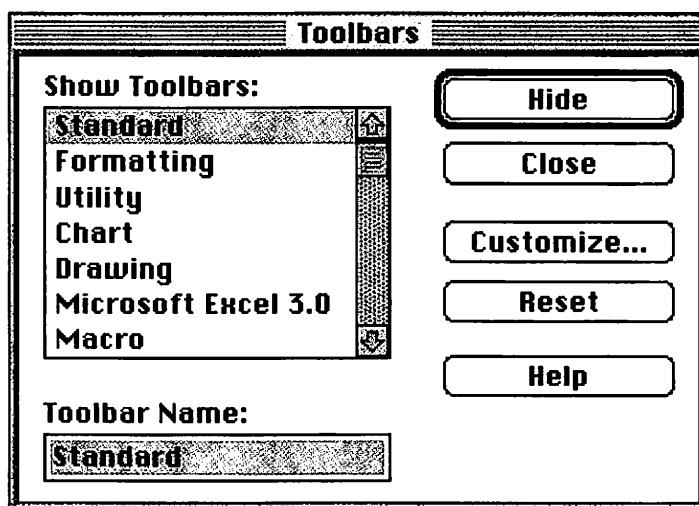


Figure 5-25: Microsoft Excel 4.0's Toolbars window.

4. When the Formatting Toolbar appears, click the Light Shade button (Fig. 5-26) to add a shade to the selected cells. Now you are ready to add a border around the cells.



Figure 5-26: Clicking the Light Shade button in Excel 4.0's Formatting Toolbar.

5. Press **⌘-Option-O**. Excel adds a border around the selected cells.
6. Select Stop Recording from the QuickKeys menu.

NOTE: Microsoft Excel 3.0 and 4.0 (**⌘-Option-O**) and Lotus 1-2-3 have keyboard shortcuts for adding borders around cells. If you use Resolve, you will find macros that select the outline, top, bottom, left, and right borders in the Resolve keyset. You can copy the QuickKeys in the OutlineBorder macro and paste them on the end of this sequence. Since we are using Excel 4.0 for the purposes of this example, we'll use its keyboard shortcut for an outline border.

When the Sequence opens, you'll notice that the return that selects the Formatting Toolbar has been included in the formatting text QuickKey. You'll need to remove the **Return** from the formatting text QuickKey and create an alias QuickKey to take its place. Your final macro should look like Figure 5-27.

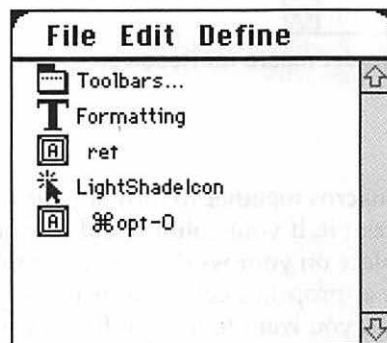


Figure 5-27: The final Border&Shade macro.

7. Name your macro and assign a keystroke.

8. Click OK to close the Sequence editor. Then click OK again to close QuickKeys and save your changes.

TRICK: If you are a Resolve user, you should look closely at the OutlineBorder, RightBorder, LeftBorder, TopBorder, and BottomBorder macros in the Resolve keyset. You'll see that each one has two CursorWait QuickKeys between the Cell Borders... QuickKey and the ClickOutline QuickKey (Fig. 5-28). When you click one of the buttons in the Cell Borders window, you'll notice that the cursor changes to a spinning beach ball before Resolve adds the border to the cells selected in the worksheet. The first CursorWait QuickKey pauses the macro until the ball appears, and the second one pauses the macro until the arrow appears. I have to add the first CursorWait because Resolve doesn't immediately change the cursor to a ball. And, if I only watch for the arrow, the macro will continue to run before the Cell Borders window ever opens.

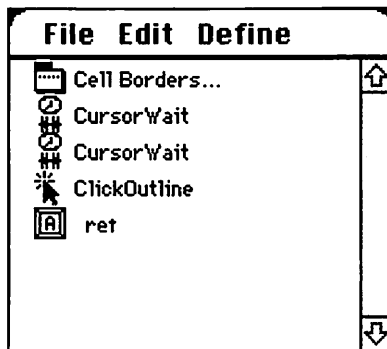


Figure 5-28: The OutlineBorder macro for Resolve.

Formatting a Worksheet

You can combine these macros together to format your report worksheet completely before you print it. If your column and row labels and totals are always in the same place on your worksheet, you can use the Go To command to jump to the appropriate cells and then use keyboard shortcuts to select the cells you want to format. If you want to make certain that the report looks right before you print it, you can add a step in the macro so it selects the Preview command.

Before we begin: Open your spreadsheet program and your report worksheet. For example, your worksheet might have column labels in row 3, row labels in column A, and totals in row 19 (Fig. 5-29).

	A	B	C	D	E	F	G
1							
2							
3		Qtr 1	Qtr 2	Qtr 3	Qtr 4	Regional Totals	
4							
5	Northeast	150000	175000	165000	180000	670000	
6							
7	Southern	130000	140000	142000	150000	562000	
8							
9	Midwest	145000	135000	150000	165000	595000	
10							
11	South Central	135000	130000	145000	155000	565000	
12							
13	Southwest	120000	100000	110000	115000	445000	
14							
15	Pacific Coast	175000	165000	150000	180000	670000	
16							
17	Northwest	125000	140000	130000	140000	535000	
18							
19	Qtrly Totals	980000	985000	992000	1085000		
20							
21							
22							

Figure 5-29: A sample report worksheet.

We can create a macro that selects and formats the row labels, the column labels, and the block of numbers. To make the row and column selections, we'll be using the spreadsheet's Go To command and selection keyboard shortcuts. I'll use Excel 4.0 for this example. Open QuicKeys and select the spreadsheet's name from the pop-up menu.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Press **[F5]-G** to open Excel's Go To window and enter the first cell number for the column labels: B5. Press **[Return]** to close the window and move the insertion point to the cell.
3. Press **[F5]-Shift-[Right Arrow]** to select the labels.
4. Select Font... from the Format menu.
5. When the Font window opens, press **[Tab]** to select the text box under the scrolling font list and type the name of the font you want to use.

6. Press **Tab** to move to the text box under the scrolling list of font sizes and type the size of font you want to use.
7. Press **⌘-B** to make the font bold.
8. Press **⌘-I** to make the font italic.
9. Press **Return** to close the Font window and apply your settings.
10. Select Alignment... from the Format menu. When the Alignment window appears, click the Center button and then press **Return** to close the window and save your changes.
11. Press **⌘-G**. When the Go To window appears, enter "a5" and press **Return**. This selects the first cell of the row labels.
12. Press **⌘-Shift-↓** seven times to select all of the labels in column A.
13. Select Font... from the Format menu.
14. When the Font window opens, press **Tab** to select the text box under the scrolling font list and type the name of the font you want to use.
15. Press **Tab** to move to the text box under the scrolling list of font sizes and type the size of font you want to use.
16. Press **⌘-B** to make the font bold.
17. Press **⌘-I** to make the font italic.
18. Press **Return** to close the Font window and apply your settings.
19. Press **⌘-G** to open the Go To window. Type "b5" and press **Return** to select the first cell in the block of cells containing the numbers.
20. Press **⌘-Shift-End**. This selects to the last cell in the block of numbers.
21. Select Font... from the Format menu.
22. When the Font window opens again, press **Tab** to select the text box under the scrolling font list and type the name of the font you want to use.

23. Press **Tab** to move to the text box under the scrolling list of font sizes and type the size of font you want to use.
24. Press **⌘-B** to make the font bold.
25. Press **⌘-I** to make the font italic.
26. Press **Return** to close the Font window and apply your settings.
27. Select **Number...** from the **Format** menu to open the Number Format window (Fig. 5-30).

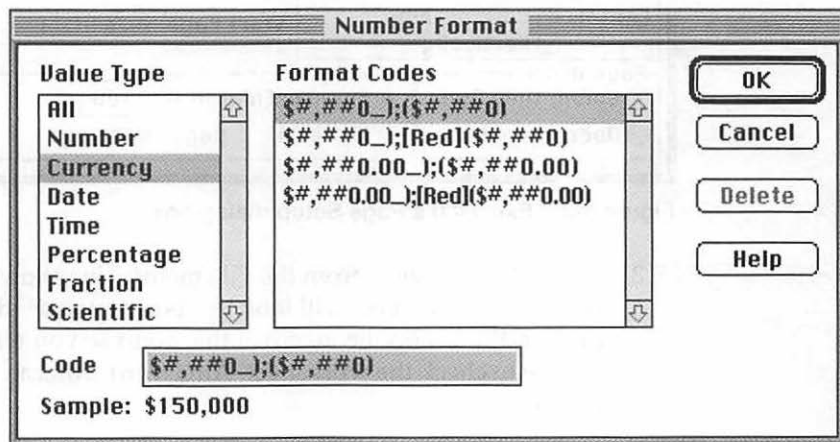


Figure 5-30: The Excel Number window.

28. Click **Currency** in the left scrolling list, click the top format code in the right scrolling list, and then press **Return** to close the window and apply your formatting. If you want to avoid the click **QuickKeys** here, you could enter the format code you see in the **Code** box at the bottom of the window and then press **Return**. This would insert a single text **QuickKey** in the sequence instead of the two clicks.
29. Select **Page Setup...** from the **File** menu. When Excel's **Page Setup** dialog box opens (Fig. 5-31), turn off the **Font Substitution**, **Text Smoothing**, **Graphics Smoothing**, **Faster Bitmap Printing**, and **Cell Gridlines** checkboxes.
30. Press **Return** to save your settings and close the dialog box.

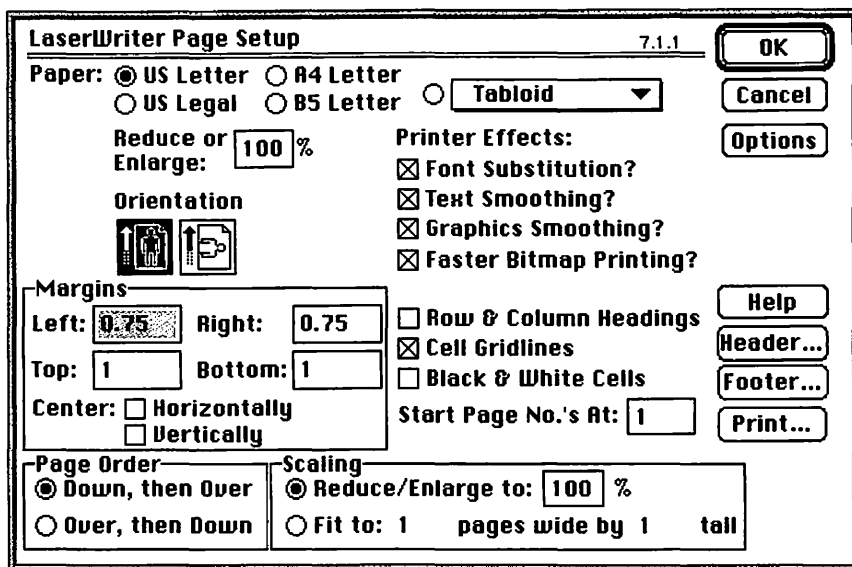


Figure 5-31: Excel 4.0's Page Setup dialog box.

31. Select Print Preview from the File menu. This allows you to see what your worksheet will look like before you send it to the printer. We'll stop the macro at this point so you will have a chance to check the worksheet. Afterward, you can click the Print button.
32. Select Stop Recording... from the QuickKeys menu. When the sequence editor opens, you'll see that some of your **Return** keys have been included with the text QuickKeys.
33. Open the "b3" QuickKey and remove the **Return** from the text box. Then move the insertion point below the "b3" text QuickKey, click the literal button, and press **Return**. You will need to complete these steps for each of the text QuickKeys in the sequence.
34. Open the NoFontSub QuickKey. When the Click editor opens, click the Control Area button and then click the Only click if button is on radio button. We change this setting so the macro does not accidentally turn on some of the options if they have already been turned off. You'll need to repeat these steps for each of the click QuickKeys toward the end of the macro. Your final macro should look like Figure 5-32, and your worksheet should look like Figure 5-33.

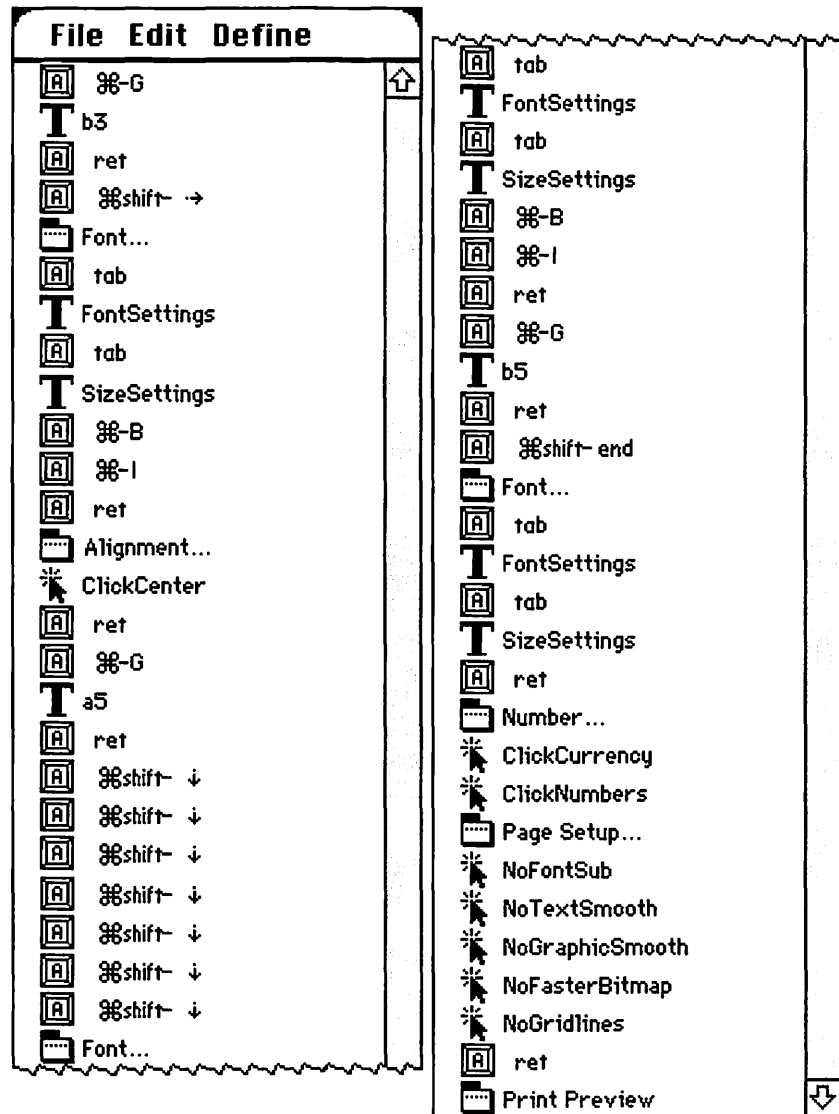


Figure 5-32: The final FormatWorksheet macro. (The macro has been split to fit on this page.) I've edited the click and text QuickKey names to make them more readable.

35. Name your macro and assign a keystroke. Click OK to close the Sequence editor and then click OK again to close QuickKeys.

Report Worksheet						
	A	B	C	D	E	F
1						
2						
3		Qtr1	Qtr2	Qtr3	Qtr4	Region Totals
4						
5	Northeast	\$150,000	\$175,000	\$165,000	\$180,000	\$670,000
6						
7	Southern	\$130,000	\$140,000	\$142,000	\$150,000	\$562,000
8						
9	Midwest	\$145,000	\$135,000	\$150,000	\$165,000	\$595,000
10						
11	South Central	\$135,000	\$130,000	\$145,000	\$155,000	\$565,000
12						
13	Southwest	\$120,000	\$100,000	\$110,000	\$115,000	\$445,000
14						
15	Pacific Coast	\$175,000	\$165,000	\$150,000	\$180,000	\$670,000
16						
17	Northwest	\$125,000	\$140,000	\$130,000	\$140,000	\$535,000
18						
19	Qtrly Totals	\$980,000	\$985,000	\$992,000	\$1,085,000	
20						
21						
22						
23						

Figure 5-33: The formatted worksheet.

Variations: Even though this macro may look complicated, it is actually very simple. We can extend it by adding the steps from the Border & Shade macro we created above. We could also add steps that create a chart from the contents of the report worksheet. In Excel 4.0, we would add the following steps:

1. Select Record Sequence from the QuickKeys menu.
2. Select Toolbars... from the Options menu.
3. When the Toolbar window opens, type "Chart" and press **Return**. This displays the Chart Toolbar. If the Chart Toolbar is already open, this step will close it.
4. Press **⌘-G**. When the Go To window opens, enter "a3" and press **Return**.
5. Press **⌘-Shift-End** to select your labels and the numbers.
6. Click the button in the Toolbar for the kind of chart you want to create.
7. Then click and drag a rectangular area on the worksheet for the chart. It may look like Figure 5-34.

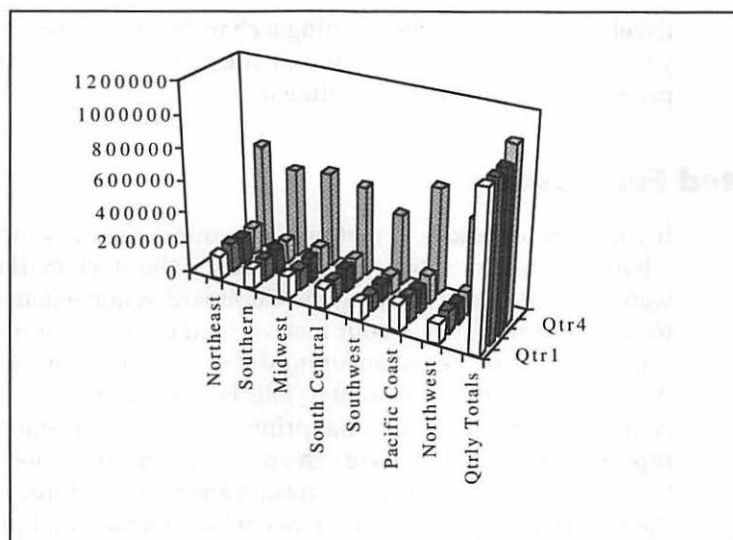


Figure 5-34: A chart generated from the report worksheet.

8. Select Stop Recording from the QuicKeys menu. When the sequence editor opens, your macro should look similar to Figure 5-35.
9. Name your macro and assign a keystroke. Then click OK to close the Sequence editor and click OK again to close QuicKeys.

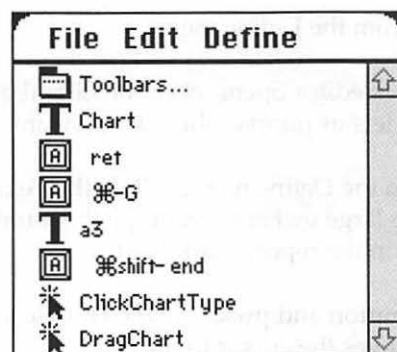


Figure 5-35: The final CreateChart macro. I've already edited the text QuicKeys and QuicKey names.

Most—if not all—of this macro could be created in Excel's macroing language. As I stated earlier, however, the two advantages of creating macros in QuicKeys are that you don't have to learn the spreadsheet's built-in macroing language and that you can extend the macro beyond

Excel. For instance, after creating a chart from your report worksheet, you could copy it and paste it into your word processor or to a graphics program like Canvas to fine-tune it.

Automated Forecasting

If you have a worksheet that projects your company's income, performs what-if calculations, or analyzes trends in the stock market, you may want to create a macro using the ClipboardAction extension to "watch" for significant changes. Your macro could copy the information from a cell and check it against an amount you've entered in the ClipboardAction extension. If the number falls below a certain point, the macro could play another macro that prints a report. For instance, using the report worksheet above, we can create a macro that checks the regional totals. When they fall below a certain amount, our forecast macro plays the FormatWorksheet macro above which formats and prints the worksheet.

Before we begin: Open your spreadsheet program and a worksheet. I'll be using Excel 4.0.

Create the macro:

1. Open QuickKeys and select the program from the pop-up menu.
2. Select Sequence from the Define menu.
3. When the Sequence editor opens, click the Literal button and press **[⌘]-G**. This lets us jump to the cell you want to check.
4. Select Text... from the Define menu. When the Text extension opens, click in the large text area and type the number of the cell, for example, "f5" in the report worksheet above.
5. Click the Literal button and press **[Return]**. This closes the Go To dialog box and moves the cursor to the cell.
6. Click the Literal button again and then press **[⌘]-C** to copy the number of the cell to the Clipboard.
7. Select Clipboard Action from the Extensions menu. When the extension opens, select the less-than symbol from the pop-up menu and enter "500,000" in the text box (Fig. 5-36).

In the Pass QuicKey text box, enter the name of the macro you want the CheckTotal macro to play if the number on the Clipboard is less than the number in the ClipboardAction extension. I'll enter the FormatWorksheet macro name. You also need to enter a macro name in the fail QuicKey. I'll enter the name of a sound QuicKey that I use to alert me that a macro is finished playing. Click OK to close the ClipboardAction extension and save your changes.

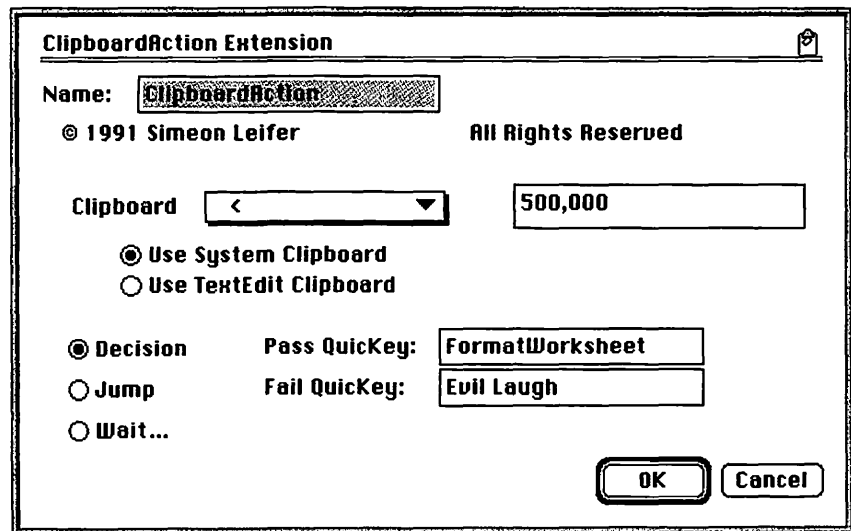


Figure 5-36: The Clipboard extension settings for the CheckTotal macro.

8. Your final macro should look like Figure 5-37. Click OK to close the Sequence editor and click OK again to close QuicKeys.

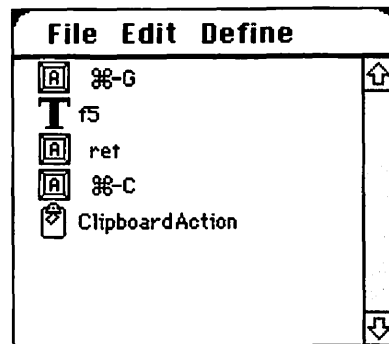


Figure 5-37: The CheckTotal macro.

Variations: While this macro only checks the contents of one cell, you could extend the macro to check every cell in the worksheet. Another variation of the macro could check stock fluctuations. If you download stock market information from an electronic service and import it into a worksheet, you could incorporate the ClipboardAction extension in a macro that watches for changes in certain stocks.

Moving On

If you have worked through the macros in this chapter, you see how QuicKeys can personalize your spreadsheet program and make your work more efficient. In the next chapter, we'll create macros for the graphics and page layout programs you use every day. You'll see how to create Apple Event macros to select tools and change your ruler settings. You will also find macros that automatically wrap text around graphics in PageMaker.

6

CHAPTER

Macros for Graphics Programs

One of the most important uses of QuickKeys with graphics programs is adding keyboard shortcuts to commands. Where most text editing programs have built-in commands or macro languages, drawing programs often have a minimum number of keyboard shortcuts. Fortunately, QuickKeys can help you automate most of the work you do in any graphics program—whether you are working with a desktop publishing package or a drawing or painting program.

This chapter shows you how to create macros for selecting tools from the palettes and toolboxes of your favorite graphics programs and assign keyboard shortcuts to them. It also contains macros for importing graphics and for combining your actions to make tool selections and formatting faster. We will be creating macros for Canvas, MacDraw Pro, UltraPaint, and PageMaker. If you do not have these programs, you should be able to adapt the macros to the programs you use. In some cases, I have included macros for other graphics programs on the companion disk.

Selecting Tools

That keyboard shortcuts are really a matter of personal taste is a given, but it is nowhere as obvious as in graphics programs. You may never take your hand from the mouse, choosing to make every menu selection and tool change with it. I find it more comfortable to keep one hand on the keyboard to make menu and tool selections and use the other hand to control the drawing tool.

I'm always amazed at the keyboard combinations developers force on the user. One of the really useful things about QuickKeys (which we

discussed in Chapter 3) is changing any keyboard shortcut. I find it difficult to use the **(Shift)-F1** through **(Shift)-F8** keyboard shortcuts that let you select the eight tools in PageMaker's tools palette, so I created aliases that are easier to reach than PageMaker's current settings. I chose keystrokes that I can perform with my left hand since I always have my right hand on the mouse to select a region or drag an object. Again, the choice is a matter of personal taste. I settled on **(⌘)-Option-1** through **(⌘)-Option-4** for the top four tools in the palette because I knew I could reach those quickly with my left hand. Since I don't use the bottom four tools on the palette as frequently, I haven't bothered creating alias QuickKeys for them. Unfortunately, not every developer provides keyboard commands for selecting tools. In the cases where keyboard commands are not included, you must create macros to select tools using click or real-time QuickKeys. These are functional, though not as elegant as built-in keyboard shortcuts.

Selecting Tools with a Click

While developers are adding more keyboard shortcuts to graphics programs for commands, very few are adding shortcuts for selecting tools. I suppose the assumption is that you will use the mouse to select a tool since you are already using it for drawing. However, it is faster to select tools from the keyboard than it is to mouse over to the tool palette and click the tool, particularly if you are toggling between two tools. Some tool palettes—like the pop-out palettes in Canvas—require a bit more mousing around. Let's create a macro for selecting a tool with a click QuickKey.

An Aside About Pop-Up Palettes and Windoids

Drawing tools in graphics and page layout programs tend to be incorporated into palettes in the document window or organized in floating palettes or windoids. These are small windows that contain drawing tools or tool settings and "float" above the document window. Click QuickKeys, which select a tool from a palette attached to the document window, are often the most reliable type of QuickKey for selecting such tools.

You can create a click QuickKey to select a tool in a windoid, but the results may not always be what you expect. Since you can move the windoid, there is always the chance that it won't be in the same location as when you created the click QuickKey. For instance, you may move the windoid to make more room on your screen for drawing. If the click QuickKey depends upon the location of the windoid, your macro will

fail. But there is hope. If the windoids are named, the click QuickKey will look for the windoid by name and click within it. The windoids in Canvas and Amazing Paint are named, while the windoids in SuperPaint are not. Usually, the windoid's name appears in its title bar. When the windoid doesn't have a name, you can also determine if it is named by creating a click QuickKey to select one of its tools or options. When the Click editor opens, click the Window button and look at the Name text box (Fig. 6-1). If a name appears in the text box, you can be sure your click QuickKey will always work.

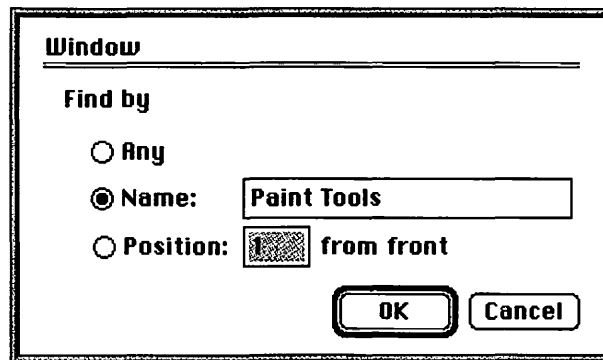


Figure 6-1: The Window options in the Click editor after clicking on the Eraser in Canvas's Paint Tools windoid.

Before we begin: Open the program for which you'll be creating the macro. I'll be using Canvas 3.0, but you can also use MacDraw since it also has a tool palette incorporated in the document window. If you are using Canvas, make sure to select the Retain selected tool checkbox in the Preferences dialog box. Open QuickKeys and select the program's name from the pop-up menu. We'll create this macro to click the Arrow tool. If you need to select a tool from a pop-up palette, you should take a look at the next macro.

Create the macro:

1. Select Record One QuickKey from the QuickKeys menu.
2. When the flashing microphone appears in the Apple menu, click directly in the center of the Arrow tool.
3. When the editor opens, click the Click button to open the Click Location editor. When you do, you'll discover that the upper-left button in the small document window is selected. If we knew that

the Canvas document window would never move on the screen or change size, this would be an ideal setting. Unfortunately, you can drag the window anywhere on the screen. Since we want the click to always strike the tool in the window in the same place, click the upper-left button in the window (Fig. 6-2). This causes QuickKeys to measure from the upper-left corner of the Canvas document window.

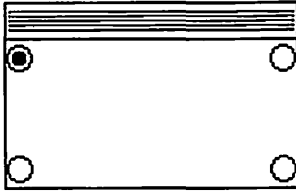


Figure 6-2: The Window icon in QuickKeys's Click Location editor.

4. Name the QuickKey and assign a keystroke. Click OK to close the Click editor and save your changes. Then click OK again to close QuickKeys.

Use and variations: Naturally, you can create macros for each of the tools you want to select from the keyboard. But, as we'll see in the next macro, it is a good idea to confine clicks to selecting tools only in palettes that are stationary. In other words, don't try to use click QuickKeys to select tools in pop-up palettes. In our next macro, you'll see how real-time macros can make pop-up palette selections more reliable. Real-time macros record your exact cursor movements and play them back at the same speed you recorded them, whereas clicks only record the starting and ending points of your actions. Another variation would be to add more functionality to the macros that select tools. As you'll discover later in this chapter, you can create macros that select the settings for the tool as well as selecting the tool.

Selecting Difficult Tools

As we saw with the previous macros, you can use click QuickKeys to select tools. But when you have to click and drag to select a tool, a click QuickKey usually doesn't work. The reason is that the click records the starting and ending points and not the movement in between. If the tool palette also has a pop-up palette (Canvas is a good example), you won't be able to select a tool in the pop-up with a click QuickKey. While you can tear off the palette to make it easier to record a click on a tool, if you move the palette, your macro may fail as described earlier. Not to

despair, QuickKeys can help you even in this situation. A real-time QuickKey, since it mimics your actions exactly, accurately repeats a tool palette selection.

Before we begin: Open your graphics program. I'll be recording a macro to select the Eraser from Canvas 3.0's pop-up menu. Because a real-time QuickKey reflects your mouse actions exactly and can't be edited, you'll want to position your pointer directly over the palette from which you are going to select the tool. This technique allows you to record the tool selection without any additional movements. Positioning the pointer in advance will also help you get the macro right the first time. If you haven't assigned a keystroke to the record real-time QuickKey, you'll need to do that; otherwise, you'll record the action of selecting the Record Real Time command from the QuickKeys menu. To assign a keystroke to the record real-time QuickKey, open QuickKeys and select Start/Stop Real Time from the Specials hierarchical menu. When the editor opens, assign a keystroke and then click OK to close the editor and save your changes. If you are ready now, open QuickKeys and select the name of the program from the pop-up menu.

Create the macro:

1. Position your cursor over the tool palette.
2. Press the keystroke you assigned to the record real-time QuickKey.
3. Click on the Paint Tools palette and drag out to the Eraser (Fig. 6-3).

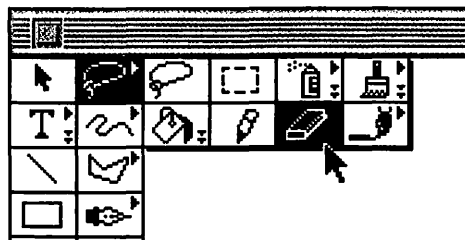


Figure 6-3: Selecting the Eraser tool from the Paint Tools palette in Canvas.

4. Release the mouse button and press the keystroke to stop recording the real-time macro. You should do this as quickly as you can so the macro won't take up any more time than necessary when you play it back.

5. When the Real Time editor opens, name your macro and assign a keystroke. Click OK to close the Real Time editor and save your changes. Then click OK again to close QuickKeys.

Use and variations: Real-time QuickKeys aren't the easiest to set up since they can't be edited. However, a real-time QuickKey is the tool for any situation where you want QuickKeys to exactly mimic your actions. Probably the best example of this in a graphics program is drawing with a brush. If you record a real-time macro that writes your name with the brush or pencil tool, it will be faithfully reproduced whenever you play back the macro. If you attempt to record the same thing using a sequence of clicks, you'll get something very unlike your signature.

Using Apple Events to Select Tools

While the preceding two macros will work with most programs, you can create a macro that is as reliable as typing a built-in keyboard shortcut to select a tool. If you are running System 7, you can use Apple Events to communicate directly with the program and tell it which tool to select. Using Apple Events in QuickKeys is more complicated than just creating a click QuickKey to select a tool and may seem like overkill. However, an Apple Event-based macro is more accurate than any macro based on screen location. I'll be using Canvas 3 for this macro because it is currently the only graphics program that understands Apple Events.

Before we begin: You should be running System 7 and have the Apple Events extension and CEIAC installed. The Apple Events extension and CEIAC let QuickKeys communicate directly with Canvas. You should also have Canvas 3 installed. Open QuickKeys and select Canvas from the pop-up menu.

Create the macro:

1. If you closed QuickKeys, reopen it and select Apple Events from the Extensions menu.
2. When the Apple Events editor opens, select Canvas from the Send Event to pop-up menu (Fig. 6-4). This will send an Apple Event from QuickKeys through CEIAC to Canvas.

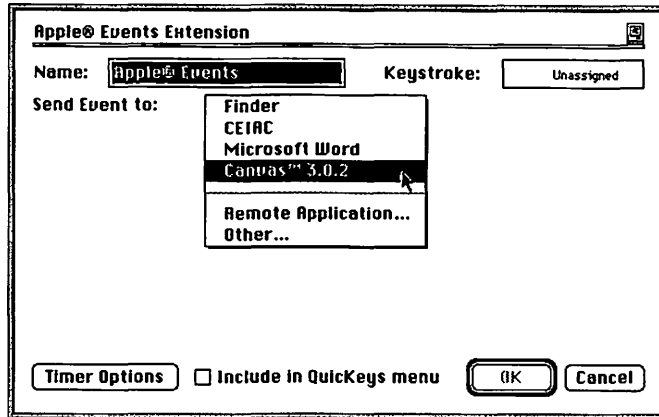


Figure 6-4: Selecting Canvas from the Send Event to pop-up menu.

3. After you select Canvas from the Send Event to the pop-up menu, the Event pop-up menu appears. Select Lookup from Target... from the Event menu (Fig. 6-5).

Because Canvas has an Apple Event Terminology Extension resource (aete resource), QuickKeys is able to look up the Apple Events it supports and display them in a list. If the program doesn't contain an aete resource, QuickKeys displays a dialog box telling you that the program doesn't have the resource. Whenever you encounter a program without the aete resource, you have to select Custom from the menu and enter the Apple Event parameters. We'll talk about this more in Chapter 11.

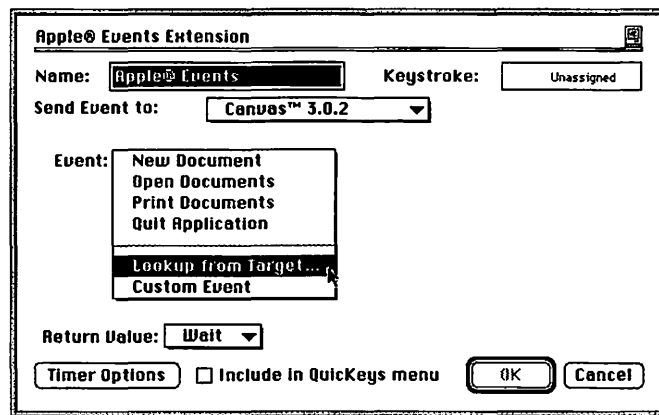


Figure 6-5: If the program takes advantage of Apple Events and contains an aete resource, QuickKeys can look up the events it supports.

After you select Lookup from Target... from the pop-up menu, an Events dialog box appears (Fig. 6-6). The Events dialog allows you to select the event that you want to send to Canvas.

4. Select Canvas Events from the Event Suites list and Built-in Tool from the Events in the selected suite list. Notice that the Class (CANV) and ID (btol) codes of the event appear below the scrolling list.

Let's take a few moments to talk about the class and ID codes. Apple Events are grouped into suites, and the class CANV refers to the suite of Apple Events that are specific to Canvas. For instance, programs that support Inter Application Communication under System 7 understand the required suite of Apple Events. These are Apple Events for opening and printing documents and quitting the application. When we send an Apple Event to Canvas with the class code of CANV, Canvas knows that it should look in its Canvas suite of events. The ID code tells Canvas which event in the suite it should use. In this example, the btol code refers to the event that selects built-in tools. If we were to send a CANV event with an etol ID, Canvas would select an external tool.

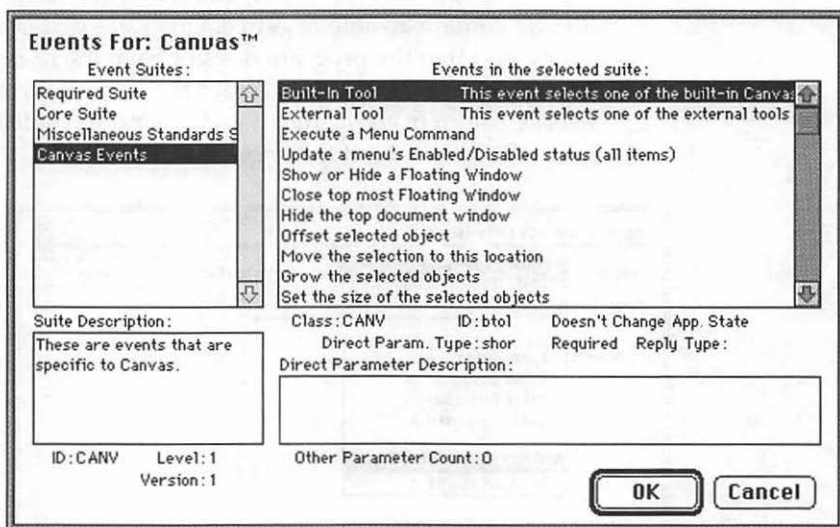


Figure 6-6: The Apple Event lookup dialog for Canvas.

Look at Figure 6-6, you can see that the event takes the direct parameter type short, which is a short integer and corresponds to

the numbers of the built-in tools. Look at Figure 6-9 to see the event IDs and numbers for the different tools. We will enter the shor parameter next. Click OK to close the dialog.

- When the Events dialog closes, you'll see that the parameter information has been added to the Apple Events editor (Fig. 6-7). To open the parameter editor, select the parameter line and then click the Edit... button.

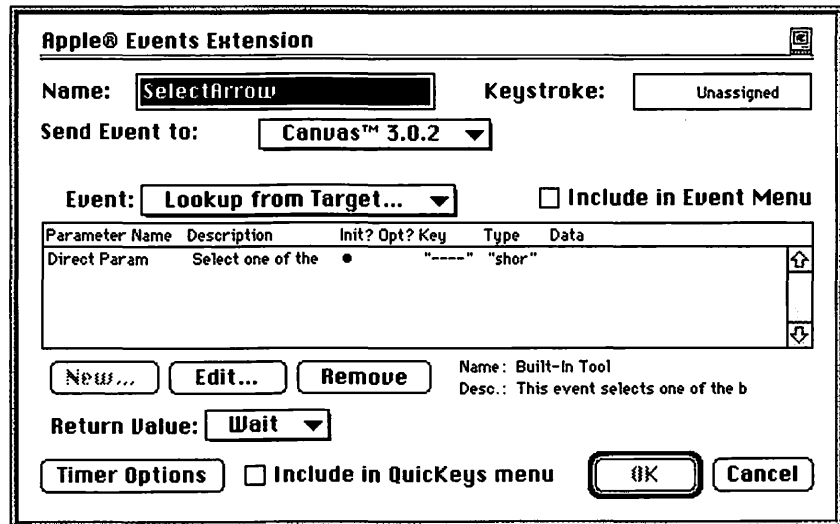


Figure 6-7: The Apple Events editor with the parameter information entered.

- When the Parameter editor opens, enter 1 in the bottom text box (Fig. 6-8). This is the btol number to select the Arrow tool.

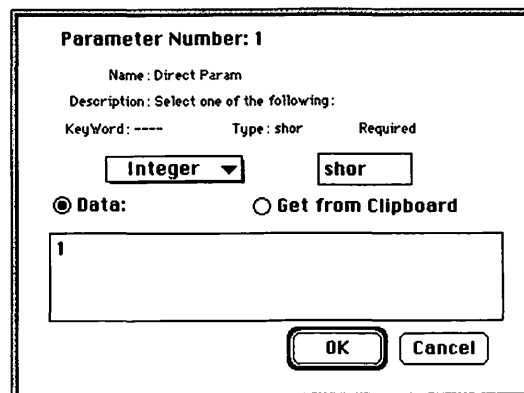


Figure 6-8: The Parameter editor dialog box.

7. Click OK or press **Return** to close the editor.
8. Name your macro and assign a keystroke. Then click OK or press **Return** to close the Apple Events editor and save your changes. Click OK to close QuickKeys.

Variations: You can select most of the tools in Canvas's palettes with Apple Events. Figure 6-9 shows you which numbers and events to enter to select the various tools. You can also use Apple Events to select patterns, colors, and magnification settings. You'll find a full set of Apple Event QuickKeys in the Canvas keyset on the companion disk.

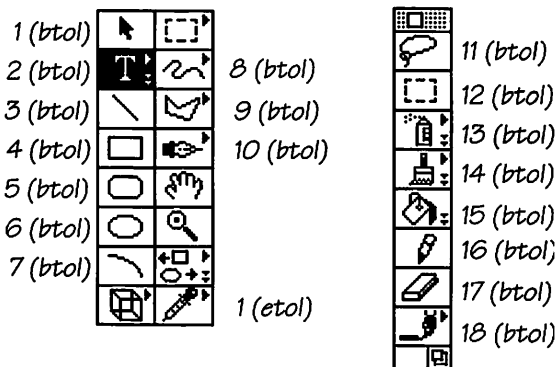


Figure 6-9: Canvas tools and the numbers and types of events necessary to select them with an Apple Event.

While this example deals with a built-in tool (btol), Canvas also has external tools (separate modules that you drop into the Canvas Tools folder) that require the etol Apple Event. For example, the tools in the Special Effects and Object palettes are external tools. If you want to select one of these tools with an Apple Event, you'll need to select the External Tool event in the Canvas events suite (Fig. 6-10).

You then need to enter the number that corresponds to the external tool. These numbers are not well documented. You can find the number of an external tool by opening the tool with ResEdit and looking at the tool's ID number. You will find macros for all of the standard external tools on the disk in the back of this book.

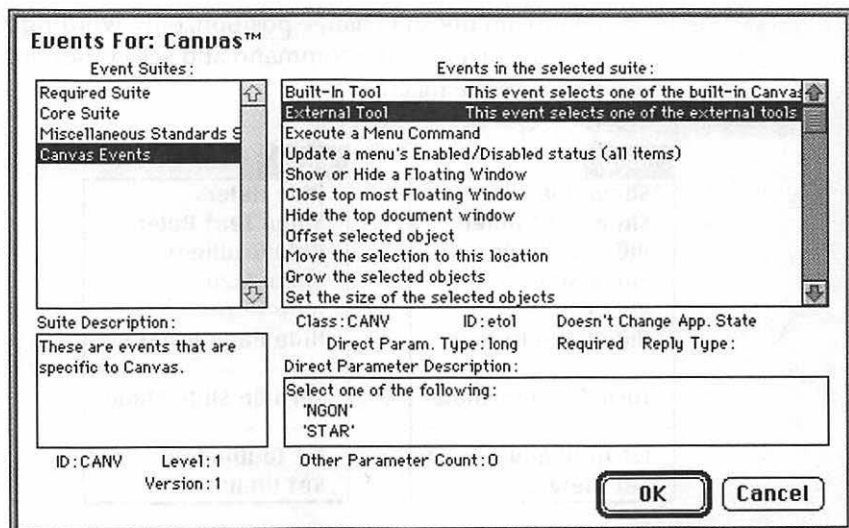


Figure 6-10: The External Tool event for selecting tools in the Objects and Special Effects palettes.

Toggles for Rulers

Besides creating macros for selecting tools, you can also create macros for selecting commands and adding features to programs. Most graphics programs have a command to let you toggle rulers on and off. A few still lack this particular feature; MacDraw Pro is one of them.

Before we begin: You should have MacDraw Pro running. Open QuickKeys and select MacDraw Pro from the pop-up menu.

Create the macro:

1. Select Record one QuickKey from the QuickKeys menu.
2. Select Show Ruler from the View menu (Fig. 6-11).
3. When the Menu editor opens, you'll notice that the Select from menu by Text button and the Match exactly checkbox are selected. Since the command changes when you toggle it on and off, we'll need to change both of these if we want to use the macro as a toggle.

Select the by position button (Fig. 6-12). This causes QuickKeys to select the command by its position in the View menu. Since the

command doesn't change position, only wording, QuickKeys can ignore the text of the command and select the command each time you play the macro.

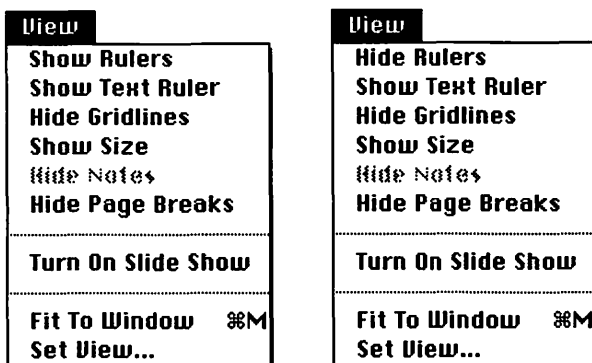


Figure 6-11: MacDraw Pro's View menu with the Show Rulers command at the top. Notice that the command changes after you have shown the rulers.

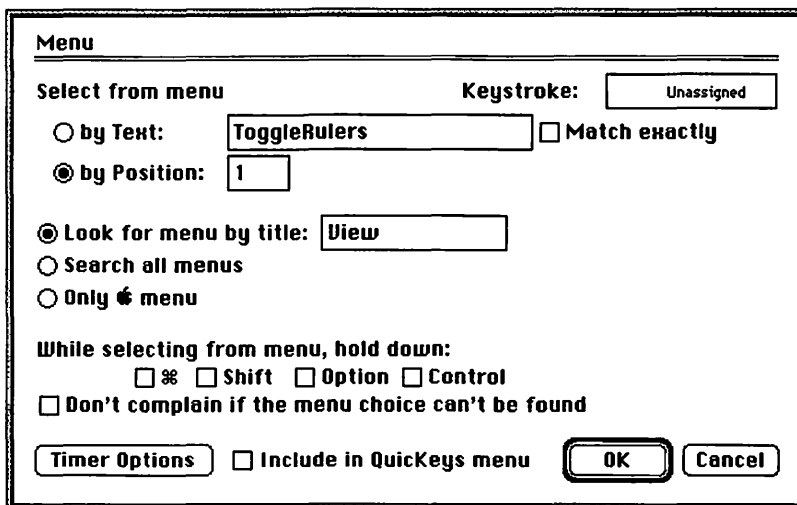


Figure 6-12: The Menu editor for the ToggleRulers macro.

4. Assign a keystroke to your macro and click OK or press **Return** to close the editor. Since QuickKeys doesn't use the text in the by Text box, you could enter a macro name there. This would let you know that it is different from other menu QuickKeys in your keyset. Click OK to close QuickKeys.

Variations: A slightly different ruler macro would change your ruler settings. For instance, you might use a standard ruler while working on a drawing but then want to switch to a scaled ruler to make certain you are drawing to scale.

1. Select Rulers... from MacDraw Pro's Layout menu. This opens the Rulers dialog box (Fig. 6-13).

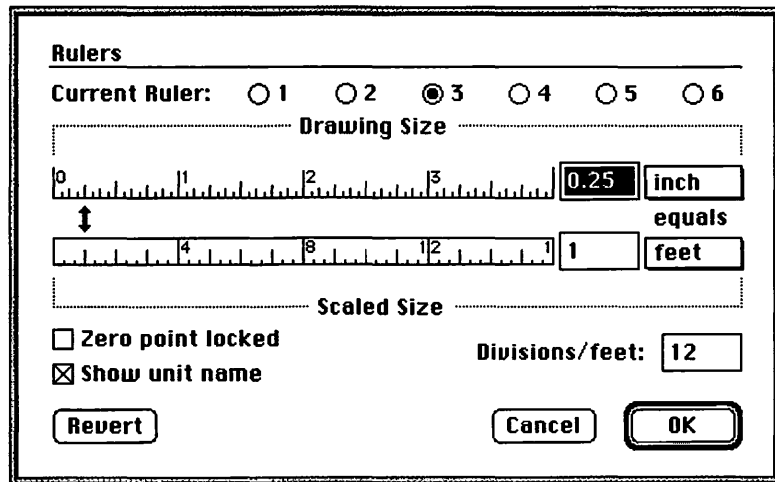


Figure 6-13: The Rulers dialog box.

2. Press **⌘-3** to select the third ruler setting. The default setting is .25 inch equals 1 foot.
3. Press **Return** to close the dialog box and save your changes.
4. Select Stop Recording from the QuicKeys menu. When the Sequence editor opens, your macro should look like the one in Figure 6-14.
5. Name your macro and assign a keystroke. Then click OK or press **Return** to close the Sequence editor. Click OK again to close QuicKeys.

You might also want to create a complimentary macro to switch your ruler back to its previous settings.

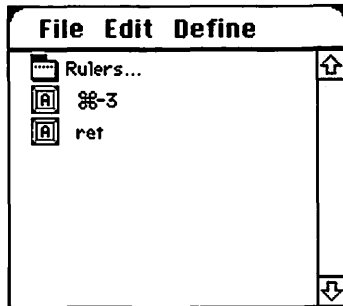


Figure 6-14: The final ChangeRuler macro.

Reduce and Magnify

A number of graphic and page layout programs have begun to incorporate commands that let you reduce or magnify the view of a drawing with a keyboard shortcut. But there are still some graphic programs for which you must select the amount of reduction or magnification from a pop-up menu. In both Canvas and Ultra Paint, you select a different view from a pop-up menu. While you can ⌘ -click with the pencil tool to increase magnification eightfold, this isn't always as handy as PageMaker's single keystroke shortcuts to change the view to 25, 50, 75, 200, or 400%. You can create macros to provide a one-keystroke alternative to changing the view using a pop-up menu. As with selecting tools from a pop-up palette above, you must use a real-time macro to select a different view from the menu.

Before we begin: You can create this macro for any graphics program that doesn't have a keyboard shortcut for reducing or magnifying the view of your drawing. I'll be using UltraPaint for this example since it has a pop-up menu for changing the view. You should have the program running. Open QuickKeys and select the name of your program from the pop-up menu. Because you will record a real-time macro to make the selection from the pop-up menu, you should position the pointer over the magnify and reduce pop-up menu.

Create the macro:

1. Move the pointer over the pop-up menu.
2. Press the keystroke you've assigned to the Record Real Time macro command.
3. Drag down to the 400% command and release the mouse button (Fig. 6-15).

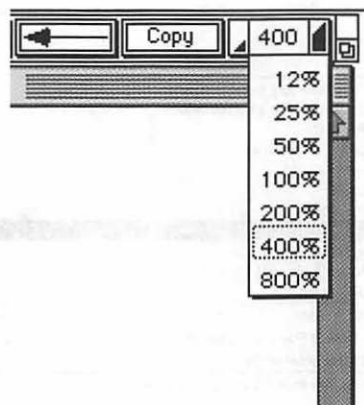


Figure 6-15: The Magnification pop-up menu in UltraPaint.

4. Press the Record Real Time keystroke quickly so you don't record any more of your mouse actions than necessary.
5. When the Real Time editor opens, name your macro and assign a keystroke.
6. Click OK or press to close the editor and save your changes. Then click OK to close QuickKeys.

Variations: While the real-time macro above isn't as convenient as the shortcuts already built into programs, it is the most accurate way to select a command from a pop-up palette. As with the tools above, you can also use Apple Events in Canvas to zoom your drawing in or out. You will need to have Canvas 3.0 to create and use this macro.

1. Launch Canvas. Then open QuickKeys and select Canvas from the pop-up menu.
2. Select Apple Events from the Extension menu.
3. Select Canvas from the Send Event to pop-up menu.
4. Select Lookup from Target... from the Event pop-up menu.
5. When the Events dialog box opens, select Canvas Events from the Event suites list and Magnification from the events in the selected suite list (Fig. 6-16). Notice that the class CANV and the ID appear below the scrolling list. This tells Canvas to use the magnification event in the Canvas suite.

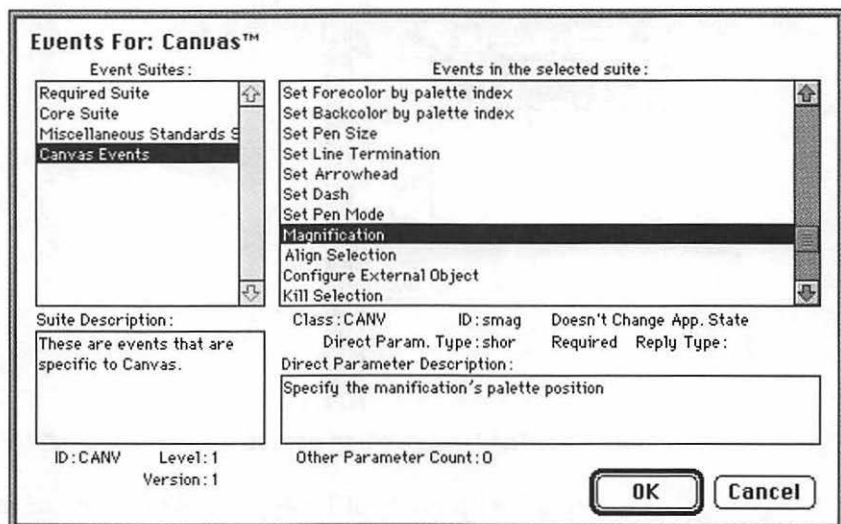


Figure 6-16: Selecting the magnification event for Canvas. Notice the class CANV and ID beneath the scrolling list of Canvas events.

- Click OK or press **Return** to close the dialog box. You'll see that the parameter information has been added to the Apple Events editor (Fig. 6-17).

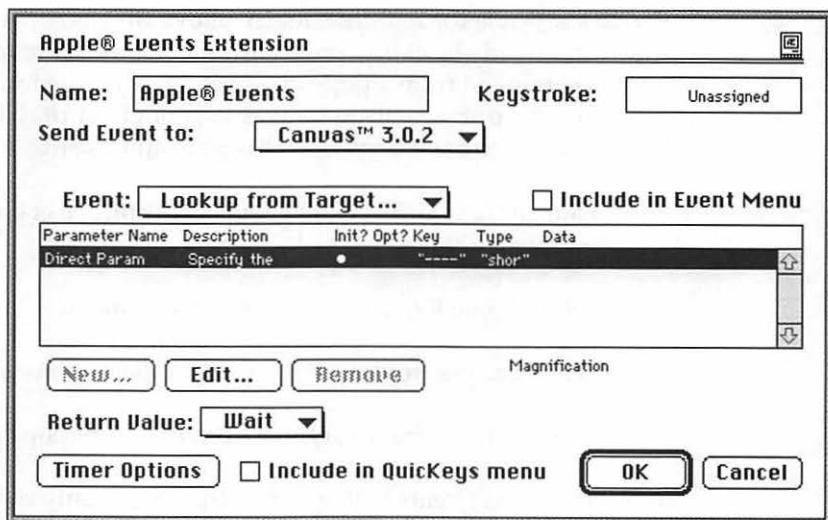


Figure 6-17: The Apple Events extension with the parameter information added.

7. Select the parameter information and click the Edit... button to open the Parameter editor (Fig. 6-18).

Figure 6-18: The Parameter editor for the 2xMag macro.

8. Enter 2 in the bottom text box and click OK or press **Return** to close the editor. The number 2 selects twofold magnification from Canvas's Zoom In/Zoom Out windoid.
9. Name your macro and assign a keystroke.
10. Click OK to close the Apple Events editor and click OK again to close QuicKeys.

You can select any magnification or reduction available in the palette using Apple Events. To select a different setting, enter a number that corresponds to the amount of magnification or reduction shown in Figure 6-19 in the Parameter editor. You can enter 1 through 14 to select one- through fourteenfold magnification or 15 through 32 to select any of the other settings.

15 16

Zoom In/Zoom Out																
Zoom In	1x	2x	3x	4x	5x	6x	7x	8x	9x	10x	11x	12x	13x	14x	16x	32x
Zoom Out	/1	/2	/3	/4	/5	/6	/7	/8	/9	/10	/11	/12	/13	/14	/16	/32

17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Figure 6-19: Canvas's Zoom In/Zoom Out windoid. To select a different magnification or reduction, enter one of the numbers along the top or bottom of the windoid in the Parameter editor.

Automatic Formatting

While you don't usually have a great deal of text to format in graphics programs, you will often find yourself setting the same attribute for several graphic objects or selecting the same tool settings in different drawings. For example, you may want all of the objects in your drawing to have the same line weight or background color. Instead of selecting each of the objects and then selecting the appropriate line weight from the pop-up palette, you could create a macro to select all of the items and then select the line weight. If you watch the way you work, you'll quickly see where you can automate the tasks you repeat two or three times a day. This section describes combining actions to automate selecting tool settings and changing text wrap in PageMaker.

Intelligent Tools

Like word processors, graphics programs sometimes require several steps before you get the settings for a tool where you want them. For instance, you may have to select the spray, select the spray shape, and then set the pattern and color you want to use. Some programs let you save the settings of tools, but those settings aren't always independent of the drawing you are working on. Whenever you begin a new drawing, you must re-create the tool setting or open the previous drawing to use the tool. QuickKeys allows you to create intelligent tools—tools that are automatically configured the way you use them. If you do a great deal of graphics work and have special settings for a number of your tools, this macro can save you the time it takes you to set up your tools each time you begin a new drawing.

Before we begin: Like the macros we created above, you'll need to use real-time macros to accurately record palette selections for most programs. You should have a keystroke assigned to the record real-time QuickKey. Open your graphics program and then open QuickKeys and select the program's name from the pop-up menu. Now position the pointer over the palette from which you want to make the selections. I'll be using Canvas for this example. We'll be recording a macro that selects the Spray tool, a pen pattern, and then a foreground color.

Create the macro:

1. Move the pointer over the Paint tool palette.
2. Press the keystroke you assigned to the record real-time QuickKey.

3. Mouse down on the Paint tool, drag out and to the Spray tool, and then release the mouse button (Fig. 6-20).

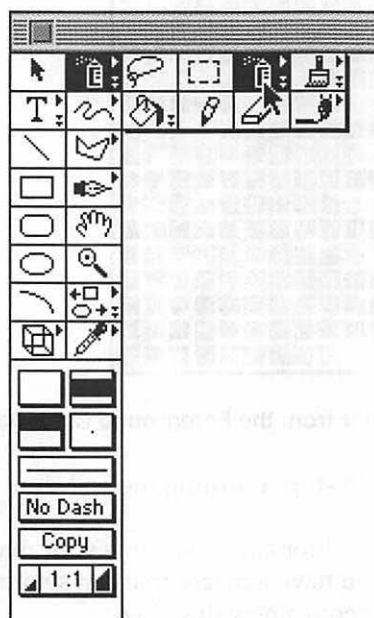


Figure 6-20: Selecting the Spray tool in Canvas.

4. Move the pointer down, select the Pen Pattern palette, and drag out to the fifth pattern (Fig. 6-21).

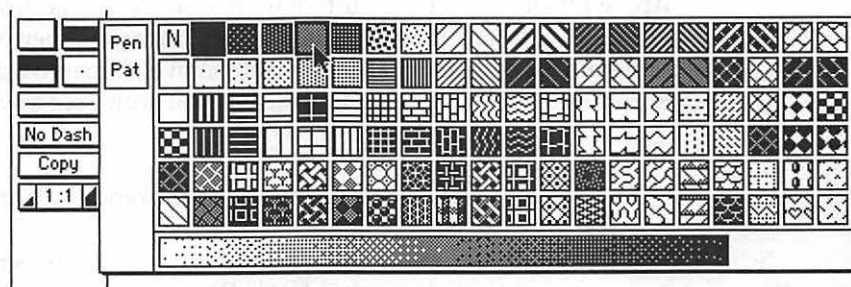


Figure 6-21: Selecting the fifth pattern from the Pen Pattern palette.

5. Move the pointer down, select the Foreground Color pattern palette, and drag over and down to the fifth row and second column (Fig. 6-22). If you want to count, that's the sixty-sixth color in the palette.

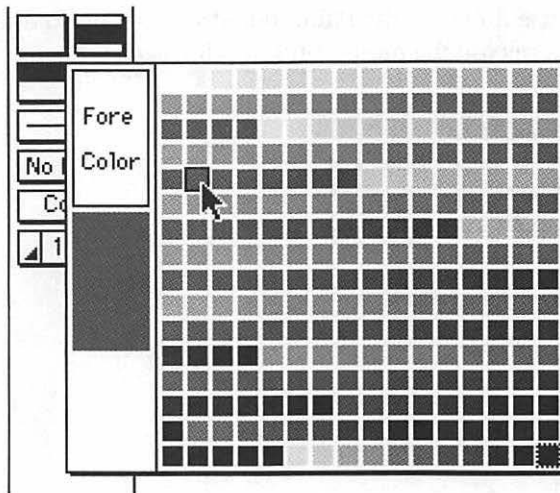


Figure 6-22: Selecting a color from the Foreground Color palette.

6. Press the keystroke to stop recording the real-time macro.
7. When the Real Time editor appears, name your macro and assign a keystroke. Now you have a macro that can set up your Spray tool whenever you begin a new drawing.
8. Click OK to close the Real Time editor and then click OK again to close QuickKeys.

Variations: If you use Canvas 3 and are running System 7, you can use Apple Events to set up your tool more quickly and accurately. For example, you can create a sequence for Canvas that performs exactly the same functions as the real-time macro above. A macro that uses Apple events doesn't take as long to run, and you won't see any palette selections.

1. Open QuickKeys and select Sequence from the Define menu.
2. When the editor opens, click the Import button. Scroll the list until you see the SelectSpray Apple Event macro you created earlier in the chapter and then click the Copy button.
3. Then select Apple Events from the Extensions menu.
4. Select Canvas from the Send Event to pop-up menu.
5. Select Lookup from Target... from the Event pop-up menu.

6. When the Events dialog box opens, select Canvas Events in the left list and then select Set Foreground by Index in the list on the right. Click OK to close the dialog. You'll see that the Apple Event information has been added to the dialog.
7. Select the Apple Event information and then click the Edit... button. When the editor opens, enter 66 in the bottom text box (Fig. 6-23). Now aren't you glad you knew how many colors down it was? Click OK to close the editor.

Parameter Number: 1

Name: Direct Param

Description: Specify the color's index

Key Word: ---- Type: shor Required

Integer ▼ shor

☒ Data: ☐ Get from Clipboard

66

OK Cancel

Figure 6-23: Selecting the foreground color.

8. Now you have completed two steps (Fig. 6-24) in setting up the Spray macro. Name your macro and click OK to close the Apple Events editor.
9. Select Apple Events from the Extensions menu again. When the editor opens, select Canvas from the Send Event to pop-up menu and then Lookup from Target... from the Event pop-up menu.
10. When the Events dialog opens, select Canvas Events in the left list and then select Set Pen Pattern from the list on the right. Click OK to close the dialog. You'll see that the Apple Event information has been added to the editor (Fig. 6-25).

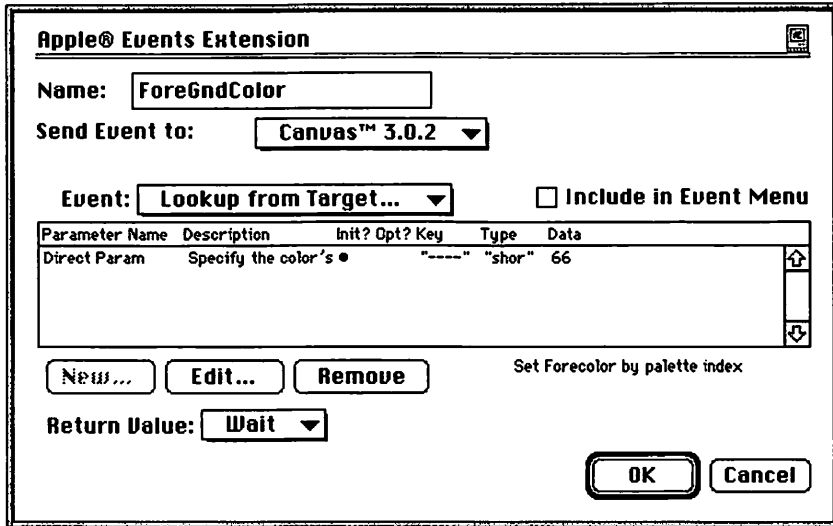


Figure 6-24: The completed Foreground Color Apple Event.

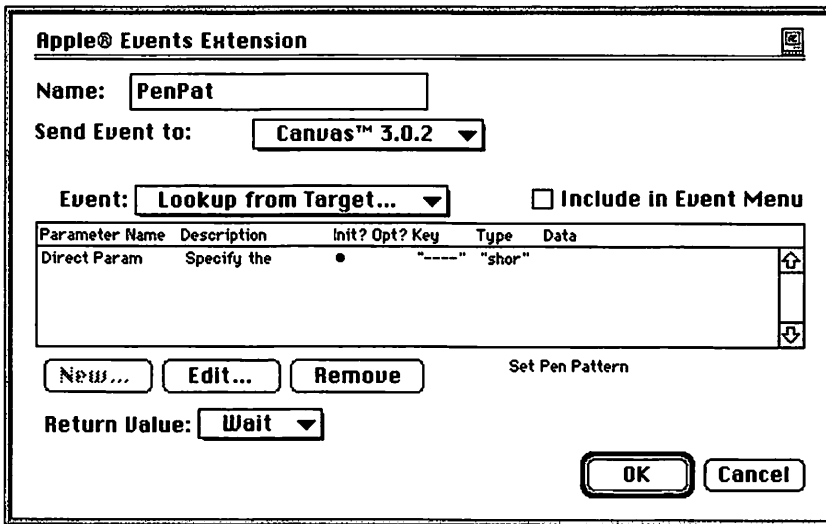


Figure 6-25: The Apple Event information for the Set Pen Pattern macro.

11. Select the Apple Event information and click the Edit... button. When the parameter editor opens, enter 5 in the bottom text box (Fig. 6-26). This will select the fifth pattern in the Pen Pattern palette (see Fig. 6-21).

Figure 6-26: The Parameter editor for the PenPattern macro.

12. Click OK to close the Parameter editor. Name your Apple Event macro and click OK again to close the Apple Events editor. Now your completed macro should look like Figure 6-27.

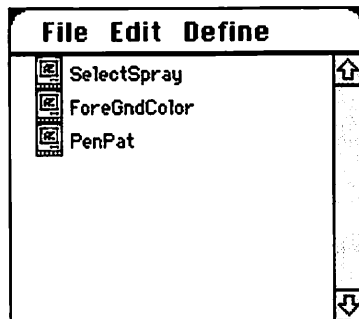


Figure 6-27: The completed SetUpSpray macro.

13. Click OK to close the Sequence editor and click OK again to close QuicKeys.

This doesn't look like a very complicated macro, but it packs a lot of power. With only three Apple Events you accomplish the same steps that you did in the longer real-time macro above. More importantly, the Apple Event version of the macro is consistently accurate.

Wrap Text

If you use PageMaker for laying out brochures or newsletters, you have probably mixed text and graphics and may even have wrapped text around the graphics. While it isn't a complicated process, it can be tedious if you have to do it very often. For instance, you must open the Text wrap dialog box, click a couple of buttons, and enter the amount of space you want the text to stand off from the graphic. This series of steps is easily reduced to a single keystroke.

Before we begin: You'll need to open PageMaker and a file that contains a mixture of graphics and text. If you don't already have a graphic placed in the middle of a block of text, you should move one there. You should also open QuickKeys and select PageMaker from the pop-up menu.

Create the macro:

1. Select the graphic that you want to wrap the text around. You do this first to activate PageMaker's Text Wrap... command.
2. Select Record Sequence from the QuickKeys menu.
3. Select Text wrap... from PageMaker's Element menu.
4. When the Text wrap dialog box opens, click on the center page icon under the Wrap option. You'll notice that the last page icon under Text flow is selected and a default standoff amount is entered in the four Standoff in inches text boxes (Fig. 6-28).

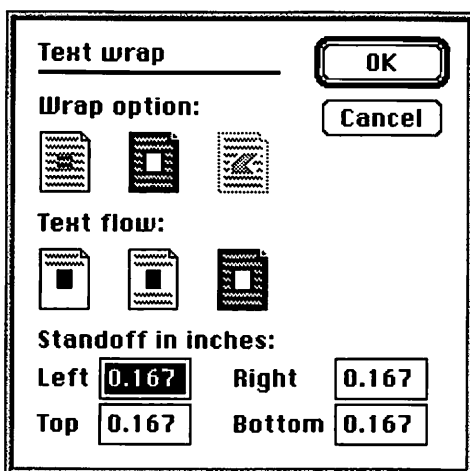


Figure 6-28: PageMaker's Text wrap dialog box.

5. Click the last Text flow icon. Since you may want to use the macro to format graphics with different text flow, clicking on this icon ensures that the text flow is always around the graphic.
6. Press **Tab** to select the text in the Right text box, enter .1 and press **Tab** to move to the next text box. Enter .1 in each text box.
7. Press **Return** to close the Text wrap dialog box and format the graphic.
8. Select Stop Recording... from the QuicKeys menu. When the Sequence editor opens, your macro should look like the one in Figure 6-29. If you open the text QuicKey, you'll see that the **Return** in step 7 is in it.

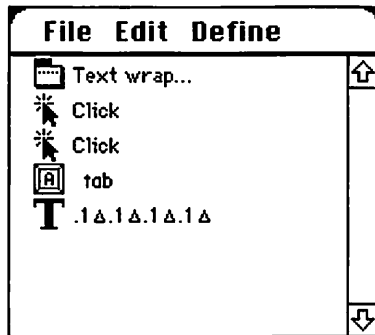


Figure 6-29: The final TiteWrap macro.

9. Name your macro and assign a keystroke. Click OK to close the Sequence editor and then click OK again to close QuicKeys.

Variations: A simple variation on our TiteWrap macro would jump the text over the graphic. Instead of clicking the last Text flow icon, you would click the one in the center (Fig. 6-30).

You would also be able to remove the **Tab** and standoff information from the end of the macro since it would no longer apply. When you do this, you'll need to add a **Return** alias to close PageMaker's Text wrap dialog box. The final version would look like the one in Figure 6-31.

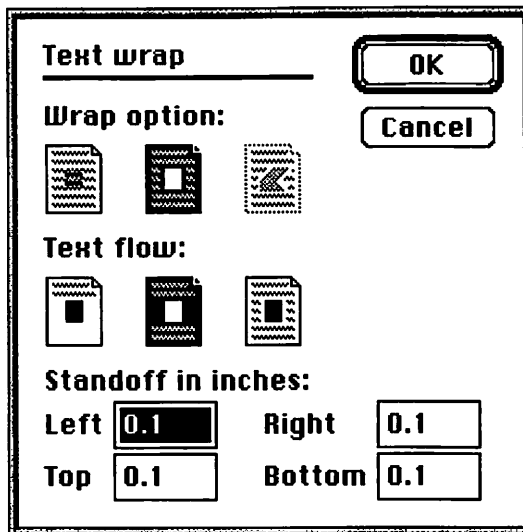


Figure 6-30: The Text wrap dialog box configured to jump text over a graphic.

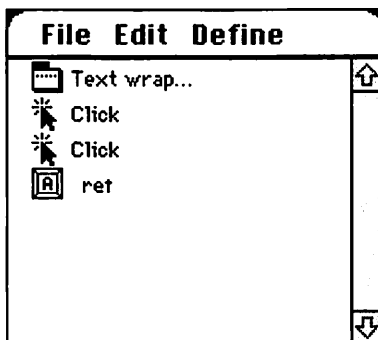


Figure 6-31: The WrapText macro.

Smart Functions

If you are creating art to place in a PageMaker document, you can create a macro that opens PageMaker and activates the Place command. While using the Place command may seem more complicated than just pasting your graphic into the PageMaker document, the command can save you time in the long run. If you later make a change to your graphic and haven't used the Place command, you will have to repaste the graphic into your document. If, on the other hand, you have used the Place command, you can use PageMaker's Update Links to automatically bring in the changed graphic. This macro will eliminate the extra steps

of switching to PageMaker, pressing **⌘-D** to active the Place command, and then navigating to the folder containing your graphics. This macro only works with System 6 and MultiFinder or with System 7.

Before we begin: You can record this macro for either System 6 or 7.

If you are running either MultiFinder or System 7 and you have enough memory, you may want to open both your graphics program and PageMaker (refer to the variation if you don't have enough memory to keep both programs open). If you have both programs running, switch to the graphics program. Open QuickKeys and select the name of your graphics program from the pop-up menu.

Create the macro:

1. Select Record a Sequence from the QuickKeys menu.
2. Under System 7, use the applications menu on the right side of the menu bar to switch to PageMaker. If you are running MultiFinder, you can switch to PageMaker using the Apple menu. We are starting in the graphics program because you would usually switch over to PageMaker from it to place your graphic.
3. Select the Place command from the File menu.
4. When the Place dialog box opens, you could navigate to the folder where you keep your graphics (Fig. 6-32). However, it is much more accurate to use a location QuickKey in the macro. Click the Cancel button.

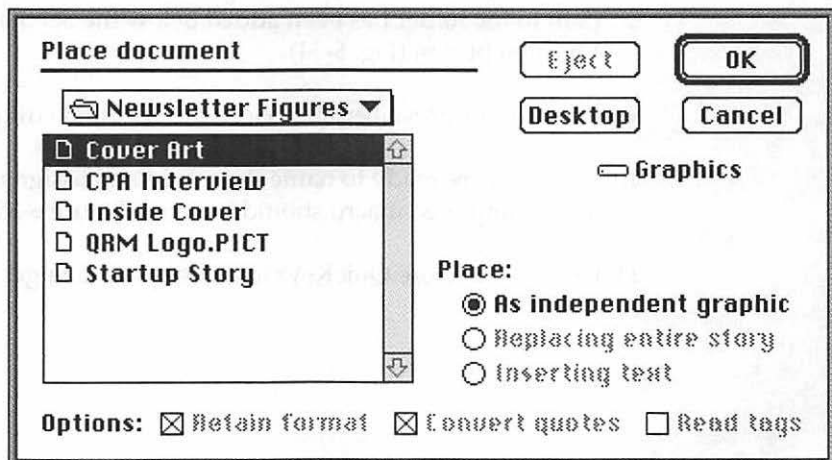


Figure 6-32: PageMaker's Place dialog box.

5. Select Stop Recording from the QuickKeys menu.
6. When the Sequence editor opens, select Location from the Extensions menu.
7. When the Location editor opens, click the Select Location button to open the Select location dialog box. Navigate to the folder where you have stored your graphic files and click the Select button (Fig. 6-33).

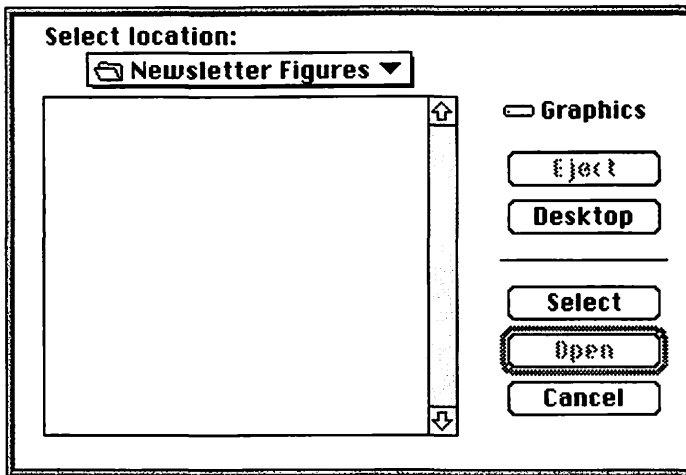


Figure 6-33: Using the Select Location dialog box to tell QuickKeys in which folder to open the Place dialog.

8. When the Select Location dialog box closes, you'll see that the path to the folder has been added below the Set Standard File Location button (Fig. 6-34).
9. Click OK or press **Return** to close the Location editor.
10. Now you are ready to name the macro and assign a keystroke. Your completed macro should look like Figure 6-35.
11. Click OK to close QuickKeys and save your changes.

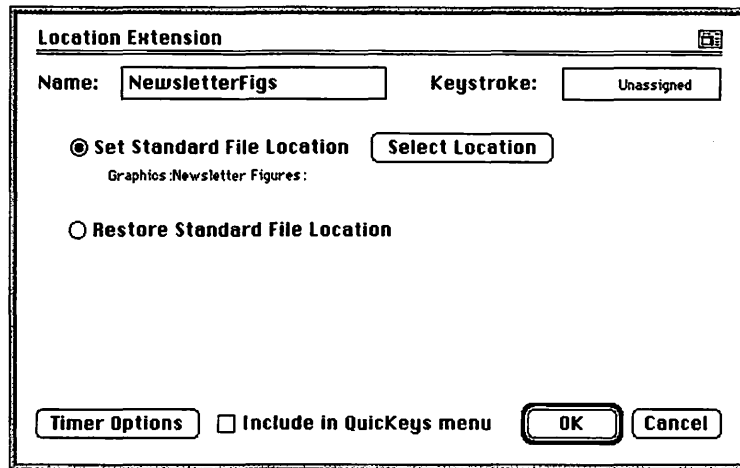


Figure 6-34: The Location editor with the path to the folder added.

Use and variations: If you don't have enough memory to keep PageMaker and your drawing program open, you could add an Open Finder Event to the macro to open PageMaker with the appropriate document and activate the Place command. If you are running System 6, you could use a file QuickKey. Since you would need to close the graphics program before you open PageMaker, you should add steps at the beginning of the macro to save your changes and quit the program (Fig. 6-35).

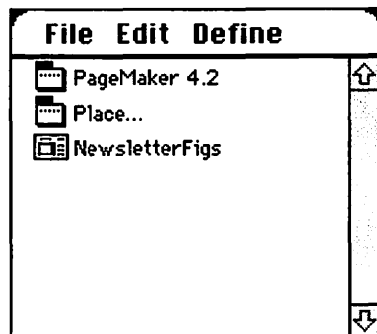


Figure 6-35: The Quit&Place macro with the changes to quit the current application and launch PageMaker.

To create the Quit&Place macro so it will close your graphics program and launch PageMaker for you:

1. Open QuickKeys and the AutomatedPlace macro.

NOTE: If you don't want to change the original macro, select the macro, select Copy from the Edit menu, and then select Paste from the Edit menu. You'll see that a second copy of the macro has been pasted into the QuickKeys window next to the original. You can make your changes to this copy. Whenever you make a copy of an original macro, the first thing you should do when you open it is change the name so you won't confuse your modified macro with the original.

2. Move the insertion arrow to the top of the macro.
3. Select Alias from the Define menu and press **⌘-S**. This alias QuickKey saves any changes you have made to your document.
4. Select Alias from the Define menu again and press **⌘-Q**. This alias QuickKey quits your graphics program and prepares to launch PageMaker.
5. Select Finder Events from the Extensions menu and then select Open from the pop-up menu when the Finder Events editor opens. A dialog box appears where you can select the file you want to open PageMaker with (Fig. 6-36). If you are running System 6.0x, you will need to use a file QuickKey.

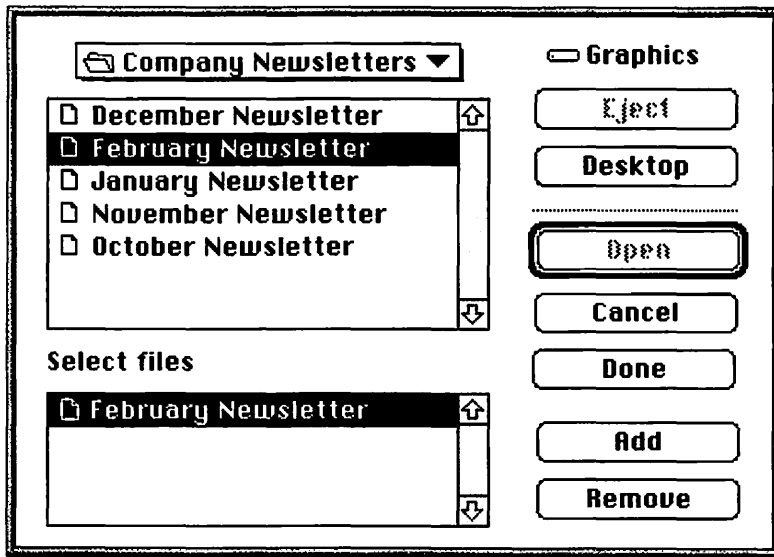


Figure 6-36: The Open dialog in the Finder Events editor.

6. Navigate to the folder containing the file you want to open, select the file in the top list, and click the Add button. The name of the file is added to the bottom list. Click Done to close the dialog box. You'll see that the file information has been added to the editor window (Fig. 6-37).

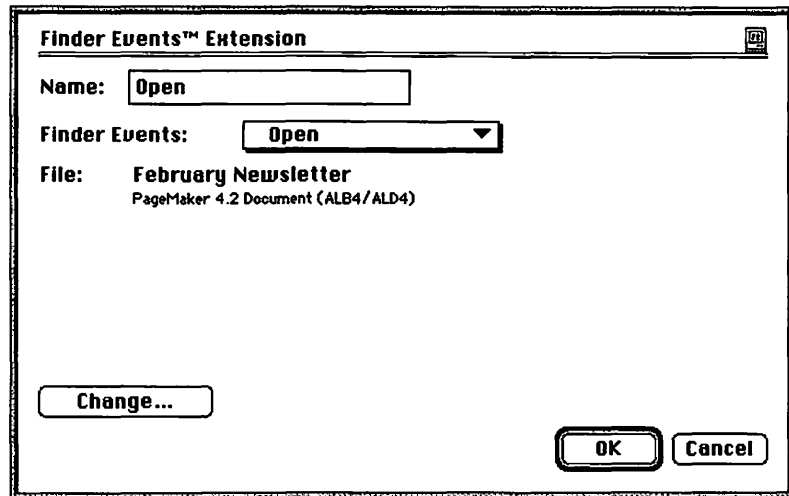


Figure 6-37: The Finder Events editor window with the file information added.

7. Now we just need to make the macro pause until PageMaker is completely loaded. Then we can select Place from the File menu. We'll use WindowAction to pause the macro.

Select WindowAction from the Extensions menu. We could enter the name of the file in the Window Name text box, but what you would discover is that the macro would continue to play before PageMaker is ready to accept a menu selection. A more reliable approach is to pause the macro until the About and loading progress dialog boxes close.

Click the Check Window Name checkbox to deselect it. Select Not Dialog from the Window Type pop-up menu and click the Check Window Type checkbox to select it. Then click the Wait... button. This QuickKey waits until the first dialog closes. Name the QuickKey and click OK to close the editor.

8. Select WindowAction from the Extensions menu again. When the editor opens, configure it exactly as you did the one above

(Fig. 6-38). This QuicKey waits until the progress dialog box closes. Name the macro so you can distinguish it from the one above and click OK to close the editor. Then click OK to close QuicKeys and save your changes.

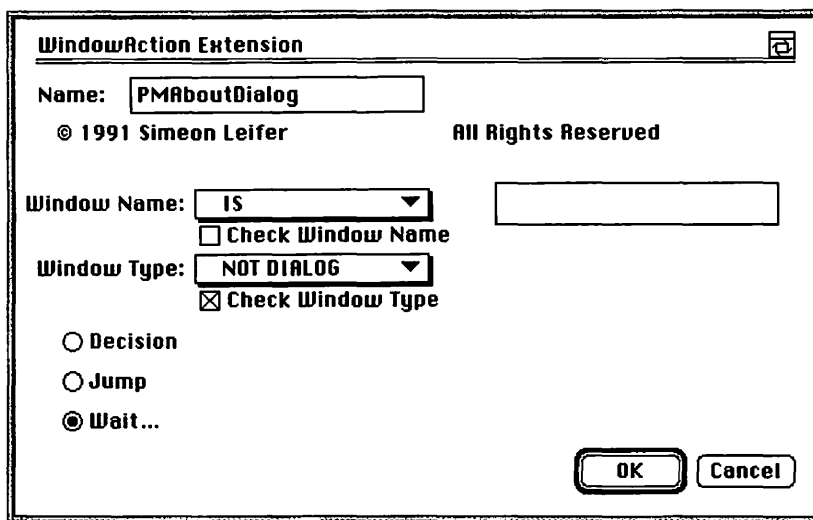




Figure 6-38: The WindowAction editor which waits until PageMaker's About dialog box closes.

Now you have a variation of the original macro that will automatically run PageMaker and open the Place dialog. If you wanted to go back to your graphics program automatically, you could add a pause and wait for user at the end of this macro and another Finder Event to launch your graphics program. When QuicKeys reached the Place dialog, it would pause the macro and wait until you continued the macro. This would give you all the time you needed to place your graphic since you have to resume the macro.

If you have a number of graphics to place in your PageMaker document, you could add a repeat to the AutomatedPlace macro that would automatically select the next graphic in the folder and place it in your PageMaker document.

You would delete the PageMaker menu QuicKey from the beginning of the macro and then use the initial Place command to get the repeat started. You would add a **Return** after the location QuicKey to open the first graphic in the folder. You would insert a repeat that asks the user how many times to play the repeating portion of the macro. Then you

would copy the first two QuicKeys and add a pause, an  alias, and another . This would be the portion of the macro that selects and opens the next graphic. A pause and wait for user at the end of the repeat would give you plenty of time to place and resize the graphic (Fig. 6-39). Then you would only have to move to the next page that needed a graphic and select Pause from the QuicKeys to continue playing the macro.

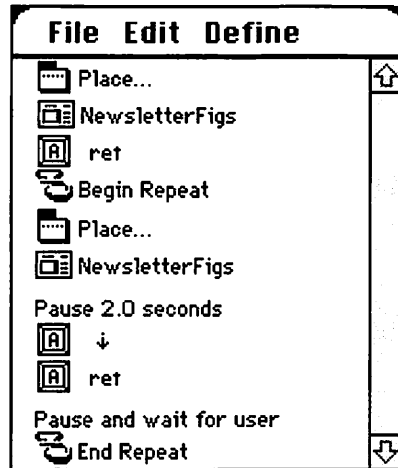


Figure 6-39: The RepeatingPlace macro.

NOTE: *If you are playing this macro, don't try to play the TiteWrap macro at the same time. QuicKeys doesn't like that.*

What's Next?

Now you see how QuicKeys can automate your graphics work. This chapter only scrapes the surface of possibilities. Only your work habits and imagination will dictate how far you can take QuicKeys. In the next chapter, we will create macros for HyperCard that simplify scripting. There are also macros for creating buttons with the same attributes, setting up a card, and navigating between stacks.

7

CHAPTER

QuicKeys and HyperCard

Most Macintosh users are familiar with HyperCard. You can think of HyperCard as an electronic card filer. HyperCard files are called stacks, and each stack is a collection of electronic cards. Since its release, HyperCard has defied categorization but has been wholeheartedly adopted by the Macintosh community. It has been used as a database, a front-end for multimedia presentations, an index tool for CD-ROMs, and many other things. No matter what you use HyperCard for, QuicKeys will help you get more out of HyperCard.

This chapter won't make you a better HyperCard scriptwriter. However, it will show you how to use QuicKeys to make your time working with and creating stacks and writing scripts more efficient. Besides creating macros to "improve" HyperCard's user interface, this chapter outlines ways you can automate scripting and building stacks. While all of the macros in this chapter have been created for HyperCard 2.0, they also may be applicable to Aldus's SuperCard, a HyperCard-like program. If you are using HyperCard 1.2.x, you may be able to use or adapt most of the macros in this chapter.

Personalizing HyperCard

As we have seen in earlier chapters, QuicKeys allows you to add features the programmers have left out. In the first part of this chapter, we'll create macros that add keyboard shortcuts to the menu items in HyperCard that don't have any. QuicKeys also allows you to modify features in a program so that it better suits the way you work.

Menu Shortcuts

Not all of the menu items have shortcuts, and some of the commands really need them. If you are checking the stack and card information for several stacks, selecting the Stack Info and Card Info commands again and again can become maddening. You can create Menu QuickKeys to select these and the other HyperCard commands that don't have keyboard shortcuts.

Before we begin: Let's create several menu QuickKeys at once. I know that I want shortcuts for all of the Info commands. You don't need to worry about selecting a button or field to activate the Button Info and Field Info menus since QuickKeys allows you to select a menu item that is dimmed. You should launch HyperCard if you haven't already.

Create the macro:

1. Open QuickKeys and select HyperCard from the pop-up menu.
2. Select Menu/DA... from the Define menu.
3. When you are prompted to select a menu item, select Button Info from the Objects menu. QuickKeys opens the Menu editor with the menu item information already entered (Fig. 7-1). You can leave all of the settings the same and close the editor, but you might want to select the Don't complain if the menu choice can't be found checkbox.

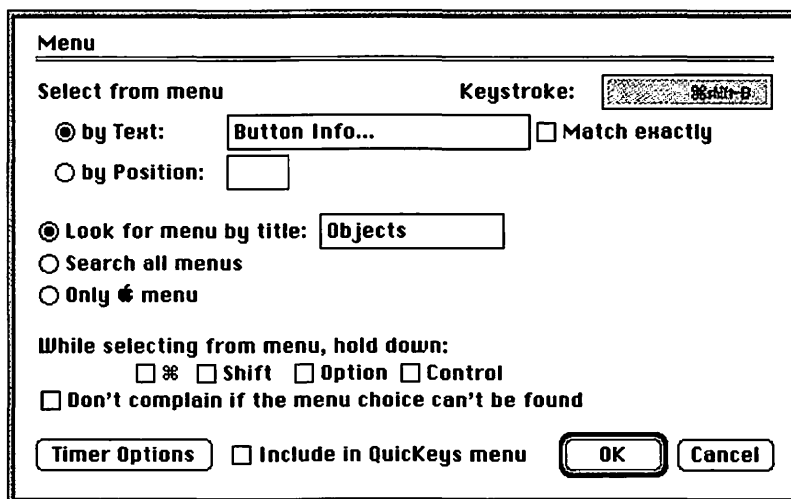


Figure 7-1: The Menu editor for the Button Info... command. I've already assigned a keystroke.

You may remember that QuicKeys displays a dialog box informing you that it can't find a menu item (Fig. 7-2). This can serve as a reminder that you need to select the Button tool. If, however, you think that it is just an annoyance because you would never select the Button Info command without first selecting the Button tool, then select the Don't complain checkbox in the Menu editor.

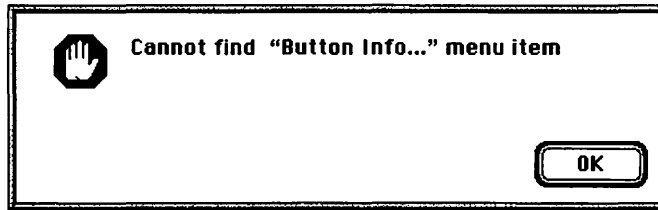


Figure 7-2: The Cannot find menu item alert dialog box.

4. Assign a keystroke to the QuicKey. It is a good idea to assign similar keystrokes to all of the Info commands. For instance, you might want to assign ⌘-Shift-B to the Button Info... command, and use the same modifier keys with the initial letter of the other commands. You will only need to choose a different mnemonic for the Background Info... command.

You should also remember that HyperCard's keyboard shortcut for Find Whole is ⌘-Shift-F . You can choose a different modifier combination like ⌘-Option , but you will discover that combination also conflicts with some of HyperCard's built-in keyboard shortcuts. ⌘-Option-B opens the Background script window, ⌘-Option-C opens the Card script window, and ⌘-Option-S opens the Stack script window.

Since I use the Info commands more than I use the Find Whole command, I've used the ⌘-Shift modifier combination. I've then reassigned the Find Whole command to ⌘-Option-F using an alias QuicKey (Fig. 7-3). This assignment doesn't conflict with any of HyperCard's built-in shortcuts, and I find it no more difficult to remember than ⌘-Shift-F .

5. Click OK to close the Menu editor.
6. Now create menu QuicKeys for the Field Info, Card Info, Background Info, and Stack Info commands. When you are through, click OK to close QuicKeys and save your changes.

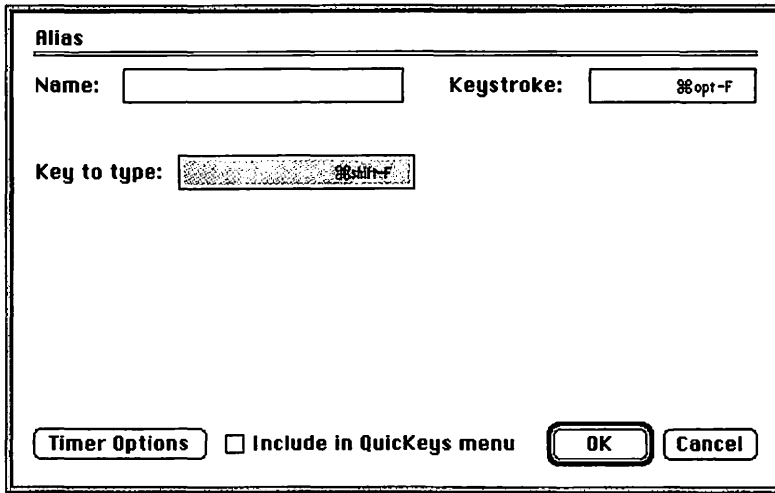


Figure 7-3: The FindWhole alias QuickKey that lets me shift the Find Whole command to a different combination of modifier keys.

Variations: While the New Field and New Button commands do not have keyboard shortcuts, there is a quick way to create a new field or button. You can select the Field tool by pressing **⌘-Tab-Tab-Tab** and then you can hold down the **⌘** key and draw a new field by dragging the cursor on the card. You can create a new button by pressing **⌘-Tab-Tab** to select the Button tool and then drawing a button using the same technique. If you create menu QuickKeys for the New Field and New Button commands, they automatically create a field or button for you and select either the button or the field tool.

Selecting Tools or Patterns

HyperCard has a set of keyboard shortcuts that are activated by clicking the PowerKeys checkbox on the Preferences card in the Home stack or by selecting PowerKeys from the Options menu in HyperCard. (Remember, you have to select one of the paint tools before you can see the Options menu.) The PowerKeys feature allows you to select some patterns such as white and black and lets you select some of the painting effects such as rotate, trace edges, and invert. The PowerKeys feature does not, however, have shortcuts to select all of the tools and patterns.

You can create macros that select each of the tools and patterns from the palettes, just as we did with graphics programs in Chapter 6. When it comes to creating click QuickKeys to select tools in HyperCard, you have

a slight advantage over most graphic programs. HyperCard's tool and pattern palettes are named, so click QuickKeys are sent to the correct palette regardless of its location on screen. This makes your macros more accurate. As in other graphics programs, the click QuickKey works better if the palette is open on screen. You can "tear" the palettes from HyperCard's menu bar by selecting the palette and then continuing to drag past the end of the palette. You can also open the tool palette by pressing **Option-Tab** and you can display the pattern palette by pressing **Tab** after you have selected a drawing tool.

Before we begin: Open the tool palette using one of the methods described earlier.

Create the macro:

1. Select Record one QuickKey from the QuickKeys menu.
2. Click on the Marquee selection tool in the upper-left corner.
3. When the Click editor opens, you may want to click on the Window button to verify that QuickKeys recorded the click in the tools window (Fig. 7-4).

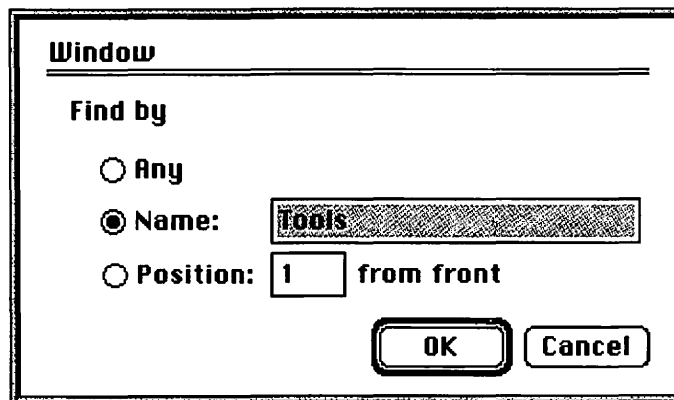


Figure 7-4: The Window settings for the ClickMarquee macro.

4. Name the click QuickKey something descriptive and assign a keystroke (Fig. 7-5).
5. Click OK to close the Click editor.

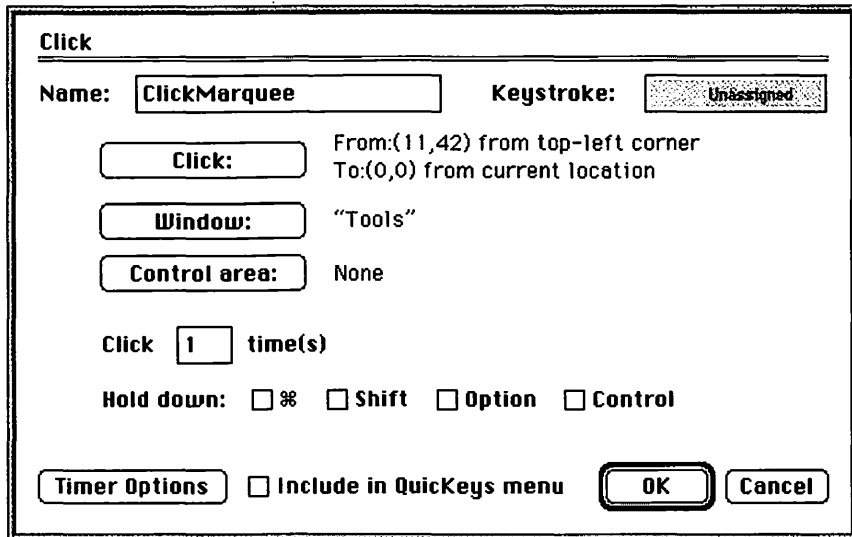


Figure 7-5: The ClickMarquee macro.

ASIDE: You could assign a keystroke based on the initial letter of the tool. **C**trl-M, for instance, would be an easy shortcut to remember. Since the tool and pattern palettes have a grid similar to the numeric keypad, you could also assign a keystroke based on the correspondence of the tool's location on the palette and the numeric keypad (Fig. 7-6). For instance, you could assign the keystroke **C**lear to the Marquee tool, [=] to the Lasso, and [/] to the Pencil (you wouldn't need to assign keystrokes to the Browse, Button, or Field tools since they already have shortcuts). Of course, the first thing you will notice is that there isn't a one-to-one correspondence between the tools and the keys. While you don't have a mnemonic connection between the tools and the keys on the numeric keypad, you do have a visual connection.

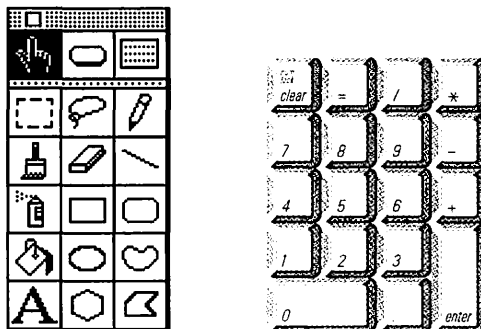


Figure 7-6: The HyperCard tools palette and a possible shortcut layout for the numeric keypad of an extended keyboard.

Variations: We only started by creating a macro to select the Marquee tool, but you can create macros to select all of the tools and patterns. If you have a Macintosh Plus, SE, or Classic and don't really have enough room to leave the tool or pattern palette open, you could create a sequence that opens the appropriate palette and then selects the tool or pattern. To create this sequence variation of this macro:

1. Select Record Sequence from the QuickKeys menu.
2. Press **Option-Tab** to open the tool palette.
3. Click the Marquee tool.
4. Press **Option-Tab** to close the tool palette.
5. Select Stop Recording... from the QuickKeys menu.
6. When the Sequence editor opens, your macro should look like Figure 7-7. Name your sequence and assign a keystroke.

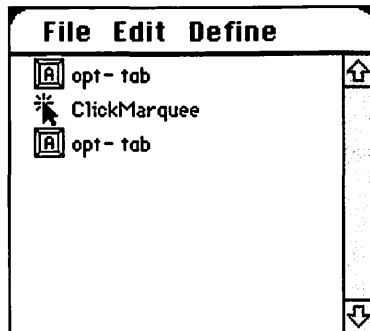


Figure 7-7: The SelectMarquee sequence. I've named the click QuickKey so I won't forget what it clicks.

7. Click OK to close the Sequence editor and then click OK again to close QuickKeys.

Navigating HyperCard

HyperCard has several keyboard shortcuts for moving between cards within a stack. **⌘-2** takes you back to the previous card, **⌘-3** takes you to the next card, **⌘-1** takes you to the first card in a stack, and **⌘-4** takes you to the last card in a stack. By force of habit, I've learned these shortcuts, but you can change these shortcuts to something easier

to remember with QuickKeys. For example, you could assign the **⌘PgUp** and **⌘PgDn** keys on the extended keyboard for moving to the previous card and to the next card. These commands would be a single keystroke compared to the existing **⌘-2** and **⌘-3** shortcuts.

Before we begin: Open QuickKeys and select HyperCard from the pop-up menu.

Create the macro:

1. Select Record one QuickKey from the QuickKeys menu.
2. Select Prev from HyperCard's Go menu.
3. When the Menu editor opens, leave the settings as they are and press **⌘PgUp** on your extended keyboard. QuickKeys warns you that another QuickKey has the same keystroke assigned to it (Fig. 7-8). You can assign **⌘PgUp** twice because the HyperCard keyset will take precedence over the Universal keyset. That is, the macro in the application keyset gets played, and the macro in the Universal keyset doesn't.

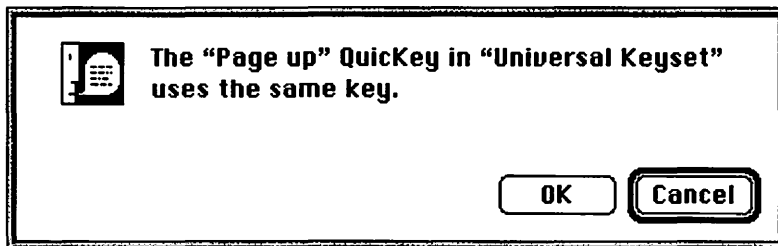


Figure 7-8: QuickKeys warns you if you are about to assign the same keystroke to two different keys.

4. Click the OK button to continue. Your Prev menu QuickKey should look like Figure 7-9. Click OK to close the Menu editor.
5. Select Menu/DA... from the Define menu.
6. When you are prompted to select a menu item, choose Next from the Go menu.
7. Leave the settings as they are when the dialog box opens and assign **⌘PgDn** for the keystroke. Again QuickKeys warns you that you have already assigned the keystroke to another QuickKey. Click OK to close the warning dialog.

Menu

Select from menu Keystroke:

☒ by Text: ☒ Match exactly

☐ by Position:

☒ Look for menu by title:

☐ Search all menus

☐ Only menu

While selecting from menu, hold down:

☐ ☐ Shift ☐ Option ☐ Control

☐ Don't complain if the menu choice can't be found

☐ Include in QuickKeys menu

Figure 7-9: The settings for the Prev QuickKey.

8. Your menu QuickKey should look like Figure 7-10.

Menu

Select from menu Keystroke:

☒ by Text: ☒ Match exactly

☐ by Position:

☒ Look for menu by title:

☐ Search all menus

☐ Only menu

While selecting from menu, hold down:

☐ ☐ Shift ☐ Option ☐ Control

☐ Don't complain if the menu choice can't be found

☐ Include in QuickKeys menu

Figure 7-10: The settings for the Next QuickKey.

9. Click OK to close the Menu editor.
10. Click OK to close QuickKeys.

Variations: In addition to the **[PgUp]** and **[PgDn]** macros, you can also create a macro for the First command under the Go menu (it selects the first card in the stack) and assign **[Home]** to it. And you can create a macro for the Last command under the Go menu (it selects the last card in the stack) and assign **[End]** to it.

Open Home in a New Window

One of the additions to HyperCard 2 was the ability to open more than one stack at a time. You can do this by selecting the Open stack in a new window checkbox in the open dialog box (Fig. 7-11). Or you can use the keyboard shortcut **[⌘]-[Shift]-O** to display the open dialog box with the checkbox already selected.

Often I need to open my Home stack while I am working in another stack, but I forget and press the **[⌘]-H** shortcut. This closes the stack I am in and then opens the Home stack. I've created a macro that is designed to open the Home stack in a new window every time using HyperCard's Open stack in a new window feature.

Before we begin: You should have HyperCard open and any stack but your Home stack open.

Create the macro:

1. Open QuicKeys.
2. Select Sequence from the QuicKeys File menu.
3. When the Sequence editor opens, click the Literal button and then press **[⌘]-[Shift]-O**. This shortcut displays the open dialog box in HyperCard with the Open in new window checkbox already selected (Fig. 7-11).
4. Select Location from the Extensions menu. When the Location editor opens, click the Select Location button. After the dialog box opens, navigate to your HyperCard folder, open it, and then click the Select button. When the dialog box closes, you will see the path to your HyperCard folder below the Set Standard File Location radio button (Fig. 7-12).

This causes QuicKeys to switch to your HyperCard folder whenever the macro runs. Normally, your Home stack is in the same

folder with HyperCard, and adding the location QuickKey ensures that the macro will always find the Home stack.

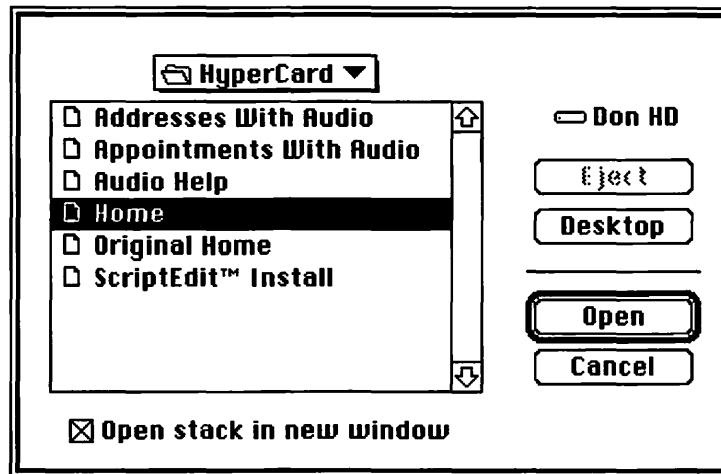


Figure 7-11: The open dialog box with the Open stack in new window checkbox selected.

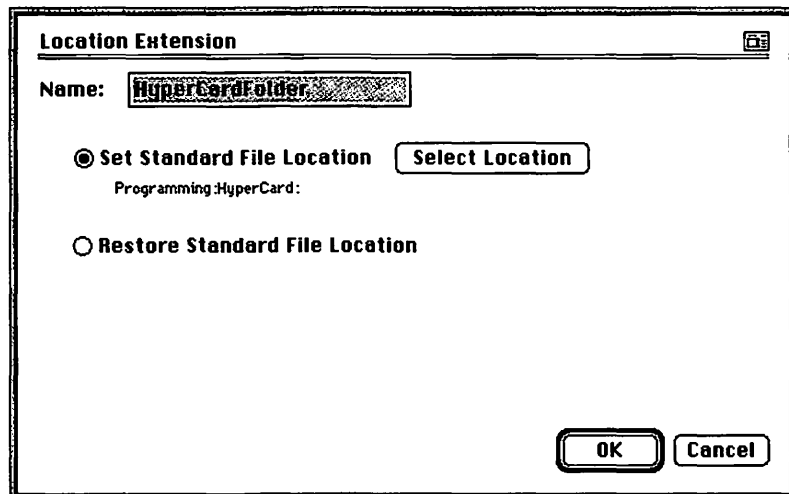


Figure 7-12: The Location settings for the OpenHome macro.

5. Select Text... from the Define menu. When the Text editor opens, click in the large text box and type "home," but don't type a Return. When you play this macro, this text QuickKey types "home" in the open dialog box. This is more accurate than trying to scroll and click on the Home stack. Click OK to close the Text editor.

6. Click the Literal button and press **Return**. When you play the macro, the **Return** clicks the OK button after the macro selects the Home stack. Your final macro should look like Figure 7-13.

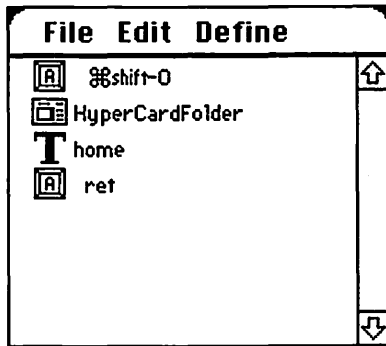


Figure 7-13: The completed OpenHome macro.

7. Name your macro and assign a keystroke. Click OK to close the Sequence editor and then click OK to close QuickKeys.

Variations: You could create another variation of this macro that opens any stack you work with on a regular basis in a new window. For example, you could change the macro so it opens the Power Tools stack in a new window. You might also want to extend this macro so it opens the stacks you are currently working in whenever you launch HyperCard.

Automating Your HyperCard Sessions

You can also use QuickKeys to automate any repetitive task. If you create stacks on a regular basis or spend time writing scripts, you can make macros that give all of your buttons or fields the same attributes. You can also create template macros that simplify writing scripts.

Scripting Shortcuts

SuperCard has a number of innovations over HyperCard beyond the obvious one of letting you enhance your stacks with color. One SuperCard feature I miss when using HyperCard is the five pop-up menus along the top of the script editing windows (Fig. 7-14). The **Ctrl** Structs, Commands, Functions, Sys Messages, and Properties menus allow you to enter templates for message handlers or functions quickly by selecting them from the pop-up menus. A HyperCard add-on called

ScriptEdit2, from the now defunct Somak Software, brought the same capabilities to HyperCard.

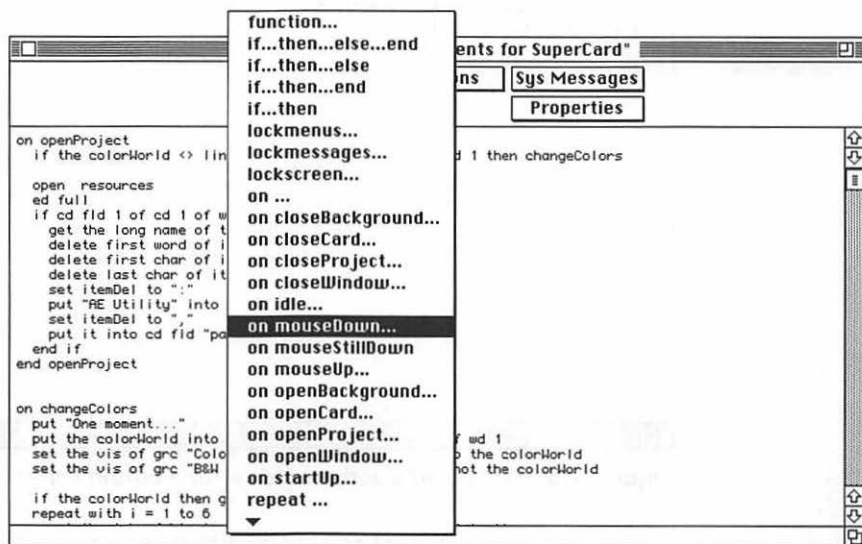


Figure 7-14: A SuperEdit script window with the Ctrl Structures pop-up menu extended.

Though you can't add pop-up menus to the top of the HyperCard script edit window with QuicKeys, you can create templates for message handlers and functions. Whenever you want to add a mouseUp handler to your button script, for example, you would only have to press a keystroke and the handler would be instantly inserted. These templates save time since the macro enters most of the script for you, and you only have to make a few modifications.

Before we begin: You will need to open a script window in HyperCard 2.0. You can open a button script by pressing **Option-⌘** to display the button outlines and then click the button. The script window appears and may contain a button script (Fig. 7-15). Move the insertion point below any existing script.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Type the first line of the handler: on mouseUp.
3. Press **Return** to move to the next line. We'll leave this line blank since it will contain the body of the handler.

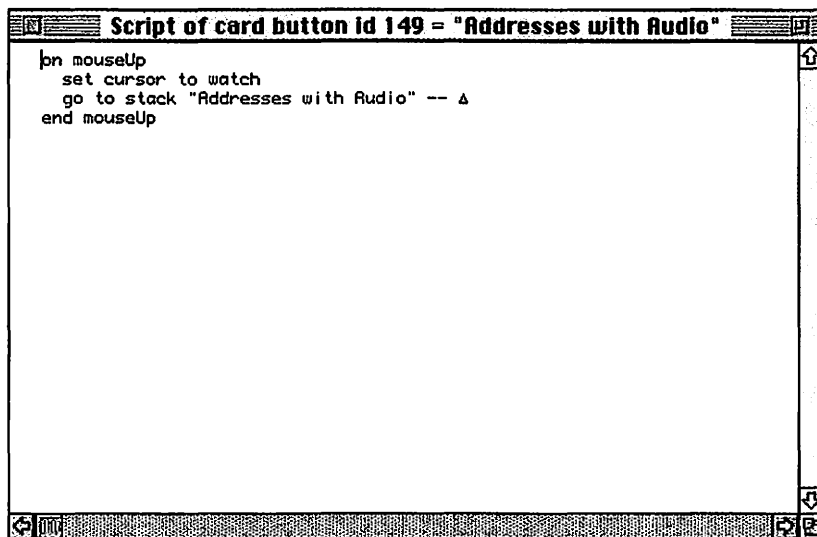


Figure 7-15: HyperCard's script window for a button script.

4. Press **Return** again to move the insertion point to the next line. Now we're ready to enter the last line of the handler.
5. Type the last line: "end mouseUp." Don't press **Return** after this line.
6. We can add a little polish to this template by moving the insertion point so it is in position for you to enter the body of the handler. Press the **↑**. This step moves the insertion point so it is on the blank line that we left above.
7. Press **Tab**. This step adds the appropriate indentation for the script.
8. Select Stop Recording... from the QuicKeys menu. When the Sequence editor opens, your macro should look like Figure 7-16.
9. Double-click the text QuicKey. You can look at the text of your template; it should look like Figure 7-17. Click OK to close the Text editor. If you backspaced while you were creating the QuicKey, you can clean up the QuicKey so it will run faster.
10. Name your macro something descriptive like "OnMouseUp" and assign a keystroke.

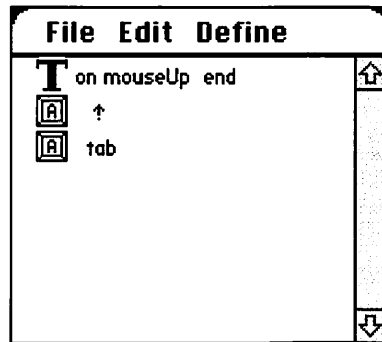


Figure 7-16: The final OnMouseUpTemp macro.

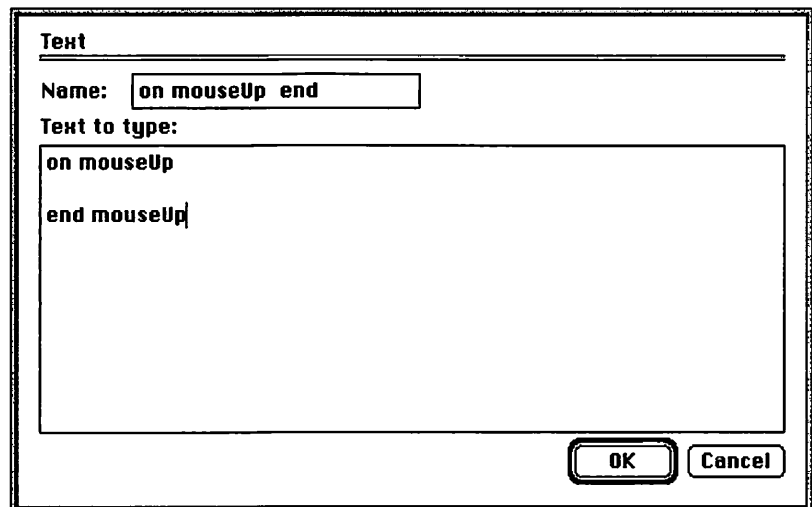


Figure 7-17: The text QuickKey for the OnMouseUpTemp macro.

11. Click OK to close the Sequence editor. Then click OK again to close QuickKeys.

Variations: Naturally, you can create a template macro that enters a function or a message in the script window. You can even have the macro enter some of the settings to jog your memory. For instance, you could create a function template that includes position markers for the function name and parameters (Fig. 7-18). You would then only have to select the position markers and enter the settings appropriate for your script.

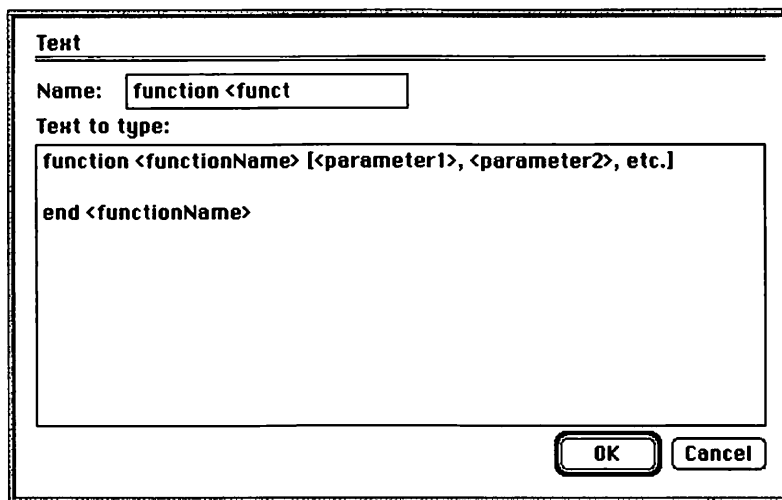


Figure 7-18: The text QuickKey for the FunctionTemp macro.

A logical way to extend this macro is to have it select the first `<functionName>` marker since it is probably the first item you will want to change in the template. Unfortunately, the script editor in HyperCard does not have any keyboard shortcuts for selecting text. Heizer Software has a HyperCard add-on called MasterScript that expands the scripting, message watcher, and variable watcher features in HyperCard. MasterScript's script editing window supports selecting text whenever you hold down **Shift** while moving the arrow keys. You can also select a word at a time by combining **Option-Shift** with the arrow keys. This allows you to create a macro that enters the template and then selects the first part of the template that you will change.

NOTE: You will find a number of additional scripting templates in the HyperCard keyset on the companion disk.





Cut to Background

If you have used HyperCard and are familiar with the structure of a stack, you know that each card you look at is a composite of its background and card layers. The background layer contains the objects (buttons, fields, and graphics) that are common to a number of cards, and each card contains objects, text, and graphics that are specific to it. Invariably, I add a button or field to the card layer when I want to add it to the background layer. After moving the buttons and fields that I've

put on the wrong layer a couple of times, I create a sequence that automates this process of moving objects between layers. Let's create a sequence to cut the object, switch to the background layer, paste the object, and then switch back to the card layer.

Before we begin: Select the button, field, or graphic you want to move to the background. You do this to activate the Cut command and to set up the macro. If we started recording before selecting an object, we would have to remove the Click recorded at the beginning of the sequence.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Press -X to cut the item from the card and place it on the Clipboard.
3. Press -B to switch to the background.
4. Press -V to paste the object onto the background.
5. Press -B to switch back to the foreground.
6. Select Stop Recording... from the QuickKeys menu. When the sequence editor opens, your macro should look like Figure 7-19.

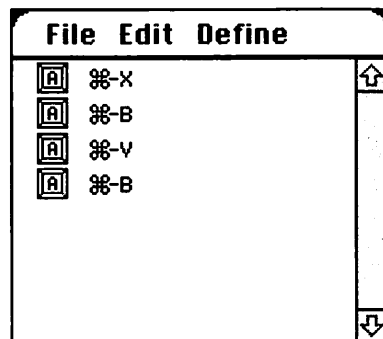




Figure 7-19: The final Cut2Bkgnd sequence.

7. Name your macro and assign a keystroke. For instance, you may want to assign it to --X. Click OK to close the Sequence editor and then click OK to close QuickKeys and save your changes.

Variations: You can create a second macro that reverses the function of this macro. That is, it copies the selected object from the background to the card layer. You can also create a macro that copies the selected graphic from the Scrapbook to the card or background layer or vice versa.

1. Select a graphic or object.
2. Select Record Sequence from the QuickKeys menu.
3. Press $\text{⌘}-C$ to copy the graphic or object.
4. Select Scrapbook from the Apple menu.
5. Select $\text{⌘}-V$ to paste the graphic or object into the Scrapbook.
6. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, your macro should look like Figure 7-20.

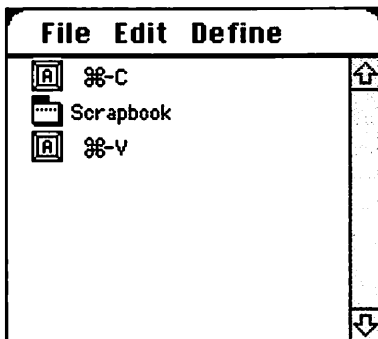


Figure 7-20: The final HC2Scrap macro.

7. Name your macro and assign a keystroke. Click OK to close the Sequence editor and then click OK to close QuickKeys.

If you want to remove the graphic or object and store it in the Scrapbook, you can substitute a $\text{⌘}-X$ for the $\text{⌘}-C$ in this macro.

Changing the User Level

HyperCard has five different user level settings. At user level 1, you are only able to browse stacks; you do not have access to scripting or any of the painting tools. At user level 5, you have access to all of HyperCard's painting tools and scripting capabilities. If you are creating a stack that

is meant to be browsed only and need to change the user level frequently as you test your scripts, you can create a toggle. The macro can open the message box and enter the set user level message in it. You can even add a pause to allow yourself time to enter the user level number so one macro lets you change to any level.

Before we begin: You should have HyperCard open. Go to the Preferences card in the Home stack and set the user level to 5. We are going to create a macro that lets us switch between user level 1 and user level 5.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Press **⌘-M** to display the Message box.
3. Type “set the userlevel to 1” in the Message box (Fig. 7-21). If you have the Blind Typing checkbox selected, you do not have to display the Message box. However, you won’t be able to follow the macro as it runs.

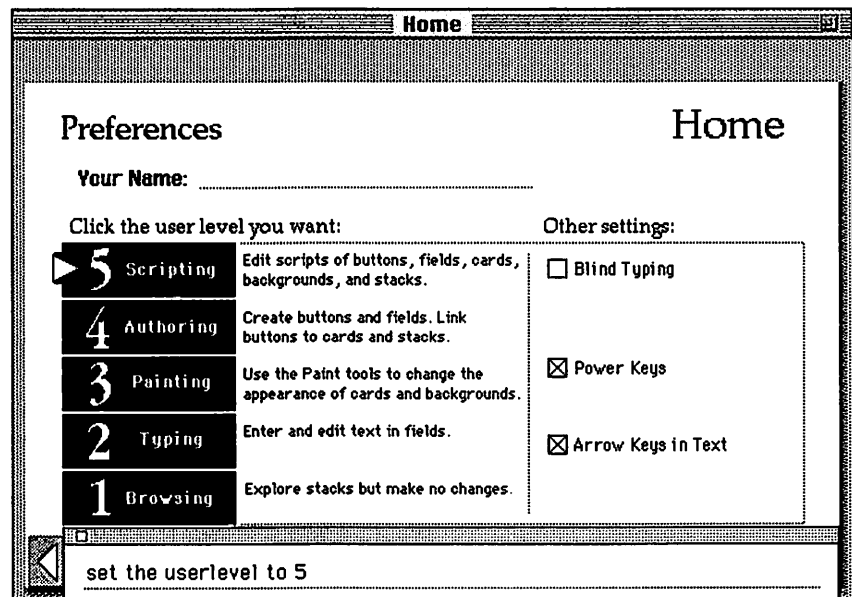


Figure 7-21: The Message box with the set userlevel command.

4. Press **Return** to activate the command.
5. Press **⌘-M** again to close the Message box.

6. Select Stop Recording... from the QuickKeys menu.
7. When the Sequence editor opens, you'll see that your **[Return]** has been added to the text QuickKey. Remove it and the 1. We want to make the macro generic so you can enter any user level and we need to remove the **[Return]** so we can pause the macro just long enough to enter that number.
8. Move the insertion arrow below the text QuickKey and select Pause from the Define menu.
9. When the dialog box opens, enter 2, and click OK to close it and save your changes. Now your macro should look like Figure 7-22.
10. Name your macro and assign a keystroke. Click OK to close the Sequence editor. Then click OK to close QuickKeys and save your changes.

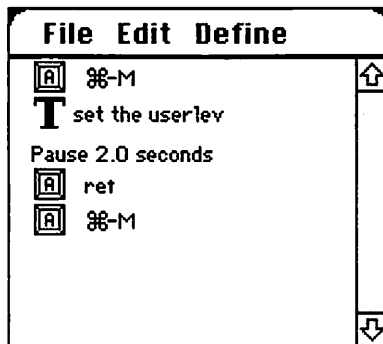


Figure 7-22: The final SetUserLevel macro.

Variations: If you only need to toggle between two user levels, you can create a macro that checks the current user level and toggles it to the other level. You would need to create a ToggleUserLevel macro that checks for the level and copies it to the Message box where it can be checked by the ClipboardAction extension. Depending on the current setting, the ToggleUserLevel macro would play either a macro that sets the user level to 5 or one that sets the user level to 3. Let's see how the macros work together.

1. Select Record Sequence from the QuickKeys menu.
2. Press **[⌘]-M** to open the Message box. We have to display the Message box for this macro because we need to copy something from it.

3. Type "put the userlevel into the Message box."
4. Press **Return** to activate the command.
5. When the user level appears in the Message box, double-click next to the number to select it.
6. Press **⌘-C** to copy the user level to the Clipboard.
7. Press **⌘-M** again to close the Message box.
8. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, we will need to make a few changes to the macro.
9. Open the text QuickKey and remove the **Return**. This isn't necessary, but it certainly makes the macro easier to read.
10. Move the insertion arrow below the text QuickKey, click the Literal button, and then press **Return**.
11. Select Pause from the Define menu. When the dialog box opens, enter 2. This pauses the macro long enough for HyperCard to put the user level into the Message box.
12. Now move the insertion arrow to the end of the sequence, and select ClipboardAction from the Extensions menu. When the ClipboardAction editor opens, select Is from the pop-up menu, and enter 3 in the text box. Make certain the Use System Clipboard and Decision radio buttons are selected (Fig. 7-23).

Enter "SetLevelTo5" in the Pass QuickKey text box. If the user level is 3, this calls a macro that sets it to 5. We'll create it in a moment. Enter "SetLevelTo3" in the Fail QuickKey text box. If the user level is not 3, this plays a macro that sets it to 3.
13. When you are through setting the ClipboardAction extension, click the OK button to close it. Your edited macro should look like Figure 7-24.
14. Click OK to close the Sequence editor.

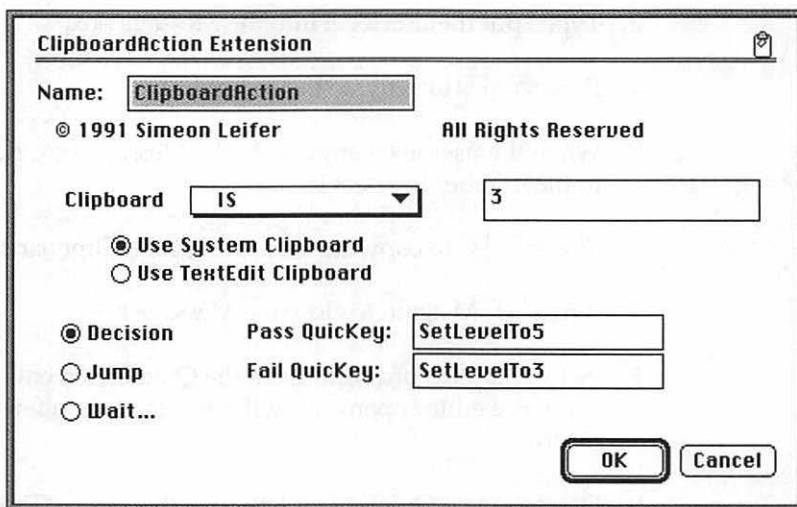


Figure 7-23: The ClipboardAction settings for the ToggleUserLevel macro.

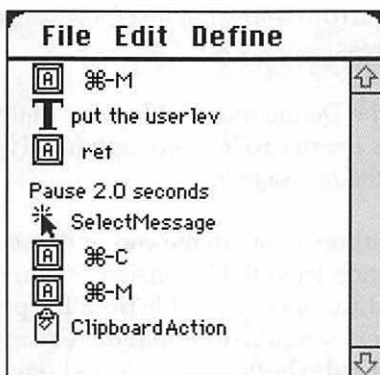


Figure 7-24: The final ToggleUserLevel macro.

Now we need to create the two macros that do the actual work of toggling the user level. We can copy the SetUserLevel macro we created above and make a couple of modifications to it and then rename it.

1. Select the SetUserLevel macro.
2. Select Copy from the Edit menu.
3. Select Paste from the Edit menu.
4. Double-click the copy of the SetUserLevel macro to open it. Change its name to SetLevelTo5 to match the name you entered in the Pass QuickKey text box of the ClipboardAction extension.

5. Select the Pause QuickKey and select Delete from the Edit menu.
6. Double-click the Text editor and add 5 to the end of the command. Your macro should look like Figure 7-25. Click OK to close the Text editor and then click OK to close the Sequence editor.

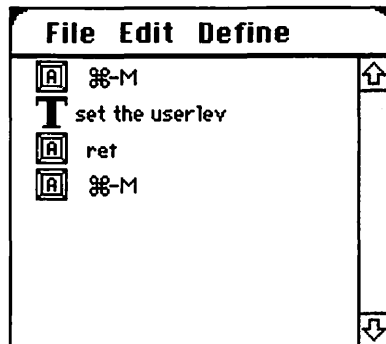



Figure 7-25: The SetLevelTo5 macro.

7. Select the SetLevelTo5 macro you just created.
8. Select Copy from the Edit menu.
9. Select Paste from the Edit menu.
10. Double-click the copy of the SetLevelTo5 macro to open it. Change the name to SetLevelTo3 to match the fail QuickKey you set in the ClipboardAction extension.
11. Double-click the text QuickKey and change the 5 to a 3. Click OK to close the Text editor and then click OK to close the Sequence editor.

Now you have a suite of macros that can check your current user level and toggle it to the other level that you are working in. Though you can't hide the Message box in the ToggleUserLevel macro, you can remove the -M QuickKeys from the SetLevelTo5 and SetLevelTo3 macros if you have the Blind Typing checkbox in the Preferences card selected. This would make the macro run a little faster. It would also be less distracting since the Message box wouldn't be popping open and closed.

Configuring Buttons and Fields

If you are creating a stack with a series of buttons you want to all have the same settings, you can create a macro that automatically changes the settings for the selected button. Not only would this macro save you time, but it would ensure that every button had exactly the same settings.

Before we begin: You should have HyperCard open and the user level set to 5. You also need to select a button for which you want to change the settings.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Select Button Info... from the Objects menu.
3. When the dialog box opens, click the Transparent radio button (Fig. 7-26).
4. Click the Effect... button.

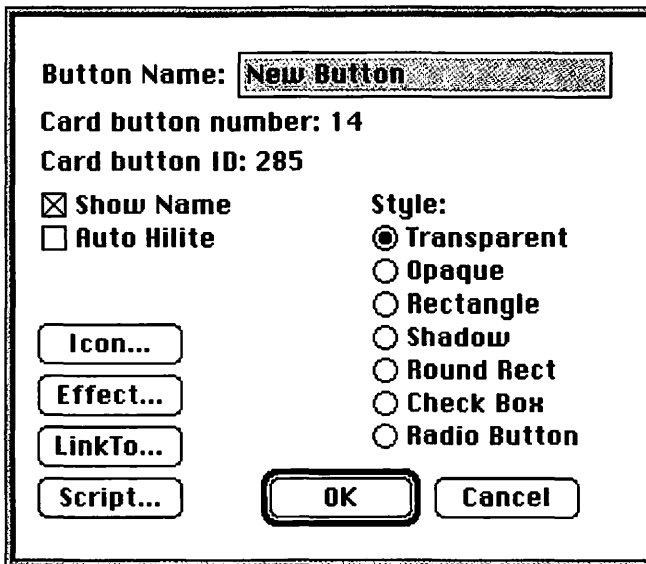


Figure 7-26: HyperCard's Button Info... dialog box.

5. When the Effect dialog box opens, enter the text for a visual effect for the button (Fig. 7-27).

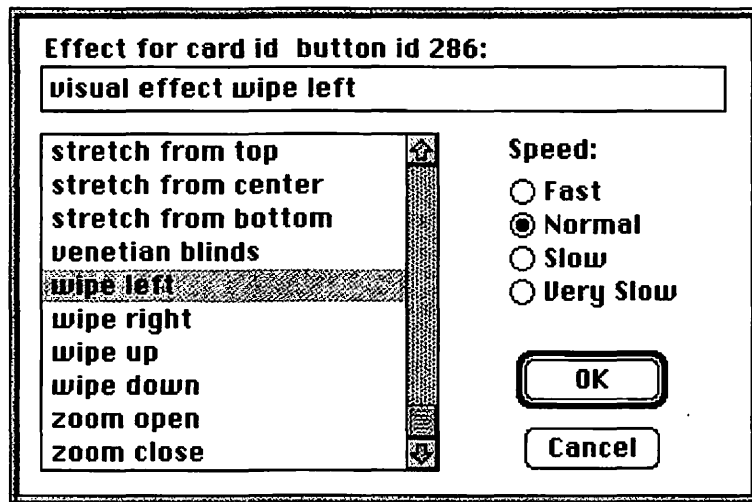


Figure 7-27: The visual effect settings for the SetUpButton macro.

6. Press **Return** to close the Effect dialog box.
7. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, you'll notice that your **Return** is incorporated in the text QuickKey. You can leave it this way, but I prefer to make it a separate QuickKey. Do that now if you like. Your final macro should look something like Figure 7-28.
8. Name your macro and assign a keystroke. Click OK to close the Sequence editor and then click OK to close QuickKeys and save your changes.

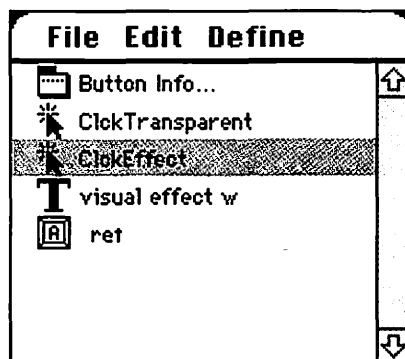


Figure 7-28: The final SetUpButton macro. I've edited the click QuickKey names for readability.

Variations: You could easily expand this macro to set up a series of fields with the same attributes. A SetUpField macro is particularly helpful when you have several font settings to make. To create a modified version of the SetUpButton macro, select a field before you begin recording.

1. Select Record Sequence from the QuickKeys menu.
2. Select Field Info... from the Objects menu.
3. When the Field Info dialog box opens, select the Wide Margins checkbox (Fig. 7-29).

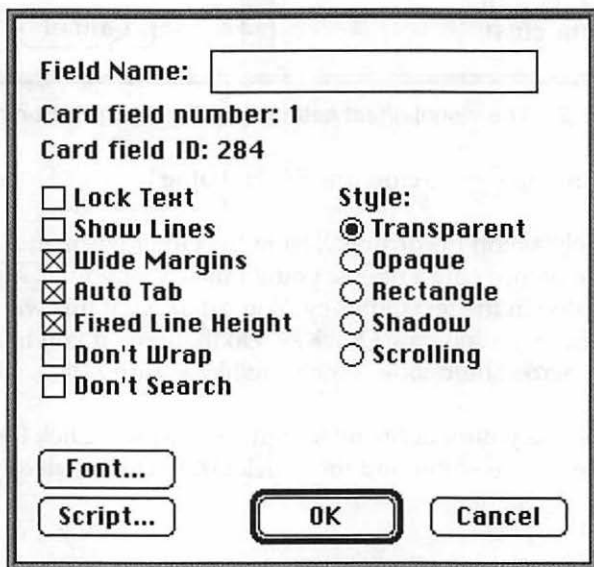


Figure 7-29: HyperCard's Field Info dialog box.

4. Select the Auto Tab checkbox.
5. Select the Fixed Line Height checkbox.
6. Select the Transparent radio button.
7. Click the Font... button.
8. When the Font dialog box opens, select the Bold checkbox (Fig. 7-30).

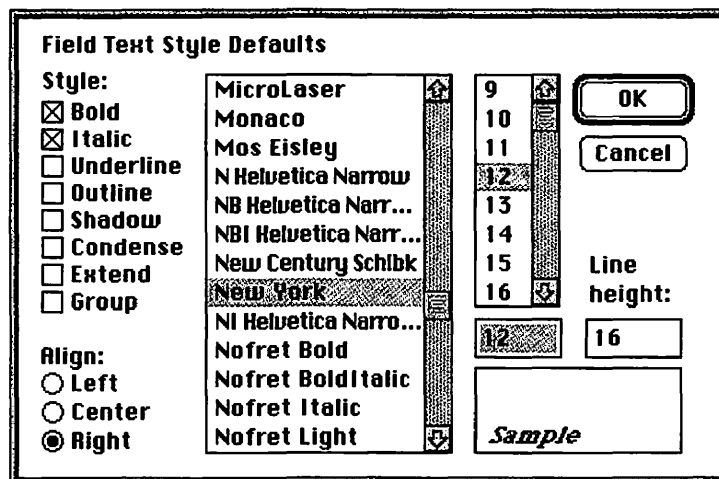


Figure 7-30: HyperCard's Font dialog box.

9. Select the Italic checkbox.
10. Type the name of the font you want to use in the field. The list of font names automatically scrolls and the font is selected. We type the name of the font here because it is more accurate than trying to scroll to the font name and click it.
11. Press **(Return)** to close the dialog box and save your settings.
12. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, your macro should look similar to the one in Figure 7-31.

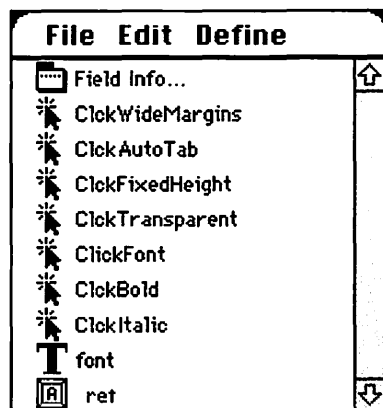


Figure 7-31: The final SetUpField macro. I've edited the click QuickKey names to make them easier to read.

13. Name your macro and assign a keystroke. Click OK to close the Sequence editor. Then click OK to close QuicKeys and save your changes.

Now you have macros that set the attributes for all of your buttons and fields. You can change any of the settings in the macros and easily change the font settings by entering another font name in the text QuicKey.

Now What?

Now that you see how QuicKeys can make your scripting and stack creating jobs easier, let's turn our attention to communication programs. In the next chapter, we'll see how QuicKeys can automate your communication programs—even those without a built-in scripting language.

8

CHAPTER

Macros for Communications

New telecommunications programs such as Microphone II and White Knight now have built-in scripting features to automatically connect to electronic services, gather your mail or download files, and then disconnect. Unfortunately, the scripting languages may be too difficult to learn and debug and can rarely be integrated with other tasks on your Macintosh. You can create QuickKeys macros that allow you to automate communication tasks and integrate those tasks with other activities. In addition to integrating programs with scripting capabilities, you can create macros to automate programs that do not have built-in scripting languages. For instance, the programs that connect to the America Online, CONNECT, and AppleLink electronic services don't have built-in scripting.

Probably the most important reason to create telecommunications macros is that very few other tasks are as time-critical as telecommunications. If you are paying connect charges, fumbling with menu choices or trying to remember the commands you have to type can add minutes to your charges—and minutes add up to hours. Another reason macros are so important for telecommunications is that they allow you to take advantage of the lower rates during non-prime time hours. An example of this is a macro that could launch your communications programs, log onto electronic services, collect your mail, and then shut down your Macintosh while you are asleep. If you have the PowerKey from Sophisticated Circuits, you can turn on your Macintosh early in the morning, run your communications macro, and then shut off your Macintosh.

WARNING: Normally, if a macro fails, you don't have to deal with anything more worrisome than the wrong menu being selected or the wrong button being clicked. When you are deleting files or downloading files from a commercial network, you should take extra care with your macros. It is a good idea to baby-sit your automatic download and e-mail macros the first few times you use them. It is better for the unexpected to happen while you are watching it than at 3:00 a.m. Waking up in the morning to discover that you've been connected to a commercial service all night could definitely put a damper on the rest of your day. You should also pay close attention to changes in commercial services you use. Those changes affect whether or not your macros function the way you've designed them.

General Communications Macros

Some of the most useful macros are those that simplify what you do every day. You may not use communications programs for anything but logging onto the company mainframe and reading your electronic mail, but chances are you have to type the same thing every time you do. Macros can automate that process so you can begin the log-in procedure with a single keystroke. Macros can also add missing features to programs.

Intelligent Shortcuts

In communication programs more than any other program, keyboard shortcuts can save you time and frustration. If you use a terminal-emulation program (i.e., a program that emulates a monitor and keyboard connected directly to a mainframe) to communicate with a mainframe, you can create macros that type the commands you have to enter every day and assign a keystroke to those command strings. Whenever you need to enter a command, you just press a keystroke instead of typing the command. QuicKeys is best suited to this type of customized macro. QuicKeys does most of the work for you, and the macros can become the basis for more complicated macros.

I use a campus mainframe to read electronic mail I receive through the Internet. The Internet is a worldwide system of networks that tie universities and corporations together. After I have read and replied to my mail, I generally delete it and enter a command that cleans up my mail box and deletes all of the old mail files. While these are only a group of text QuicKeys, they save me the trouble of typing—and remembering—

since I rarely remember the command for changing directories on the mainframe.

Before we begin: You should be in your telecommunications program and ready to type the commands you want QuickKeys to enter. For the purposes of this example, I'll be using VersaTerm-PRO (a terminal emulation program from Synergy Systems) and I'll be cleaning up my mailbox on the campus mainframe. Open QuickKeys and select the name of your program from the pop-up menu.

Create the macro:

1. Select Record Sequence from the QuickKeys menu. We could open QuickKeys and create a sequence of text QuickKeys, but it is easier to type the commands as you enter them and clean up the resulting sequence afterward. By doing this, you don't run the risk of leaving out any commands.
2. Type the command for the required task. I'll type "compress." This creates a temporary mail file, compresses the mail file, and renames both the old and the temporary mail files.
3. The next command that I type is "quit" to quit the mail program on the mainframe and return to the command prompt.
4. Now I'm ready to delete the old mail file, but first I have to change directories. I'll type "set default [pfterry.mail]" to change to the mail directory. (For the diehard Macintosh-only users, a directory is equivalent to a folder.)
5. Next I'll type "del mail.old;*" to delete the old mail file. I've entered a wildcard character (the asterisk) to delete any additional old mail files I've missed.
6. I need to change back to the original directory so any files I create aren't saved in the mail directory. I'll type "set default [pfterry]" to change directories.
7. I don't have very much space in my mainframe account and always check the amount of free space I have. I'll type "show quota" to find out how much free space I have. You can see the result of the commands in Figure 8-1.
8. Select Stop Recording... from the QuickKeys menu.

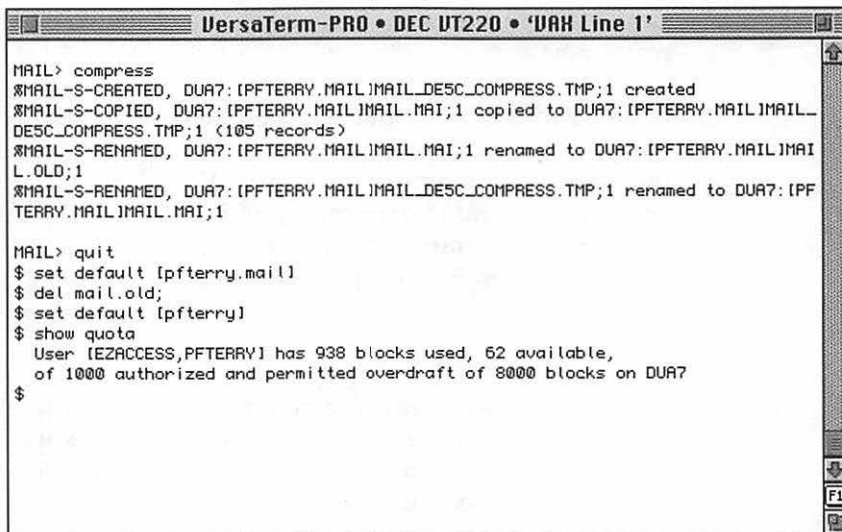


Figure 8-1: The VersaTerm-PRO window showing the commands to compress the mail file and delete the old mail file.

9. When the sequence editor opens, you'll see that all of the commands you typed are a single text QuickKey (Fig. 8-2). You could leave your macro this way, but it would be very hard to later enter pauses or additional steps, so we'll open it and create separate text QuickKeys for each of the commands.

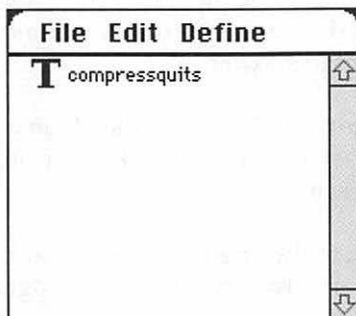



Figure 8-2: The unedited CompressMail sequence.

10. Open the text QuickKey. You'll see all of the commands in the text box. Select all of the commands except for "compress" (Fig. 8-3) and press -X to remove them from the text box and place them on the Clipboard. Click OK to close the Text editor.

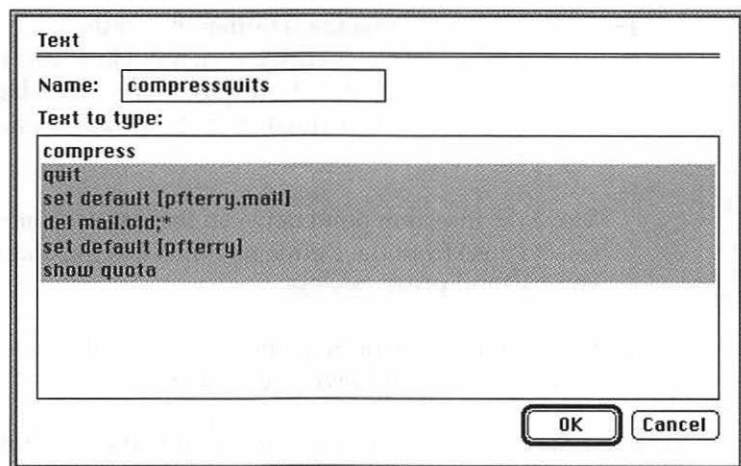


Figure 8-3: The CompressMail text QuickKey before individual text QuickKeys are made for each of the commands.

11. Select Text from the Define menu to create another text QuickKey. When the window opens, make certain that the insertion point is in the text box on the bottom and press $\text{⌘}-V$. This pastes all of the commands, except for `compress`, into the text box.
12. Now select all of the commands except for “quit” and perform the same actions as in steps 10 and 11 above. Each time these steps are performed, this moves all of the commands, except the top one, to the Clipboard. When you are through, you should have a macro with one text QuickKey for each of the commands (Fig. 8-4).

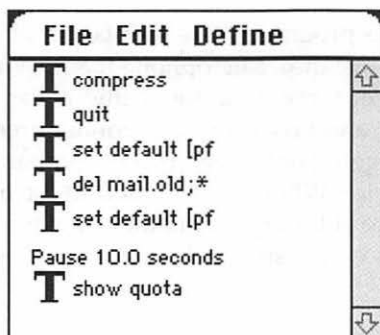


Figure 8-4: The completed CompressMail macro. You'll notice that I've edited the titles of the Text QuickKeys to make them more readable.

13. Each of these commands is buffered, and the mainframe responds as soon as it completes each command. However, I usually get an error message with the "show quota" command because too many commands have queued up. So I'll add a pause to the sequence.

Move the insertion point between the last two commands and select Pause from the Define menu. When the dialog box appears, enter 10 and press .

14. Click OK to close the Sequence editor and then click OK again to close QuicKeys and save your changes.

Use and variations: It is very important that you notice how long it takes a mainframe to respond to your commands and insert pauses at the appropriate places in your macro. Otherwise, the macro will fall out of step with the mainframe and fail. While a failed macro is not dangerous, it is a bother since you have to reenter your commands. After all, you are trying to avoid additional typing.

You should develop a macro for any communications task you do every day. If you can chain several macros together to automate your whole communications session, so much the better. For instance, I can take several intelligent shortcuts that I use whenever I log onto a mainframe and chain those together to completely automate the log-in process. Let's create an automated log-on macro.

Automate Log-Ons

Some telecommunications programs have scripts that let you automate the log-in procedure; for instance, Microphone II and White Knight both have scripting features. For those programs without scripting features (VersaTerm-PRO, Telnet, and FreeTerm, to mention three), you can use QuicKeys to automate logging onto a campus or company mainframe, BBSs, or commercial service. Whenever I connect to the mainframe with VersaTerm-PRO, I have to enter my user name and password. I have created macros that reduce each step of the log-in process to a single keystroke.

Before we begin: You should be in your telecommunications program and ready to type the commands you want QuicKeys to enter. For the purposes of this example, I'll be using VersaTerm-PRO and will be connecting to the campus mainframe. You could use this macro to log into any text-based mainframe or commercial service using any

telecommunications program. Open QuickKeys and select the name of your program from the pop-up menu.

Create the macro:

1. Select Record Sequence from the QuickKeys menu. As above, we could open QuickKeys and create a sequence of text QuickKeys, but we might forget a step. If we record the commands as we type them, we aren't as likely to forget steps. It also gives us a chance to notice how quickly or slowly the mainframe responds to our commands.
2. Type the first command of the log-in sequence. I'll type "c.vx" to connect to the campus mainframe.
3. After a second or two, the log-in banner and a user prompt appear. I'll type my user name "pferry" at the prompt.
4. Then I am prompted for my password. I'll type my password "youguessme."

CAVEAT: *It is not a good idea to enter your password in a text QuickKey. (I'm doing it here since I know—or at least think—my Macintosh is secure.) Anyone who sits down at your computer can open the text QuickKey and get your password. In some cases, this may only be an annoyance, but it can also mean hefty bills from a commercial service. Whenever possible, pause your macro so you can enter the password manually.*

5. Aside from reading *UseNet News*, I only use my VAX account for electronic mail. I'll type "mail" since I'm planning to always make this my first stop.
6. Select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, you'll see that QuickKeys has recorded all of the commands as a single text QuickKey (Fig. 8-5). First, we'll need to make separate text QuickKeys for each command.
7. I'll select the commands from my user name (pferry) through the "mail" command. Press [⌘]-X to cut the commands to the Clipboard. Click OK to close the Text editor.

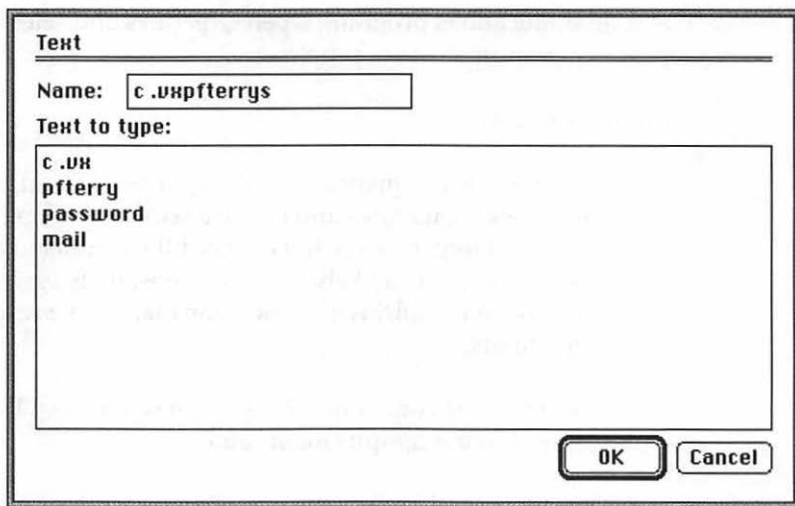


Figure 8-5: The unrefined VAXLogin macro.

8. Select Text... from the Define menu. When the Text editor opens, click in the Text to type box and press $\text{⌘}-V$ to paste the contents of the Clipboard. Repeat these steps, cutting all but the top command to the Clipboard, until you have moved all of the commands into separate text QuickKeys and the macro looks like Figure 8-6.

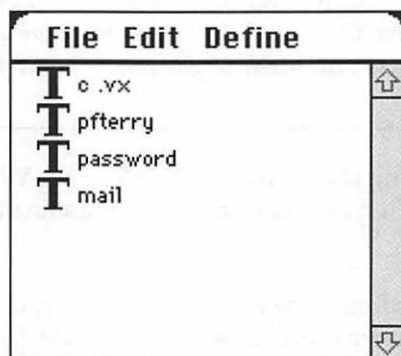


Figure 8-6: The VAXLogin command once all of the commands have been moved to separate text QuickKeys.


9. Click OK to close the Sequence editor and then click OK to close QuickKeys.

At this point, the macro automates everything I have to type to log onto the VAX. However, I want to automate the whole process of dialing the mainframe and checking for a connection, so I'm going to create another macro that plays my VAXLogin macro when my modem successfully connects to the VAX.

1. While you are still in your telecommunications program, open QuickKeys and select Sequence from the Define menu.
2. When the Sequence editor opens, create either a menu/DA... or text QuickKey to connect to the mainframe. In VersaTerm-PRO and Microphone II, you can set up a service and add it to a menu. So I'll create a menu QuickKey that selects VAX from the Sessions menu in VersaTerm-PRO. This dials the VAX.
3. Now I'll need to add a pause that waits until the modem connects before preceding with the macro. Select Pause from the Define menu. When the dialog box appears, enter the length of time it takes for your modem to connect. You will have to time your own modem to get an accurate number, and you should always allow some extra time. After all, this is a macro and you don't have to sit and wait on it.
4. Click the Record More button in the Sequence editor. We need a click QuickKey to copy the contents of the window. When QuickKeys closes and the flashing microphone icon tells you that QuickKeys is recording, click at the top of your communications program's window and drag down. This selects the contents of the window. Select Stop Recording from the QuickKeys menu or press the keyboard shortcut you have assigned to the Recorder On/Off special. The QuickKeys window opens again with the Click editor in front. You may want to name the click QuickKey to make your macro easier to read. Click OK to close the Sequence editor.

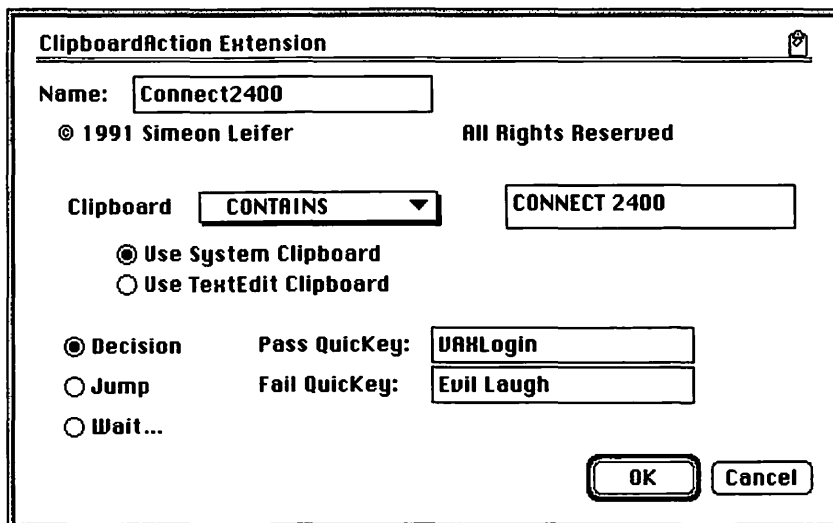
NOTE: *If your communications program has a Select All command, use it instead of this click QuickKey.*

5. Since QuickKeys can't currently monitor the modem port of your Macintosh and play a QuickKey when the computer you are calling responds, we have to test the screen contents for the connection. Once we have selected the contents of the window, we can copy it to the Clipboard and test for a connection.

Select Alias from the Define menu. When the editor opens, press -C. This will copy the window contents to the Clipboard.

6. Select ClipboardAction from the Extensions menu. When the editor opens, select Contains from the Clipboard pop-up menu. Enter "Connect 2400" in the text box. Then click the Decision button (Fig. 8-7). Enter VAXLogin in the Pass QuickKey text box. This is the name of the macro we created above and the one I want played when the modem makes the connection to the mainframe. Enter the name of a QuickKey for Clipboard Action to play if the window contents don't contain "connect 2400." For example, I have a sound QuickKey called Evil Laugh that is played whenever the macro fails so I'll know to play the macro again. You can use the Sound extension to play any sound in your system or the Message extension to display a message on your screen telling you to play the macro again.

NOTE: *VersaTerm-PRO prints "connect 2400" in the window whenever it connects to the mainframe. Your communications program may display something different depending on its configuration, the speed of your modem, and the speed of the modem you are calling.*



The image shows a dialog box titled "ClipboardAction Extension". It has a "Name:" field with "Connect2400" entered. Below this is a copyright notice "© 1991 Simeon Leifer" and "All Rights Reserved". There is a "Clipboard" section with a dropdown menu set to "CONTAINS" and a text box containing "CONNECT 2400". Below this are two radio buttons: "Use System Clipboard" (selected) and "Use TextEdit Clipboard". There is a "Decision" section with three radio buttons: "Decision" (selected), "Jump", and "Wait...". To the right of these are two text boxes: "Pass QuickKey:" with "VAXLogin" and "Fail QuickKey:" with "Evil Laugh". At the bottom right are "OK" and "Cancel" buttons.

ClipboardAction Extension	
Name:	Connect2400
© 1991 Simeon Leifer All Rights Reserved	
Clipboard	CONTAINS
CONNECT 2400	
<input checked="" type="radio"/> Use System Clipboard <input type="radio"/> Use TextEdit Clipboard	
<input checked="" type="radio"/> Decision	Pass QuickKey: VAXLogin
<input type="radio"/> Jump	Fail QuickKey: Evil Laugh
<input type="radio"/> Wait...	
OK Cancel	

Figure 8-7: The Clipboard Action configuration for the AutoConnect macro.

7. Name the macro and assign a keystroke. Your final macro should look similar to the one in Figure 8-8. Click OK to close the Sequence editor.

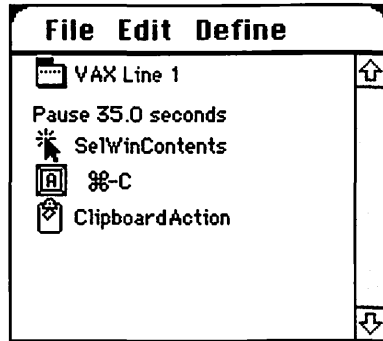


Figure 8-8: The complete AutoConnect macro.

8. Click OK again to close QuicKeys and save your changes.

Use and variations: You can modify this macro to work with any communications program that doesn't have a built-in scripting feature and with just about any university or company mainframe or electronic service. As I mentioned above, you should always remember to time connections and give your macro plenty of time to run.



Adding Missing Features to Programs

As we've seen in previous chapters, it is very easy to add features to programs using QuicKeys. In this book, I may already have dealt with some topics you may consider very arcane. Let me warn you now: This macro pushes that envelope. The macro itself is simple. Just as we created alias QuicKeys in Chapter 2 to mimic Word 4 keyboard shortcuts in other programs, I'll be creating text QuicKeys to mimic a VT-102 terminal emulator. The top four keys on the keypads on these terminals are labeled "PF1," "PF2," "PF3," and "PF4." The PF1 key is also known as the Gold key; pressing the Gold key before pressing another key on the keypad modifies the function of that second key. You may find the circumstances under which I use these macros a bit esoteric. But then, you may not, and they'll be of invaluable use to you.

I normally use VersaTerm-PRO for reading NEWS (a newsfeed that rides on the worldwide Internet network). I like VersaTerm-PRO because it configures your Macintosh keypad to function like the keypad of a VT-102 terminal. When you are in NEWS, you can press a

single keystroke or two to reply or scroll through the messages. However, VersaTerm-PRO 3.5 only supports a single FTP connection at a time (FTP is the file transfer protocol computers use over the Internet to connect to other computers), and I have to stop reading NEWS if I want to connect to a remote connection over the Internet and look for a file. The public domain program Telnet supports multiple FTP connections; unfortunately, however, it doesn't support the VT-102 keypad the way Versa-Term Pro does. QuickKeys allows you to create macros for these commands and assign keystrokes similar to those on a VT-102 terminal.

Before we begin: You should be in the telecommunications program you'll be using. I'll be using Microphone II version 4.0 for this macro (since it also doesn't support the VT-102 commands exactly like VersaTerm-PRO). Open QuickKeys and select the program from the pop-up menu.

The actual keypad layout for NEWS takes advantage of the Gold modifier key on the VT-102 keyboard. Where you would press  and one of the keypad keys on your Macintosh, on the VT-102 you would press the Gold key (the Num Lock key on your Macintosh's keypad) then the  key (see Figure 8-9). While we cannot mimic this keyboard layout exactly with QuickKeys, we can get close.

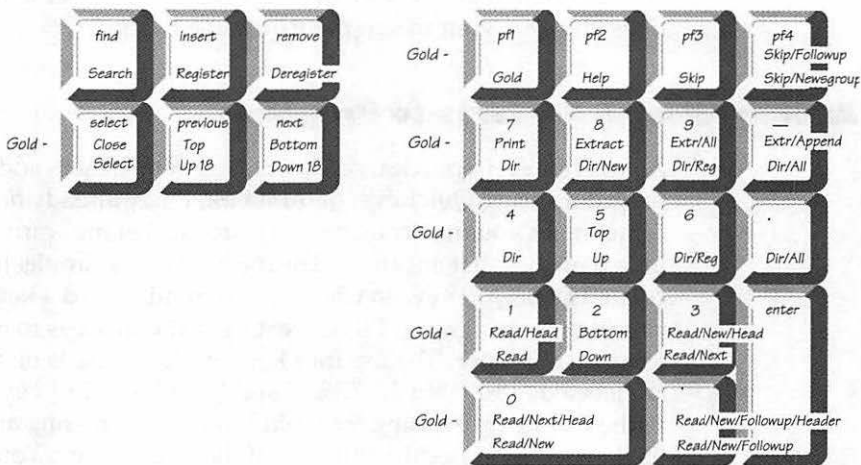


Figure 8-9: The NEWS configuration for the VT-102 keypad.

Create the macro:

1. Open QuickKeys if you have closed it, and select Text... from the Define menu.

2. Let's make a couple of text QuickKeys for the Page Down key on the extended keyboard. When the Text editor opens, type "down 18" and then press **Return**. Inserting the **Return** in the text QuickKey saves you an additional keystroke. Down 18 scrolls the NEWS screen down one page.
3. Click in the Keystroke box.
4. Press the **PgDn** key. This keystroke exactly matches the NEWS assignment on a VT-102 terminal. There is a better than average chance that you already have the **PgDn** key assigned to scrolling a page down in the Universal keyset; QuickKeys will warn you if that is the case (Fig. 8-10).

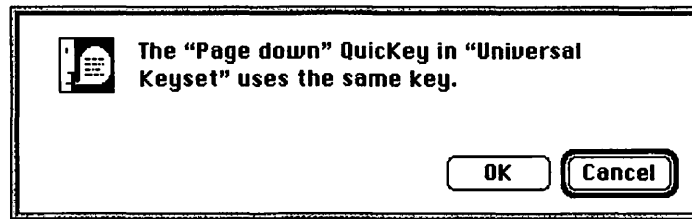


Figure 8-10: QuickKeys warns you when you are about to assign a keystroke that has already been assigned to another macro.

The keystrokes assigned in a program's keyset always override those in the Universal keyset. You can assign the **PgDn** key to your "down 18" macro in Microphone II, and it will play instead of the Universal keyset macro when you are in Microphone II.

NOTE: Double-assigning keystrokes in this way lets you double the functionality of your keyboard. The only danger is that you may want to use the universal macro when you are in a program in which you have assigned a program-specific macro to the same keystroke.

5. Your macro should look like the one in Figure 8-11. Click OK or press **Return** to close the QuickKey and save your changes.
6. Now let's create the second macro for the **PgDn** key. Select Text... from the Define menu.
7. When the Text editor opens, enter "bottom" in the Text to type box and press **Return**.

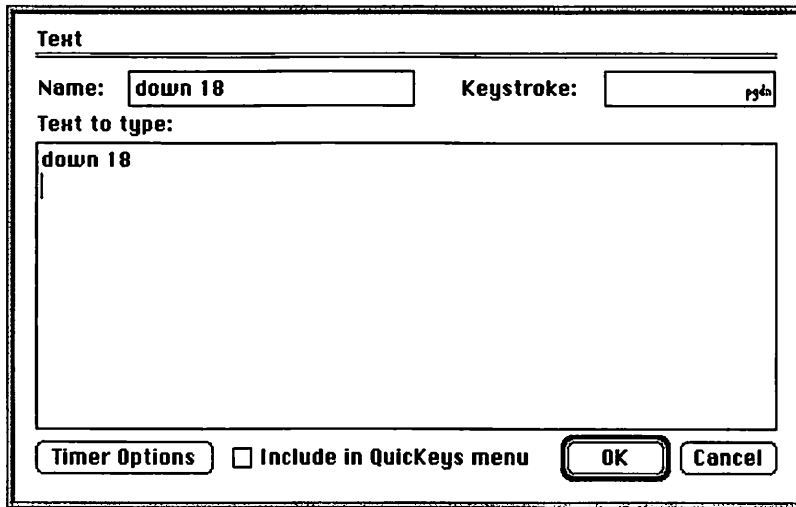


Figure 8-11: The “down 18” macro for NEWS.

8. Click the Keystroke box and press **Option-PgDn**. If you were using a VT-102 keyboard to read NEWS, you would press the Gold key (the **Num Lock** key on an extended keyboard) and then press the **PgDn** key. Since we can't assign that exact keystroke in QuickKeys, **Option-PgDn** will work.

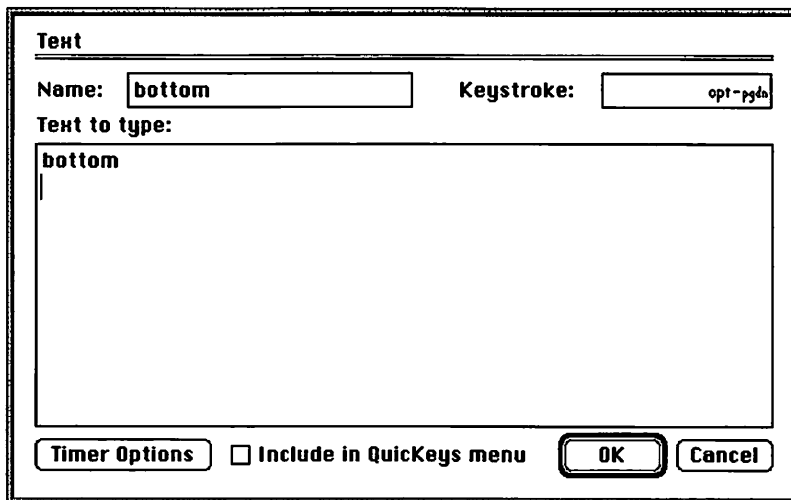


Figure 8-12: The “bottom” macro for NEWS.

Use and variations: Before you accuse me of being a bit obsessed with macros, type “close” (I won’t even mention “read/new/followup/

header”) and then press the **(Option)-(Del)** keys (the Microphone II NEWS keystroke assigned to “close”). Which is easier? While “close” is only a six-character command (counting the **(Return)**), you can quickly tire of typing it. Though I am not going to walk through the creation of every NEWS shortcut (you get the idea), you’ll find complete keysets for both Microphone II and Telnet on the companion disk.

Automating E-Mail and Remote File Sharing

You can team QuicKeys with Remote Access to create a very powerful communications duo. Although you could use QuickMail Remote or Microsoft Mail’s Dial-In control panel, they force you to dedicate a line solely to dial-in mail service. Remote Access gives you access to all of your network resources—printers, file servers, and mail servers. For instance, you could connect to the office network from home and get your mail. You could also automate file transfers to the office by combining DiskTop with AppleTalk Remote Access (a communications program that allows you to use two modems to connect remote networks) and QuicKeys. If you are running System 7, you could use the Finder’s Alias function to create an alias of your office server or your office Macintosh. Then you could use QuicKeys to send a Finder Event to open the file server alias. Once you have mounted the server, you can use DiskTop to automate copying files. If you are running System 6, you could use Farallon’s Liaison instead of AppleTalk Remote Access and you could use the Mounity extension in your macro to mount the office file server.

Getting Your Mail from the Office

Whenever I get home at the end of the day, I always remember that I was going to read my electronic mail before leaving the office. If you can dial into your office network and you are running AppleTalk Remote Access at home and at the office, you can create a macro that automatically calls your office Macintosh and gets your most recent messages. This macro allows you to call and check your mail before you leave for work in the morning. It also makes it possible for you to check your mail when you are on the road and take advantage of lower rates early in the morning.

Before we begin: You must have Remote Access running and configured to receive calls on your office Macintosh or server. You should also close your mailbox and sign out before you leave the office. You will need to have System 7 running on your home computer to use Remote Access.

You should also have configured Remote Access to call your office Macintosh and save the settings. Finally, you should have Microsoft Mail or Quick Mail installed on your home Macintosh or your Macintosh Portable or PowerBook. Open QuickKeys and select Universal from the pop-up menu.

Create the macro:

1. Launch Remote Access and open the settings file for calling your office.
2. Select Record Sequence from the QuickKeys menu.
3. Click the Connect button in the Remote Access window (Fig. 8-13) and then select Stop Recording... from the QuickKeys menu. At this point, Remote Access initiates the call in the background, and the Click editor opens. Click OK to close Click editor.



Figure 8-13: The Remote Access window.

4. Move the insertion arrow above the Click and select Finder Event from the Extensions menu. When the editor opens, select Open from the pop-up menu. When the Select Files dialog opens, select your Remote Access document. Click OK to close the editor and save your changes.
5. Move the insertion arrow below the click QuickKey and select Pause from the Define menu. When the Pause window opens,

enter the amount of time for the macro to wait while your Macintosh places the phone call and connects to the office Mac. As you will discover with other communications macros, setting the amount of time for a pause is tricky. If you don't give the macro enough time for your Macintosh to make the connection, you have guaranteed its failure. If the macro runs before you get out of bed in the morning and you aren't paying long distance charges, give it plenty of time so the menu/DA QuickKey works.

6. Select Menu/DA... from the Define menu. When QuickKeys prompts you, select Microsoft Mail from the Apple menu.
7. Select WindowAction from the Define menu. WindowAction lets you pause the macro until the Microsoft Mail 3.0 window opens. When the WindowAction editor opens, select Is from the Window Name pop-up menu, enter "Microsoft Mail 3.0" in the text box, and click the Wait... radio button (Fig. 8-14).

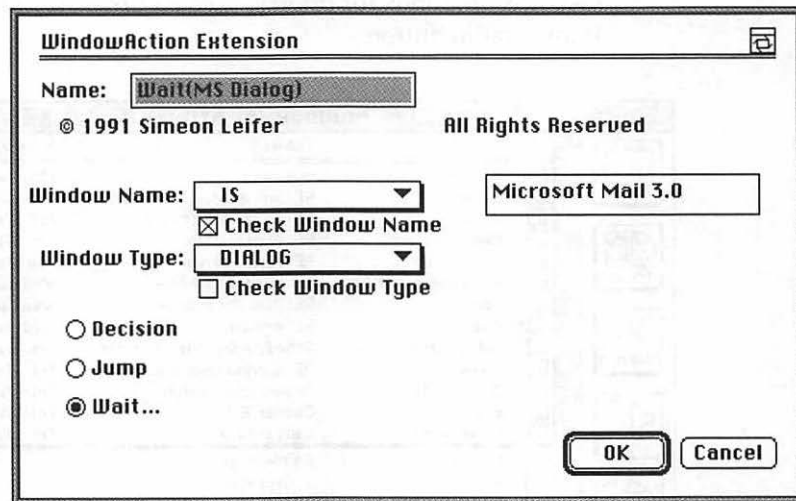


Figure 8-14: The WindowAction settings for the Microsoft Mail 3.0 window.

8. If you have signed out of Microsoft Mail before leaving the office, the dialog box opens with the cursor in place for you to enter your password (Fig. 8-15). Select Text... from the Define menu. When the Text editor opens, enter your password.
9. Click the Literal button. When QuickKeys prompts you to press any key, press **Return**. This alias clicks the OK button in the Microsoft Mail 3.0 window.

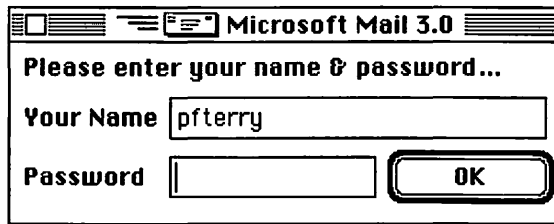


Figure 8-15: The log-in window for Microsoft Mail.

NOTE: We could enter **Return** in the text window after the password (as we did with the NEWS text QuickKeys above), but we wouldn't be able to enter a pause later if we needed to slow the macro down.

10. Now we want the macro to wait until the mailbox opens. Select WindowAction from the Define menu. When the editor opens, select Is from the pop-up menu, enter your mailbox name (for instance, "Mailbox for pfterry"; see Figure 8-16), and click the Wait... radio button.

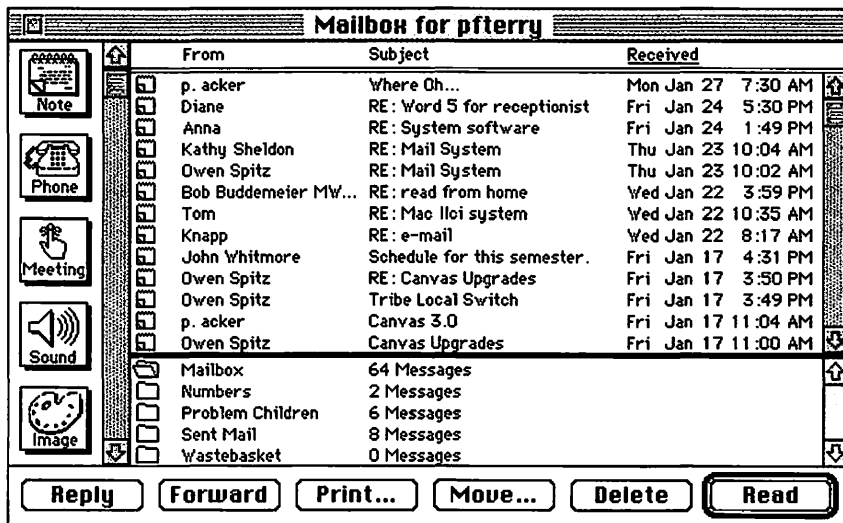


Figure 8-16: The Microsoft Mail mailbox window.

11. To read the most recent messages, you'll need to set up a repeat. Select Repeat... from Extensions menu. When the Repeat editor opens, enter the number of the messages you want the macro to open. For instance, you might want to enter five. Then click the OK button to close the editor and save your changes.

NOTE: Do not click the Put up dialog checkbox because that will stop your macro until you respond. This portion of the macro opens the five most recent messages in your mailbox.

12. Select Alias from the Define menu. When the editor opens, press the key. This alias selects the next message in the list.
13. Click the Literal button. When QuickKeys prompts you, press . Where the previous QuickKey selected the message, this alias clicks the Read button and opens the message.
14. Select Second Window from the Specials menu. This brings your mailbox window to the front so the can move to the next message and read it when the macro repeats.
15. Select Repeat... from the Extension menu again. When the Repeat editor opens, click the End Repeat button and the OK button to close the editor and save your changes.
16. Name your macro and assign a keystroke. Your completed macro should look like Figure 8-17.

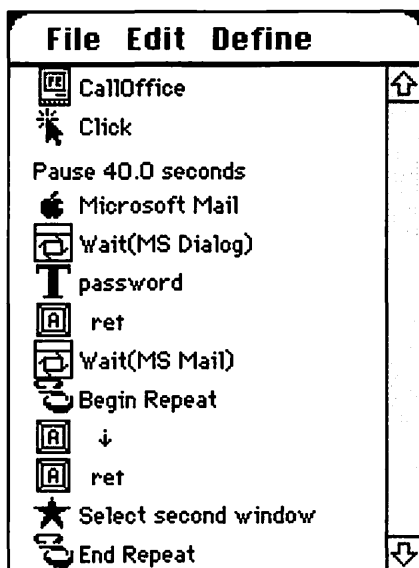


Figure 8-17: The completed GetMSMail macro.

17. Click OK to close the Sequence editor and then click OK again to close QuickKeys.

Variations: If you would prefer to avoid the repeat loop, you could modify this macro so it gets only the urgent messages from Microsoft Mail. To do this, you should select the Open Urgent Messages in your Microsoft Mail Preferences dialog box. Then when the macro opens your mailbox, the messages are automatically opened.

You could also modify this macro to connect to your office and get your QuickMail messages. The macro is primarily the same, but you can remove the repeat because QuickMail supports the Select All command. With Select All, you can open your mailbox and select all of the messages and read them. If you keep your incoming folder empty (you can tell that I don't), only the new messages are opened.

1. The beginning of the macro is the same. Instead of selecting Microsoft Mail from the menu, however, you would select QuickMail.
2. The WindowAction extension would wait for the QuickMail connect dialog to open. You would select Dialog from the Window Type pop-up menu and click the Wait... radio button.



Figure 8-18: The QuickMail connect dialog.

3. The text **QuicKey** remains the same. It enters the password when the connect dialog opens.
4. The **Return** alias clicks the Connect button.
5. The next **WindowAction** waits for the QuickMail mailbox to open. You would select **Is** from the Window Name pop-up menu and enter "QuickMail™" in the text box (press **Option**-2 to get the "TM" symbol) and then click the **Wait...** radio button. Now you are ready to change the remainder of the macro. Click **OK** to close the editor.
6. Click the **Literal** button. When **QuicKeys** prompts you to press any key, press **⌘**-A. This alias selects all of the messages in the incoming folder.
7. Select **Alias...** from the **Define** menu. When the editor opens, press **⌘**-R. This alias clicks the Read button and opens all of the selected messages.
8. Name the macro and assign a keystroke. The completed **GetQuickMail** macro should look like Figure 8-19.
9. Click **OK** to close the Sequence editor and then click **OK** to close **QuicKeys**.

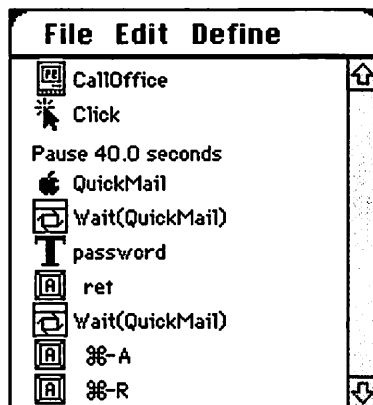


Figure 8-19: The GetQuickMail macro.

Now you have modified the macro so it will work with QuickMail. It is much more accurate than the Microsoft Mail macro and will get all of your new mail (provided you keep your mailbox tidy).

Backing Up Files Remotely

Backing up files is one of the most annoying necessities of computing. It takes time and a certain amount of compulsion to do it on a routine basis. Once you've experienced a hard disk head crash or you accidentally overwrite the most recent version of a file with an older version, you begin to really understand the value of backups. Off-site backups are even more important since they provide an additional level of security: If a fire guts your office, you still have backups of your work safely stored in another building. For instance, you could back up the files you are working on at home on the office server and backup files from the office on your home Macintosh. This macro performs a remote backup using QuickKeys, Remote Access, and a file management program such as DiskTop or DiskTools.

Before we begin: You will need to have a phone line and modem at your office with the modem connected to your Macintosh or some Macintosh on the network. Both your home and office Macintoshes must be running System 7 and AppleTalk Remote Access. If you are running System 6, you could use Liaison instead of Remote Access.

Call the office from home, open the Chooser, and select and mount the server or your office Macintosh's hard disk. When your office server or Macintosh appears on the desktop, select it and then select Make Alias from the Finder's File menu. (If you aren't familiar with making aliases, refer to your System 7 documentation.) If you keep the alias on your desktop or in the folder with Remote Access, you'll be able to eliminate a couple of steps from the macro by using a Finder Event to open the alias. This is the same as double-clicking on the alias.

Create the macro:

1. Open QuickKeys and select Universal from the pop-up menu. Then select Sequence from the Define menu.
2. When the Sequence editor opens, select FinderEvents from the Extensions menu.
3. Select Open from the pop-up menu. When the dialog box opens, select the server alias you created (Fig. 8-20). Click the Add button and then click Done to close the dialog box. Click OK to close the Finder Events editor.

When QuickKeys invokes the Finder Event, Remote Access displays a dialog box with the progress of the connection (Fig. 8-21).

This dialog box appears in front of any window that you have open.

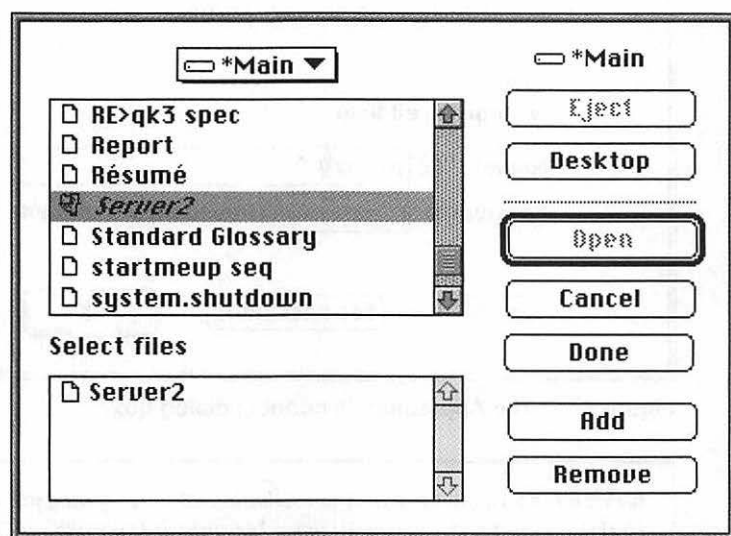


Figure 8-20: The Finder Event Select Files dialog box.

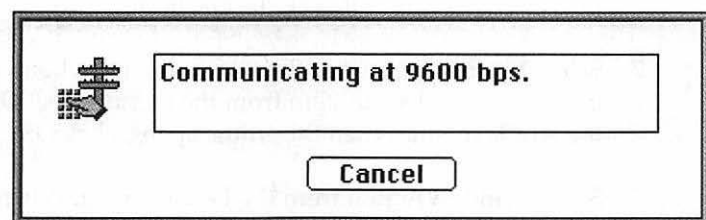


Figure 8-21: When you use a system alias to dial your server or office Macintosh, Remote Access displays a dialog box with the progress of the call.

4. Select WindowAction from the Extensions menu. When the editor opens, select Dialog from the Window Type pop-up menu and click the Wait... radio button. This causes the macro to wait until the AppleShare connect dialog box appears (Fig. 8-22). Click the Check Name checkbox to deselect it. Click OK to close the editor.
5. Select Text... from the Define menu. When the editor opens, enter your password in the text box. When the dialog box above opens, QuickKeys enters your password. Click OK to close the Text editor.

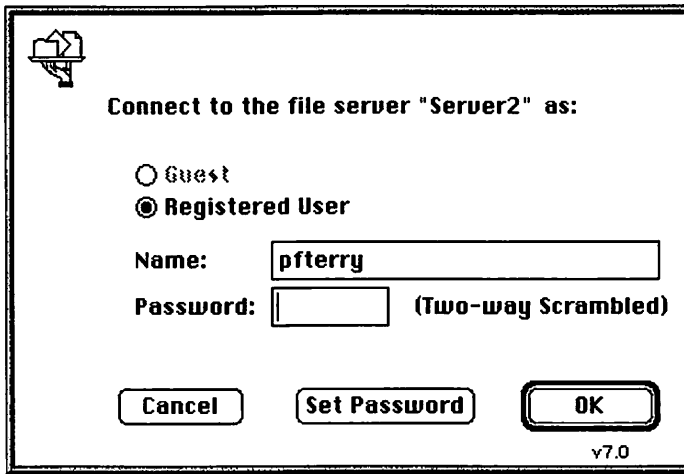


Figure 8-22: The AppleShare's connect dialog box.

CAVEAT: Remember that you should not enter your password in a text QuicKey unless you are very your Macintosh is secure.

6. Click the Literal button. When you are prompted to press any key, press **Return**. This will click the OK button.
7. Select Menu/DA... from the Define menu. When you are prompted to select an item from the menu, select DiskTop from the Apple menu. When the editor opens, click OK.
8. Select WindowAction from the Define menu. When the WindowAction editor opens, select Is from the Window Name pop-up menu, select the Check window Name checkbox, and enter DiskTop in the Window Name text box. Then click the Wait... radio button (Fig. 8-23).
9. Name the WindowAction QuicKey to make it easier to read the macro and close the editor to save your changes. This causes the macro to pause until the DiskTop window comes to the front. While it isn't necessary, this extension can avoid some annoying problems. For example, if you are in a document, **⌘-A** could select all of the text in your document before the DiskTop window has a chance to open.
10. Click the Literal button. When you are prompted, press **⌘-T**. This is DiskTop's keystroke to toggle from the files view to the

DiskTop view. Because DiskTop defaults to the current folder, we need to change the view before we change the folder with the Location extension.

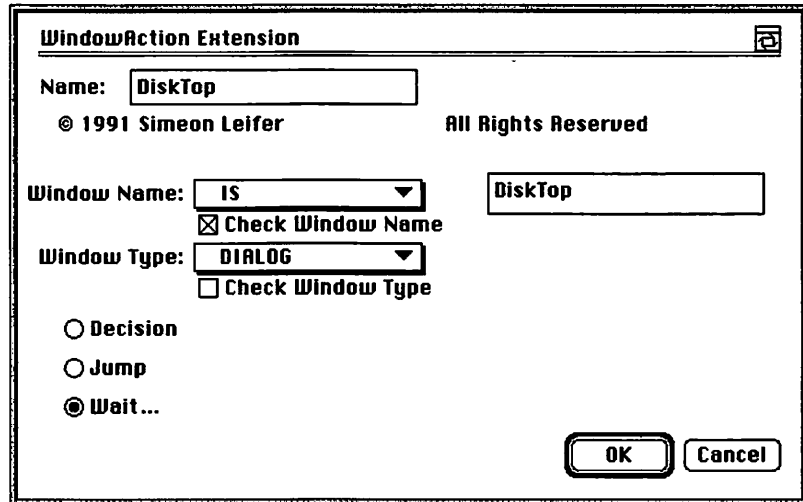


Figure 8-23: Specifying a window name for the WindowAction extension to wait.

11. Select Location from the Extensions menu. When the Location editor opens, click the Select Location button (Fig. 8-24).

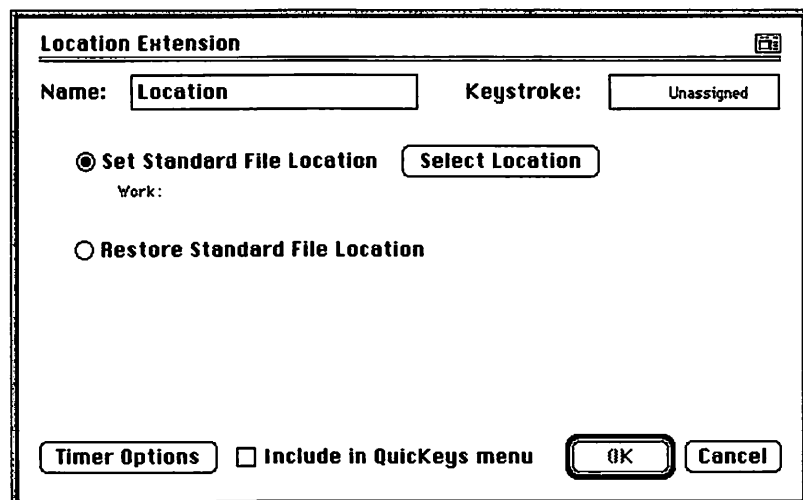


Figure 8-24: The Location editor. Notice the beginnings of a path below the Set Standard File Location button.

A Select location dialog box opens so you can choose the folder that contains the files you want to back up and click the Select button (Fig. 8-25). Be sure to enter the folder (the Open button should be dimmed). When the dialog box closes, you should see the path to the folder you have just selected below the Set Standard File Location button.

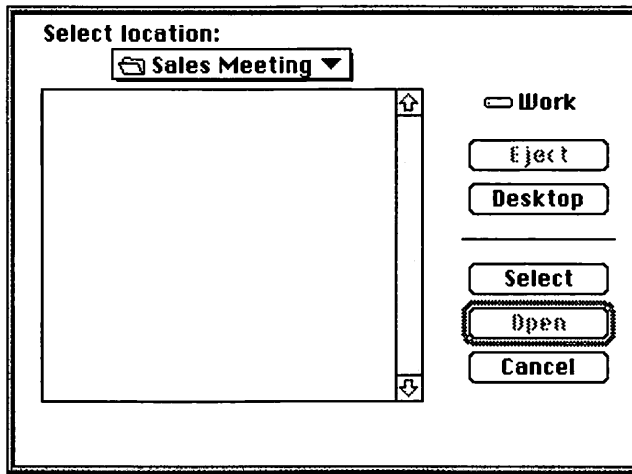


Figure 8-25: Using the Location editor to select the folder that contains the files that you want to back up.

12. Click OK to close the Location editor and save your changes. When the macro runs, this sets DiskTop to open in your project's folder.
13. Click the Literal button and press **Enter** when you are prompted for a keystroke. This is DiskTop's keystroke to return to the files view in the DiskTop window. Pressing this keystroke after you have set the location displays your project folder.
14. Click the Literal button and press **⌘-A**. Now that we have DiskTop opening to the folder containing the files and folders that you want to back up, **⌘-A** selects the contents of the folder in DiskTop's window (Fig. 8-26).
15. Click the Literal button and then press **⌘-C**. This clicks the Copy button in DiskTop's window.
16. Select Location from the Extensions menu again. When the editor opens, click the Select Location button and choose the destination

folder for the backup files. Close the editor. At this point, when you run the macro, DiskTop opens a dialog box for you to select a destination for the files/folders you are copying. This location QuickKey selects the backup folder (Fig. 8-27).

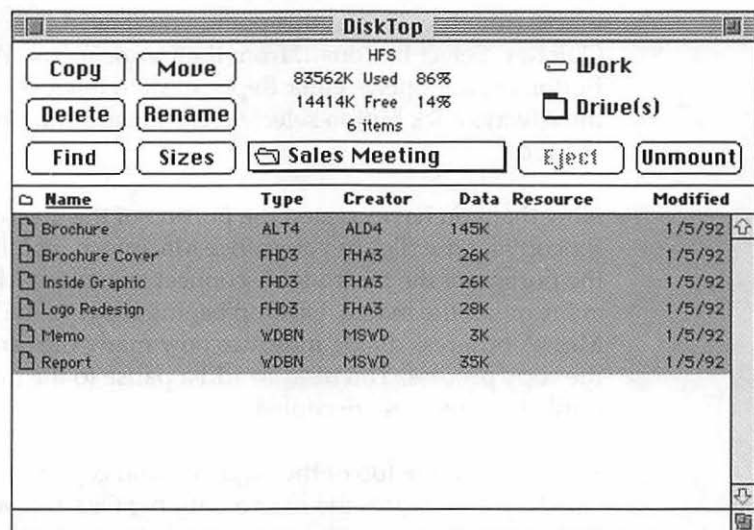


Figure 8-26: The DiskTop window open to your project folder.

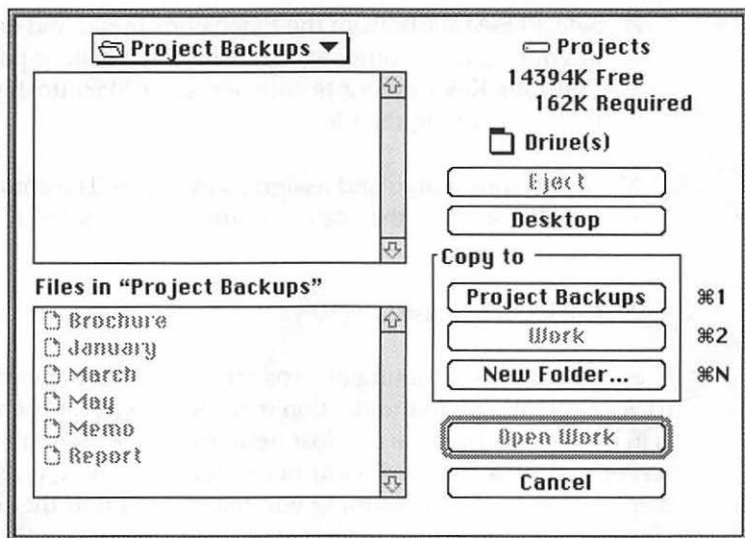



Figure 8-27: DiskTop's Copy To dialog box.

17. Click the Literal button and press -1. This clicks the Project Backups button in DiskTop's Copy To dialog.
18. If you are replacing files on your server, DiskTop asks if you want to replace the files with the ones you are copying. To keep the macro from breaking at this point, you'll need to add a button QuickKey. Select Buttons... from the Define menu. When the Buttons editor opens, enter Replace All in the text box and leave the Always click button selected. Close the editor to save your changes.
19. Now that you have created the portion of the macro responsible for copying the files to your office Macintosh, you'll need to create the portion of the macro to disconnect from your Mac. When you are running the macro, Disktop begins copying the files to the Mac. A progress dialog indicates how many files are remaining in the copy process. You need to add a pause to the macro to wait until all of the files are copied.

Scroll back to the top of the sequence and copy the WindowAction QuickKey that pauses the macro until the DiskTop window comes to the front. Then scroll down to the bottom of the sequence and paste the QuickKey. This pauses the macro until the DiskTop progress dialog goes away and the DiskTop window comes to the front.
20. Select DisMounty from the Extensions menu and enter the name of your server or office Macintosh's hard disk or partition name. This QuickKey unmounts your server or Macintosh when DiskTop finishes copying the files.
21. Name your macro and assign a keystroke. The completed macro should look like the one in Figure 8-28. Click OK to close the Sequence editor.
22. Click OK to close QuickKeys.

Use and variations: Creating macros for AppleTalk networks is just as tricky as creating communication macros. Whenever the zone the server is in disappears because of a lost network connection or the zone's or server's name is changed, your macro will fail, and QuickKeys will display a dialog box informing you that it can't find the server in the specified zone. You also have to pay close attention to the speed of your network. If you insert pauses for specific periods of time, make certain that you have allowed enough time for the server to respond on even the slowest day.

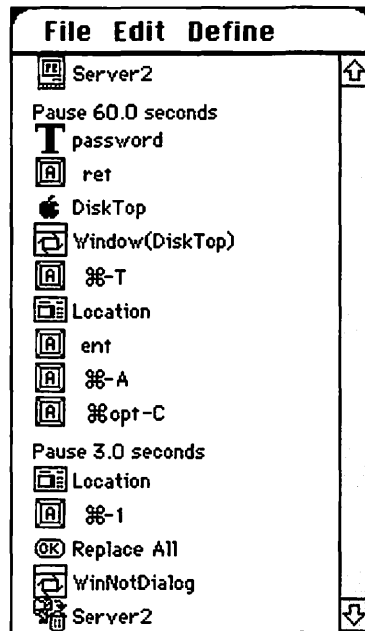


Figure 8-28: The final RemoteBackUp macro.


Now that we have called the office and transferred the files, we need to hang up the phone. While connecting to the office Macintosh by opening the server alias takes fewer steps and is more reliable than launching Remote Access and clicking the connect button, this method poses a problem when it is time to disconnect from the office. Remote Access made the connection, but it isn't an open application that you can quit. To disconnect, you'll have to launch Remote Access, click the Disconnect button, and then quit. Since this is a macro that has the potential to be used in a number of different circumstances, let's develop it separately from the RemoteBackUp macro above.

Disconnecting Remote Access

When you are through transferring files or gathering your e-mail, you will need to disconnect the call. You could send Remote Access a Quit Application Apple Event, but that won't disconnect the call. You have to switch to the Remote Access program and click the Disconnect button. If you've dialed a service by opening or double-clicking on its alias, you'll have to launch Remote Access and click the disconnect button. We'll create a simple macro that you can reuse in a number of different situations.

Before we begin: As before, you'll need to have System 7 and Remote Access running. Open QuicKeys and select Universal from the pop-up menu. We'll put the macro in the Universal keyset so it will be available anywhere.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Select Remote Access from the Applications menu on the right side of the menu bar. This creates a menu/DA QuicKey to bring Remote Access to the front.
3. Select Status from the Windows menu. This creates a menu/DA QuicKey that brings the Remote Access Status window to the front.
4. Click the Disconnect button to create a click QuicKey.
5. Now press -Q to quit the program.
6. Select Stop Recording... from the QuicKeys menu. When the Sequence editor opens, your macro should look like the one in Figure 8-29.

NOTE: Did you notice that we don't have to tweak this macro or add any steps?

7. Click OK to close QuicKeys.

Use and variations: You can copy the steps from the HangUpOffice macro and paste them into RemoteBackUp if you want the latter macro to run unattended. You will also find this macro useful if you have established a connection with your office network that you want to disconnect. Keeping this macro separate lets you invoke it with a single keystroke and allow it to be called by the Decision option in the WindowAction extension at the end of another macro. We'll use this WindowAction feature later in this chapter when we create the GetMail1 and GetMail2 macros.

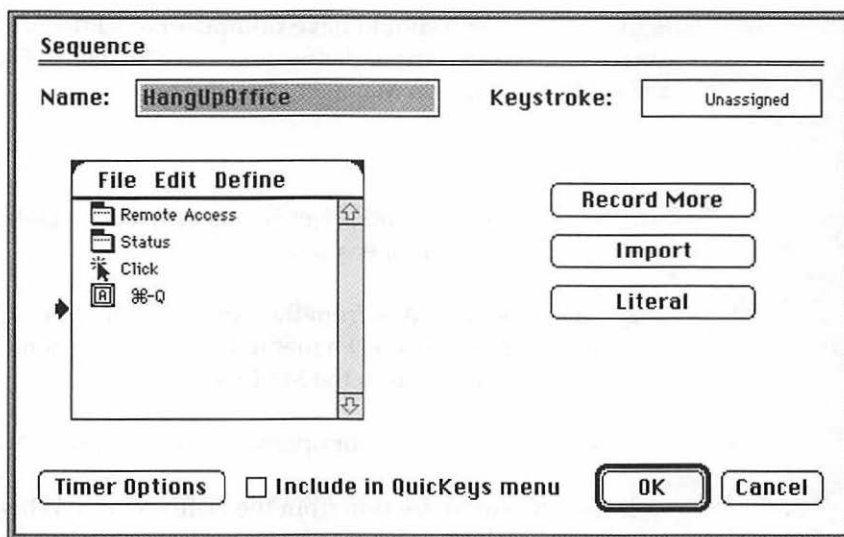


Figure 8-29: The completed HangUpOffice macro.

CompuServe Information Manager

CompuServe is one of the largest and the oldest commercial electronic networks. Unfortunately, it has the typical mainframe command line interface. In recent years, CompuServe has added Navigator and CompuServe Information Manager (CIM) to make the service more palatable to computer users whose first computer experience wasn't a terminal attached to a hulking computer down the hall. While Navigator was designed to run unattended during non-prime hours, CIM is completely interactive and requires a human to point and click the way through a session. This section describes macros that allow QuickKeys to stand in for you and run CIM unattended.

Get Mail for Information Manager

Even though CompuServe provides a powerful program in its Information Manager, it does not provide the user with an automated way of logging on and retrieving electronic mail; however, you can automate picking up your electronic mail with a macro. While this macro is very useful for picking up your electronic mail during non-prime time, it can also save you time if you are typing the keystrokes. Since the Information Manager is a graphical interface, QuickKeys can select the menu items and click the buttons faster than you can.

Before we begin: You should have CompuServe's Information Manager. You can run this macro under System 6.x or System 7. Open QuickKeys and select Information Manager from the pop-up menu.

Create the macro:

1. Open QuickKeys and select Sequence from the Define menu. The Sequence editor opens.
2. Select Menu/DA... from the Define menu. When QuickKeys prompts you to select a menu item, select the Send & Receive All Mail command from the Mail menu.
3. When the Menu editor opens, click OK or press .
4. Select WindowAction from the Define menu. When the editor opens, select Is from the Window Name pop-up menu and enter "Send & Receive All Mail Status" in the text box. Then click the Wait... radio button. You may want to enter the name of the dialog box so the macro will be easier to read. Click OK to close the WindowAction editor.

When the macro runs, this pauses the macro until the Send & Receive All Mail Status window appears (Fig. 8-30).

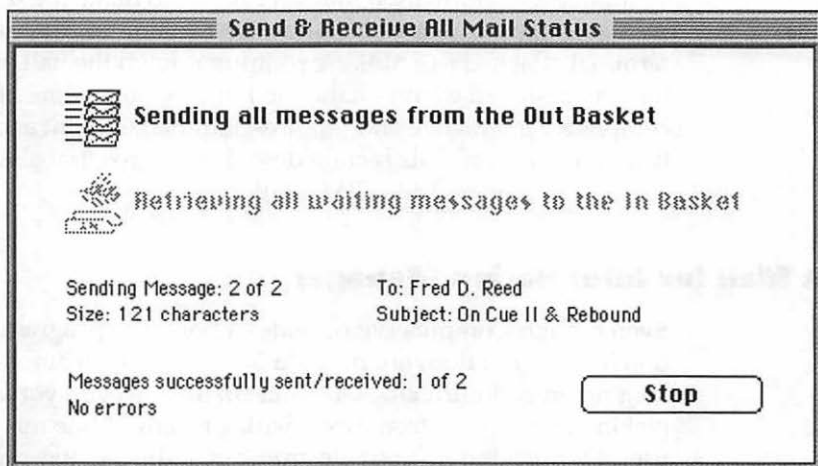


Figure 8-30: CompuServe Information Manager's Send & Receive window.

5. Select CursorWait from the Extensions menu. When the editor opens, select Is and arrow from the Cursor pop-up menus and

then select the Check Cursor and Check Cursor Mask checkboxes. Once the Send & Receive window appears, the macro resumes, but the CursorWait extension watches the spinning beach ball cursor and pauses until the cursor becomes an arrow. When CIM has gathered your mail to the program's in basket, the cursor changes back to an arrow, and the macro continues to run. Click OK to close the CursorWait extension.

6. Click the Literal button and press **Return**. This clicks the OK button in the Send & Receive window after all of the messages are gathered to the in basket.
7. Click the Literal button again and press **⌘-D**. This is the shortcut for disconnecting from CompuServe.
8. Select Wait... from the Extensions menu. When the editor opens, select Window changes from the pop-up menu. This pauses the macro until CIM can disconnect from CompuServe. Click OK or press **Return** to close the editor.

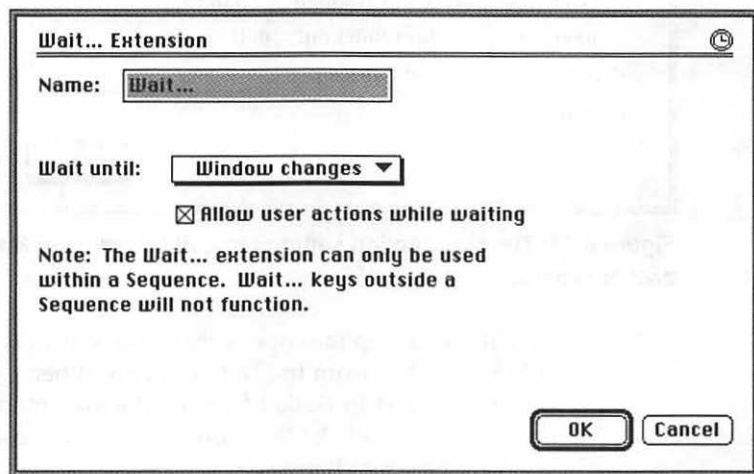


Figure 8-31: The Wait extension editor.

9. Let's add a step to our macro to quit the program if you haven't received any mail. We can add a MenuAction QuickKey that checks to see if there is any mail in the in basket and quits the program if there isn't.

Select MenuAction from the Extensions menu. When the dialog box opens, click the Select Menu button, and select the In Basket

menu item from the Mail menu (Fig. 8-32). Click the Is Not Enabled button; this makes QuickKeys check to see if the In Basket menu is not enabled.

Now click the Decision button and enter Quit in the Pass QuickKey text box. This sets the action that the MenuAction QuickKey performs. If the In Basket menu item is not enabled, your sequence automatically quits the Information Manager. If you have mail in your in basket, however, the sequence leaves the program open with the In Basket window open.

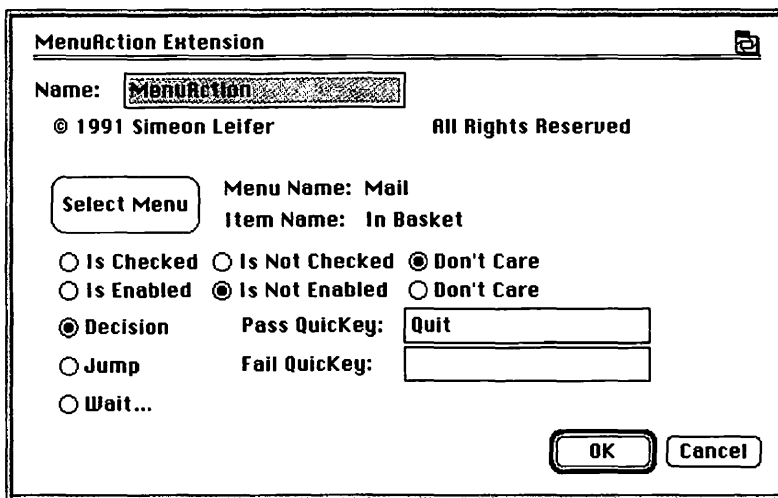


Figure 8-32: The MenuAction settings to quit Information Manager if the in basket is empty.

10. Now we'll add a step that opens the in basket if you have mail. Select Menu/DA... from the Define menu. When QuickKeys prompts you, select In Basket from CIM's Mail menu. When the Menu editor opens, click OK or press **Return** to close the Menu editor and save your changes.
11. Name your macro and assign a keystroke. Your macro should look like Figure 8-33.

Variations: If you want to run this macro while you sleep, you will need to make a couple of modifications to the beginning of the macro. In addition to the File or Finder Event QuickKey to open the program, you will need to add a WindowAction QuickKey that pauses the macro until the program is completely loaded. You should also add a click in the

menu bar or on the CompuServe Information Manager splash screen that comes up when you first open the program, since it won't go away until you click. You will also need to add a timer to the macro so it will run while you are asleep.

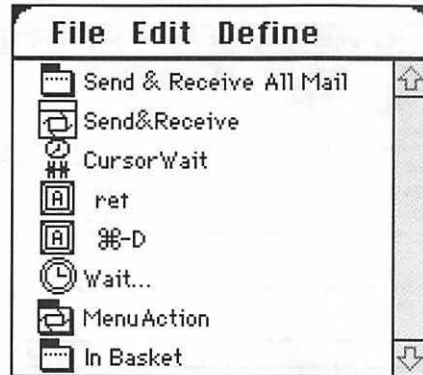


Figure 8-33: The completed Get&SendMail macro for CompuServe Information Manager.

Set Up a Reply

Once you have mail in your in basket, you can automate the process of replying to the message. When you click the Reply button in the CIM message window, the Reply window opens on top of the message window. I have a full-page display and have created a macro that drags the reply window down so I can see the message I'm replying to while I'm composing the reply. Even if you have a smaller screen, you can modify this macro to adjust the window size for you. Once you have done that you will be able to move the windows around to reply to a message with a single keystroke.

Before we begin: As before, you should have CIM installed. You'll need to open QuicKeys and select Information Manager from the pop-up menu.

Create the macro:

1. Open a message window in CompuServe Information Manager.
2. Select Record Sequence from the QuicKeys menu.
3. Click the title bar of the message window and drag it to the home position (the upper-left corner of your screen below the menu bar).

NOTE: Moving the window into the home position ensures that the Reply button will be in the right location on screen every time QuickKeys clicks on it.

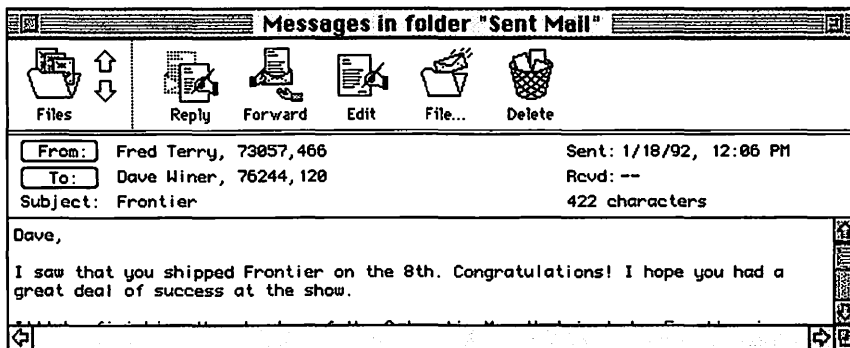


Figure 8-34: A message window in CIM.

4. Click the Reply button in the message window (Fig. 8-34).
5. When the Reply window opens, click on the title bar and drag it below the message window.
6. Select Stop Recording from the QuickKeys menu. When the Sequence editor opens, you'll see that QuickKeys created three click QuickKeys. You might want to rename the QuickKeys so the macro will be easier to read (Fig. 8-35).

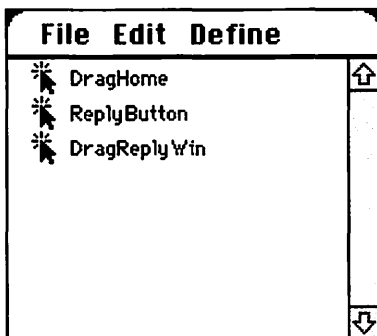


Figure 8-35: The final SetUpReplyWin macro.

7. Name the macro and click OK or press **Return** to save your changes.

Variations: As I mentioned in the previous description, you may want to modify this macro to resize the windows if you have a small monitor. Although I don't discuss it in the next section, I've included a macro called SetUpAOLReply in the America Online keyset that changes the window sizes as it sets up a reply. You could also extend this macro to automatically open all of the messages in the in basket and set up reply windows for each one.

America Online

A more recent arrival on the commercial network scene is America Online. AOL, as its subscribers refer to it, is every Macintosh user's dream service. It provides a number of the same services as CompuServe, but it is almost completely a graphical interface. You reach different forums by clicking buttons and names in scrolling lists. The interface makes it a joy for first-time telecommunicators and rabid mouse-lovers, but a pain for macro creators and power users who want to do everything from the keyboard. Because the window sizes and contents can change whenever they update or add services and the buttons in the windows aren't conventional Macintosh buttons, it is very difficult to create macros for AOL. If you practice a couple of tricks, you can automate your on-line sessions and log onto AOL while you sleep.

I've included macros for both America Online 1.0 and America Online 2.0 in this chapter and in the keysets on the enclosed disk. While I was working on this book, America Online 2.0 was being beta tested; it was just beginning to ship as I was finishing the book.

Get Electronic Mail

Creating a macro to collect your mail automatically from America Online 1.0 poses a special challenge. America Online has a keyboard shortcut for opening your mailbox and a Read All button; it does not have a command that will let you save all of your new messages to your flashbox—your on-disk mailbox. We are going to create a suite of sequences that can automatically collect all of your electronic mail and disconnect from America Online. Although we will create these macros to be run while you are sitting at your Macintosh, we'll discuss automating them later in the chapter.

QuickKeys has no way of getting the number of messages in your mailbox, inserting that number in a repeat, and repeating the sequence until

all of your mail is collected, but you can create a pair of sequences that check to see if all of the messages have been saved to your flashbox. Why create a pair of sequences? Because we need a sequence that plays itself again and a sequence cannot do that. With a pair of sequences, each one will save the frontmost message and then check to see if your mailbox is the next window. If it isn't, the sequence calls its twin which does exactly the same thing. When the New Mail window comes to the front, the sequence quits America Online. First we'll create the two sequences that save your mail. Then we'll create a sequence that quits America Online. Finally, we'll create sequences that log onto America Online and check for new mail.

Before we begin: As before, you should have America Online 1.0 installed and be connected to America Online with at least one electronic mail message window open (Fig. 8-36). Open QuickKeys and select America Online from the pop-up menu.

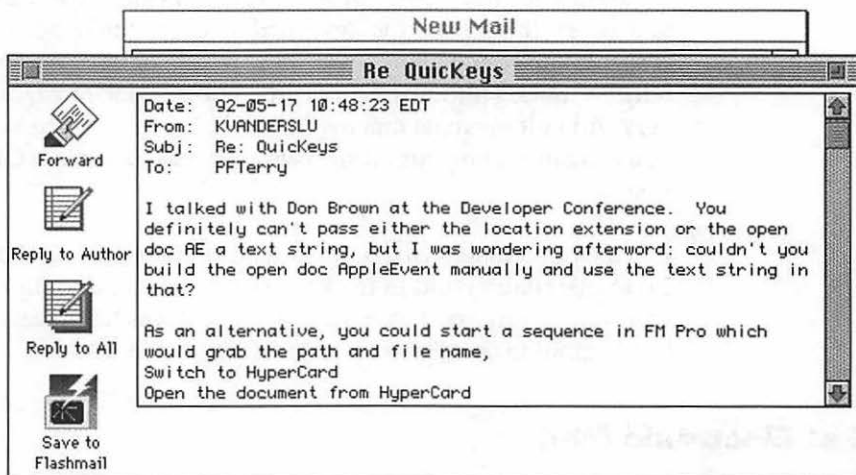


Figure 8-36: An America Online electronic mail message window. Notice the New Mail window behind it. This is the window we will be testing for with our macro.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Click the Save to Flashmail icon.
3. When America Online has saved your mail, a dialog box appears to let you know that your mail has been saved. Press **Return** to close the dialog box.

4. Press **⌘-W** to close the message window.
5. Select Stop Recording from the QuicKeys menu. If you are creating this macro as you read, you may also want to save your macro at this point and sign off from America Online so you will not run up connect charges. Name your macro GetMail1.
6. If you have closed QuicKeys, reopen it now and open your GetMail1 sequence. Select WindowAction from the Extensions menu.
7. When the WindowAction extension opens, select Is from the Window Name pop-up menu and then enter "New Mail" in the text box (Fig. 8-37).
8. Click the Decision radio button and enter "SignOff" in the Pass QuicKey text box. When you play the macro, this causes the WindowAction extension to play the SignOff QuicKey if the frontmost window is named New Mail.

Now enter "GetMail2" in the Fail QuicKey text box. If the window behind the message that your sequence just saved is not the New Mail window, this will play the GetMail2 sequence.

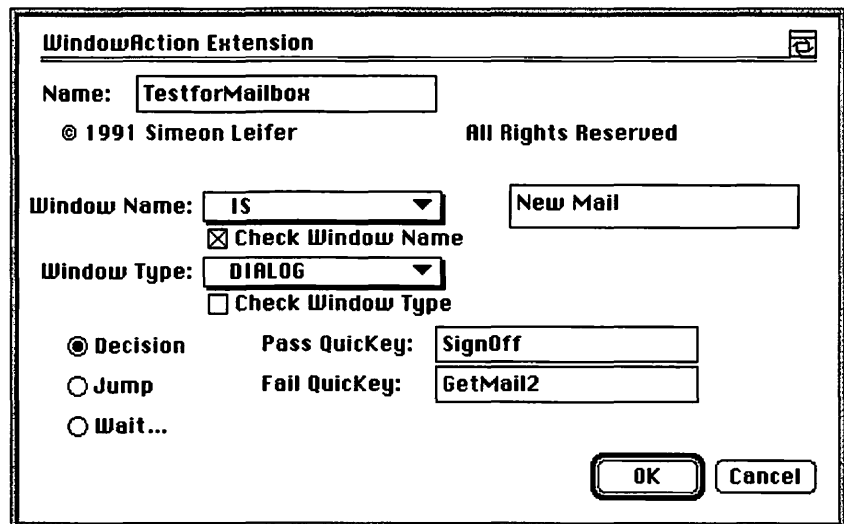


Figure 8-37: The WindowAction extension configured for the GetMail1 sequence.

9. Close the WindowAction extension dialog and name the sequence GetMail1. This is crucial for the macro to work correctly. Your completed macro should look like Figure 8-38.

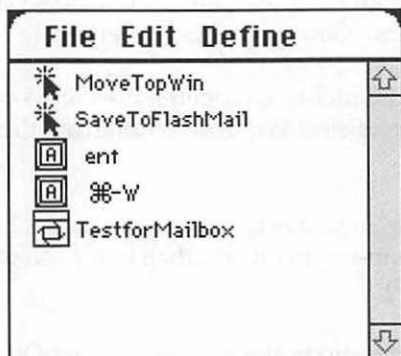


Figure 8-38: The final GetMail1 sequence.

Now we need to create the twin sequence that is played if the New Mail window has not come to the front. This sequence is exactly like the GetMail1 sequence with the exception that the fail QuickKey in the WindowAction extension says GetMail1.

1. With QuickKeys still open, select the GetMail1 sequence and select Copy from QuickKeys' Edit menu or press **⌘-C**.
2. Select Paste from the Edit menu or press **⌘-V**.
3. Open the new GetMail1 sequence and change its name to GetMail2.
4. Open the WindowAction sequence and change the fail QuickKey to GetMail1.
5. Close the sequence.

The pair of sequences we just created take care of saving your new mail, but we need to create the SignOff macro that they play when all of the mail has been saved to your flashbox. You should still have the America Online program open.

1. Open QuickKeys and select Sequence from the Define menu.
2. When the Sequence editor opens, select Menu/DA... from the Define menu.

3. When QuickKeys prompts you to select a menu item, select Quit from the America Online File menu and then click OK when the Menu/DA editor opens.
4. Click the Literal button and press **Return**. When you are connected to America Online and select Quit from the File menu, America Online asks you if you are sure that you want to sign off. This QuickKey clicks the Yes button in that dialog box.
5. Select CursorWait from the Extensions menu. When the editor opens, select Is and Arrow from the two pop-up menus and make certain that the Check Cursor and Check Cursor Mask checkboxes are selected. Click OK to close the CursorWait editor to save your changes. This QuickKey pauses the sequence until America Online disconnects from the service and the window appears that gives you the option of working off-line. The next QuickKey clicks the Quit button in the window.
6. Click the Literal button again and press **Return**.
7. Name the sequence "SignOff" and click OK to close the editor window and save your changes. Your final sequence should look like Figure 8-39.

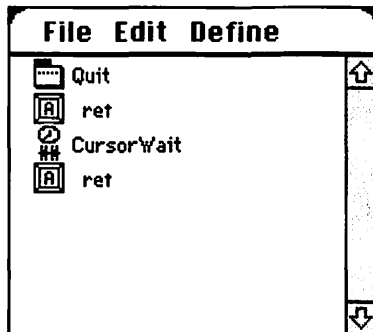


Figure 8-39: The final SignOff sequence.

Now that you have created the SignOff sequence, you are now ready to create the sequences that launch America Online and check for new mail. We will need to create two sequences: one to launch America Online and connect to the service, and a second to open the New Mail window and play the GetMail sequences.

1. Open QuickKeys and select Universal from the pop-up menu. You should save the sequence to Universal keyset so it will always be available.
2. Select File... from the Define menu. When the dialog box appears, navigate to the America Online program and select it. If you are using System 7, you can select FinderEvents from the Extensions menu. When the extension appears, configure the Finder Event to open the program. Click OK to close the editor.
3. Select WindowAction from the Extensions menu. When the WindowAction editor opens, select Contains from the Window Name pop-up menu, enter "America Online" in the text box, and then select the Check Window Name checkbox and the Wait... radio button. When you play the sequence, QuickKeys pauses until the America Online window appears. Otherwise, the sequence would continue to play and—depending on the speed of your Macintosh and hard drive—would be finished before the window even appeared. Click OK to close the WindowAction editor.
4. Select Alias... from the Define menu. When the Alias editor appears, press **Return** while holding down the mouse button. This QuickKey clicks the Sign On button in the America Online window.
5. Select Pause from the Define menu and enter 45 in the Pause dialog. Click OK to close the dialog and save your changes. This may seem like too much time, but you are waiting for the modem to dial (this will take longer if you have a rotary phone), for the America Online modem to answer, and for the service to recognize your user name and prompt you for the password.

The amount of time you enter here is completely dependent on the modem you are calling and the time of day you call. If you are calling a connection that is not very busy, 45 seconds is probably too long. However, if you are calling a connection that is sometimes busy, 45 seconds will not be enough time. If you plan to run the macro at 2:00 a.m., the connection may be made very quickly. To make the most accurate guess possible, you should log on at the time of day you plan to run the macro and time each step as you make the connection.

TIP: *If the spinning beach ball never stops spinning when you are logging in, you may be able to substitute a CursorWait extension in place of the Pause. But I should warn you that I have noticed QuicKeys tries to run the rest of the sequence if it is watching for the Arrow cursor and the spinning beach ball pauses. Even though you can't see it, the beach ball changes to an arrow briefly, and CursorWait triggers QuicKeys to continue the macro.*

6. Select Text... from the insert Define menu. When the Text editor opens, type your password followed by **[Return]**. Press **[Shift-Tab]** to select the Name box and enter "password" so you'll know what the QuickKey is for. Click OK to close the editor.
7. Select WindowAction from the Extensions menu. Select Is from the Window Name pop-up menu and enter "Welcome" followed by your user name and an exclamation point. Select the Check Window Name checkbox and the Wait... radio button. This QuickKey pauses the sequence until the America Online welcome window appears. Click OK to close the editor.
8. Select MenuAction from the Extensions menu. When the editor opens, click the Select Menu button and select Read New Mail from the Mail menu. Select the Is Enabled and Decision radio buttons and then enter "NewMail" in the Pass QuickKey text box and "SignOff" in the Fail QuickKey text box. This QuickKey checks the Read New Mail menu to see if you have any mail. If the menu is enabled, it plays the NewMail sequence. If the menu isn't enabled (i.e., it is gray), the QuickKey plays the SignOff sequence.
9. Check your settings for the MenuAction extension against those in Figure 8-40. Name your QuickKey and click OK to close the editor. Your LogOnAOL sequence should look like Figure 8-41.

We are now ready to create the final sequence of the suite for getting your electronic mail. The NewMail sequence opens the New Mail window, clicks the Read All button (Fig. 8-42), and then calls the GetMail1 sequence. You may want to copy the sequence from the companion disk and follow along as I explain how it works. If not, you will need to reconnect to America Online to create this sequence. Since you need electronic mail to create this sequence, you may have to send yourself mail.

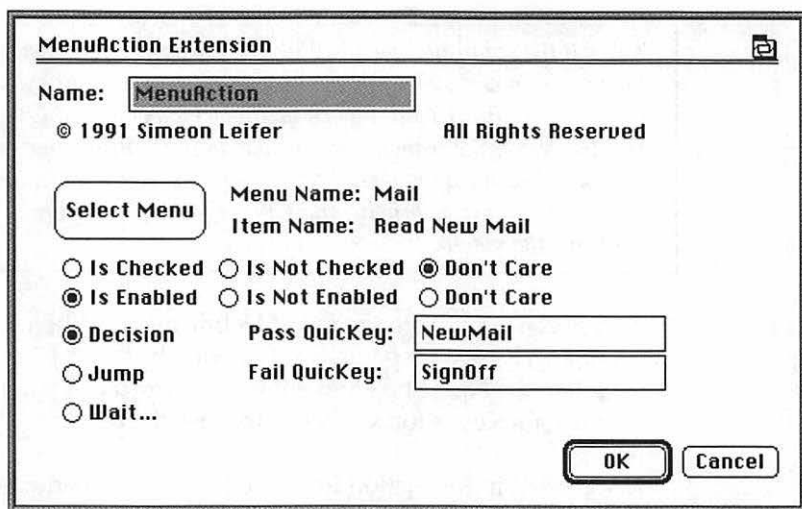


Figure 8-40: The settings for the MenuAction extension.

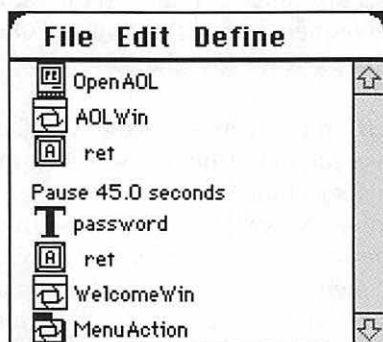


Figure 8-41: The final LogOnAOL sequence.

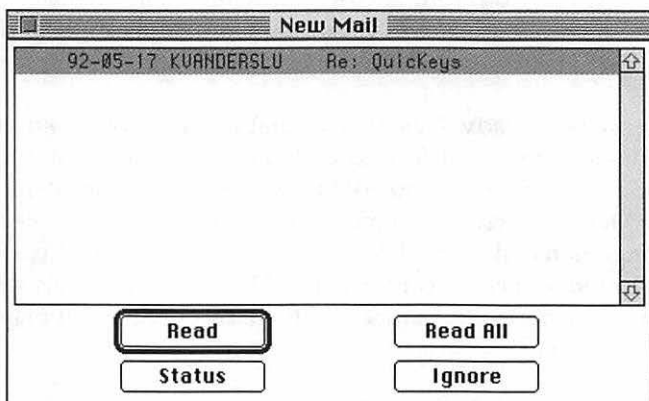


Figure 8-42: The New Mail window in America Online.

1. Open QuickKeys (if you have closed it) and select America Online from the pop-up menu. Open the NewMail sequence if you have copied it from the companion disk (Fig. 8-43).

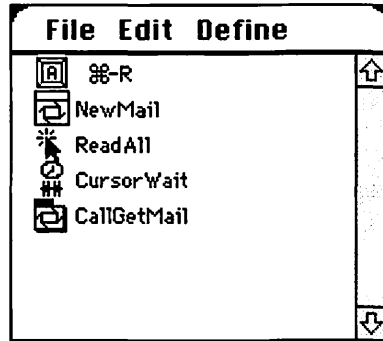


Figure 8-43: The NewMail sequence.

2. The -R alias selects the Read New Mail menu item and opens the New Mail window.
3. The NewMail WindowAction QuickKey pauses the sequence until the New Mail window opens.
4. The ReadAll click QuickKey clicks the Read All button in the New Mail window. This will open all of your messages at once.
5. The CursorWait QuickKey pauses the sequence until the cursor changes from a spinning beach ball to an arrow.
6. The CallGetMail MenuAction QuickKey is inserted here to play the GetMail sequence (Fig. 8-44). The QuickKey checks to see that the Read New Mail menu is not enabled and then calls the GetMail1 sequence.
7. Click OK to close the NewMail sequence.

You have now created a suite of sequences that can connect to America Online and collect your electronic mail. All you need to do is set a timer option for the LogOnAOL sequence so it can log on automatically while you are busy eating dinner or sleeping.

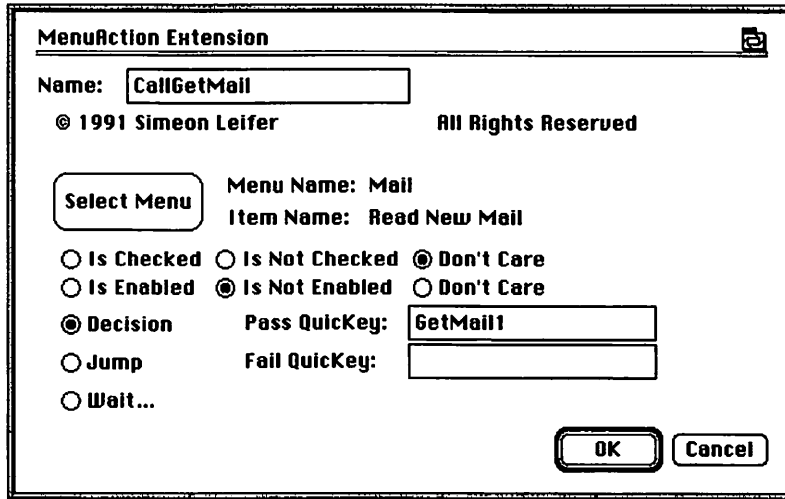



Figure 8-44: The CallGetMail QuickKey.

Variations: Though it may seem a bit klunky to use a pair of sequences to save messages and test for the New Mail window, it is the only way I know of to get around a sequence's inability to play itself again. Whenever you think you need a sequence to play itself again, you should try duplicating the sequence and letting one sequence call the other.

America Online 2.0 contains a FlashSession feature that allows you to automate collecting electronic mail and downloading files. You can also specify the time you want America Online to connect to the service and collect your electronic mail. While the new version of America Online does most of the hard work we have done in our macro, you may want to modify the LogOnAOL sequence for use with it. For instance, America Online cannot launch itself at a specified time, nor does it have as many time options as our sequence does.

Open New Files Folder

In both America Online 1.0 and 2.0, you can create -key shortcuts for the forums you visit frequently. For instance, you can add the Macintosh Utilities forum or the Macintosh Development forum under the Go To menu. America Online will automatically assign a keystroke between 1 and 0. Pressing the keystroke opens the forum window. While this does make the service more convenient, you can create macros to open more than just the ten forum windows allowed by America Online's shortcuts. You can even take this a step further and automate everything you do when you open a forum window. If you always check the new

files in a forum, you can create a sequence to open the forum window and then open the new files folder.

Before we begin: You should have either America Online 1.0 or 2.0 open. You should have the Auto-scroll Incoming Text option turned off in the Preference dialog box; this will making clicking in the forum window more accurate. If you are going to walk through creating the macro, you will need to be connected to the service. Open QuicKeys and select America Online from the pop-up menu.

Create the macro: The CursorWait extension is necessary to incorporate the correct amount of time for QuicKeys to pause. The time it takes to open the folder may vary according to the speed of your modem and how busy the service is, and you can be assured that it will never be the same any two times. The WindowAction extension pauses the sequence until the New Utilities/Desk Accessories files window opens. The CursorWait extension pauses the sequence until the cursor changes to an arrow; otherwise, the sequence would continue to play while the file information is still coming to your Macintosh from AOL.

1. Select Record Sequence from the QuicKeys menu. If you don't want to incur connect charges while you are creating the macro, you can copy the sequence from the companion disk and follow along as we create it.
2. Press **⌘-K** to open AOL's Keyword window (Fig. 8-45). The Keyword window lets you enter the keyword abbreviation for the forum you want to go to. If you know the keyword of the forum, it saves you the trouble of clicking through all of the folders and windows.
3. When the Go to Keyword window appears, type "mut" and press **Return**. This is the keyword for the Macintosh Utilities forum. Notice that while you are waiting for the MUT window to appear, the cursor changes to a spinning beach ball.

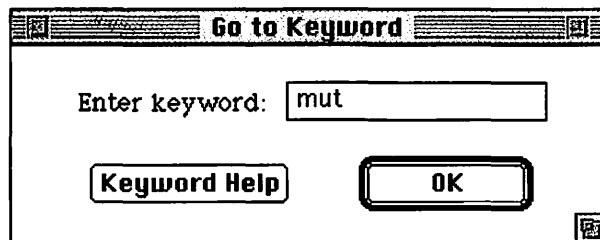


Figure 8-45: America Online's Go to Keyword window.

- When the Macintosh Utilities window appears (Fig. 8-46), wait until the cursor changes to an arrow and then double-click on the New Files area to open it.

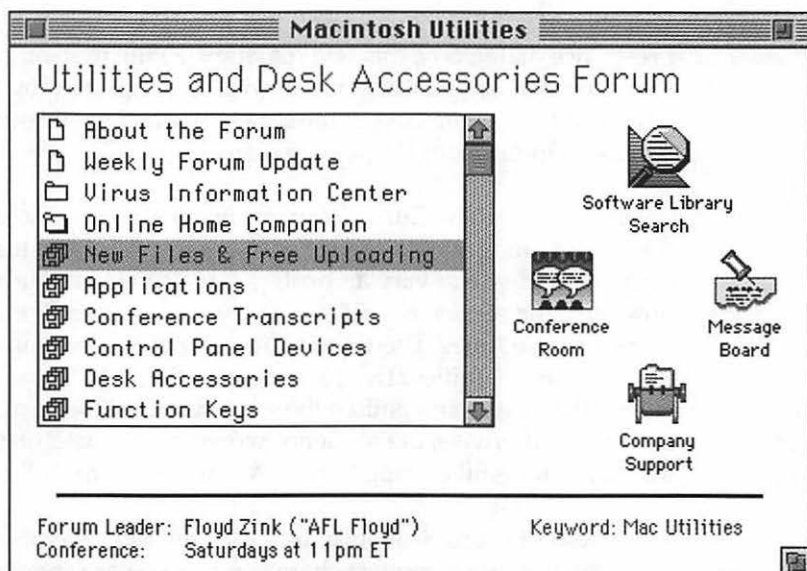


Figure 8-46: The Macintosh Utilities forum window on America Online

- When the New Files window appears (Fig. 8-47), you can select Stop Recording... from the QuickKeys menu. When the Sequence editor window opens, name your macro and close QuickKeys so you can disconnect from America Online.

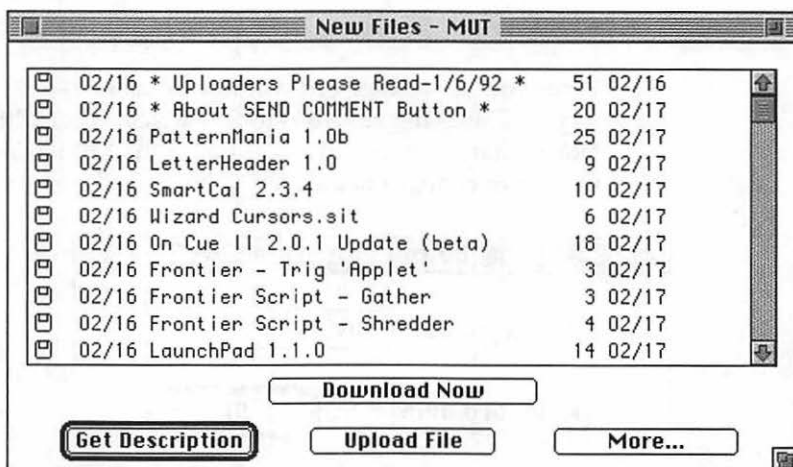





Figure 8-47: The Macintosh Utilities New Files window.

You have the basis for the New MUT Files macro, but we need to add wait conditions so the macro will not try to open windows before they appear on screen. We will use the CursorWait and WindowAction extensions to accomplish this.

1. Open QuicKeys again and open the New MUT Files sequence.
2. Move the insertion arrow between the -K alias and the text QuicKey.
3. Select CursorWait from the Extensions menu. When the editor opens, select Is not and ball from the pop-up menus then select the Check Cursor Mask checkbox. When you play the sequence, QuicKeys waits until the beach ball stops spinning and the Keyword window opens before it continues playing the sequence.
4. You'll notice that the  you pressed to select the OK button in the Go to Keyword window has been included in the text QuicKey. In the interests of modularity and having easy-to-read macros, open the text QuicKey and remove it.
5. Move the insertion arrow after text QuicKey and select Alias from the Define menu. When the editor opens, hold down the mouse button and press . Close the editor to save your changes.
6. Select WindowAction from the Extensions menu. When the WindowAction editor opens, select Is from the Window Name pop-up menu and enter "Macintosh Utilities" in the text box. Select the Wait... radio button and then close the editor. When you play the sequence, QuicKeys waits until the window appears before it continues. However, it is still possible for the sequence to continue the instant the window appears on screen, so we need to add another CursorWait QuicKey.
7. Select CursorWait from the Extensions menu. When the editor opens, select Is not and ball from the pop-up menus and then select the Check Cursor Mask checkbox. Now the sequence remains paused until all of the folder names have been read into the window and the cursor has changed to an arrow. Click OK to close the editor.
8. Name the sequence and close the Sequence editor to save your changes. The final New MUT Files sequence should look like Figure 8-48.

9. Click OK to close QuickKeys.

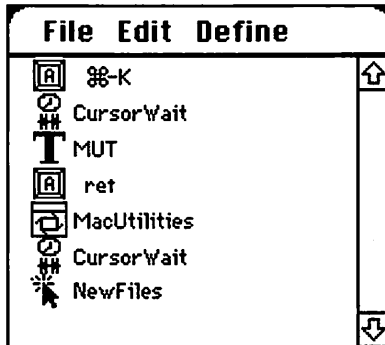


Figure 8-48: The final New MUT Files sequence.

NOTE: If Quantum, the owner of the America Online service, decides to add additional folders in the forum windows, this macro will break, and you will have to adjust the location of the click in the forum window to make the macro work again. An additional folder name was added to the MUT forum while I was writing this book. You should always pay close attention to any messages describing changes in the forums whenever you plan to let your Macintosh run on autopilot.

Variations: The most obvious variation on this macro would be to create a macro for each forum you check whenever you log onto AOL. You could do this by duplicating the New MUT Files sequence and changing the text, WindowAction, and click QuickKeys. You would enter a different keyword in the Go to Keyword window, change the window name in the WindowAction, and change the click location. If you know the keyword and window name of the forum, you would only need to log onto AOL to find the click location of the New Files folder. To help you keep your telecommunicating costs down (and your spouse or significant other happier), I've included sequences for most of the forums on AOL. Don't forget, however, that America Online may change the forums by the time you are reading this chapter.

Automated Communications

Once you have created macros to log onto the individual electronic services and retrieved your mail, it is easy to combine those into a larger macro. Since we have created the macros in this chapter in a modular

fashion, we can make simple changes to them so they work together to log onto each service, collect your electronic mail, log off, and shut down your Macintosh. We need to add a file or FinderEvents QuickKey and click QuickKey to the beginning of the CompuServe Information Manager's Get&SendMail sequence to launch the program and dismiss the splash screen. Then we will add a file or FinderEvents QuickKey to the end of the sequence to play the LogOnAOL sequence. We will also need to set the Timer option to play the sequence whenever we want CIM to collect our electronic mail. The only change we will have to make to the America Online GetMail suite is the addition of a ShutDown QuickKey at the end of the SignOff sequence.

Before we begin: You should be in the Finder. If you are using this sequence to collect your mail during non-prime time, you should either leave your Macintosh turned on and in the Finder or use a timer to start it up in the Finder. Open QuickKeys and open the Information Manager keyset. If you don't remember how to do this, you select Open from QuickKeys' File menu. QuickKeys opens the Keyset folder which is buried in your System folder. When the Information Manager keyset opens, it is added to QuickKeys pop-up menu.

Create the macro:

1. If necessary, scroll until you see the Get&SendMail sequence. Select it and select Copy from the Edit menu.
2. Using the pop-up menu, select the Universal keyset and paste the sequence. Since we want the sequence to play at a specified time from within any program, we need to make it a universal sequence. If we left it as an Information Manager-specific sequence, it would only play after we opened the Information Manager.
3. Open the Get&SendMail sequence and remove the wait, MenuAction, and menu QuickKeys from the end of the macro.
4. Click the Record More button in the Sequence editor. We need a click QuickKey to dismiss the splash screen. When QuickKeys closes and the flashing microphone icon tells you that QuickKeys is recording, click the pointer in the middle of the screen. Select Stop Recording... from the QuickKeys menu. QuickKeys opens again to the Click editor. Name the QuickKey ClickSplashScrn and then click OK to save your changes.

NOTE: This may seem like an oddball way to insert the click, but you'll notice that the click, sequence, and real-time menu items are disabled in the Sequence editor's Define menu. Clicking the Record More button takes care of closing QuickKeys and then reopening it when we are through recording. This forces QuickKeys to do all of the work, and we only have to move the QuickKey into position.

5. When the Click editor closes, you'll see that the click QuickKey is at the end of the sequence. Select the QuickKey and then select Cut from the Edit menu. Move the insertion arrow to the top of the sequence and select Paste.
6. Move the insertion arrow to the top of the sequence and select File... from the Define menu. When the dialog box opens, navigate to your copy of Information Manager, select it, and click Open. The path to the program is saved in the File editor (Fig. 8-49). If you are running System 7, you can use the FinderEvents extension in the same way. Click OK to save your changes.

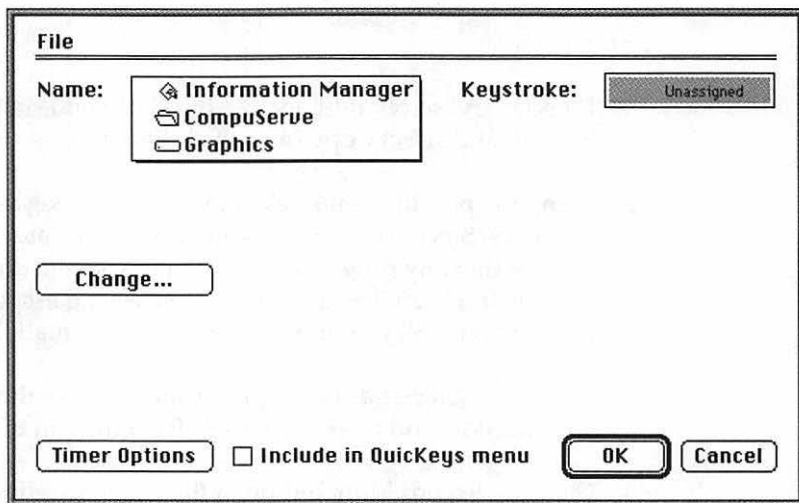


Figure 8-49: The File editor with the path to CompuServe's Information Manager.

7. Select CursorWait from the Extensions menu, select Is not and Ball from the pop-up menu, and then select the Check mask checkbox. This pauses the sequence until Information Manager has a chance to open and the splash screen appears. Click OK to close the editor.

8. Now move the insertion arrow to the bottom of the sequence and open on the \mathbb{E} -D alias. When the Alias editor opens, type \mathbb{E} -Q. In the original sequence, the alias only disconnected from CompuServe; Now, it quits Information Manager. Click OK to close the editor.
9. Select Pause from the Define menu and enter 60 seconds. You only need to enter enough time for your Macintosh to return you to the Finder. But be generous. After all, you probably aren't sitting and waiting on it to finish collecting the mail. Click OK to close the dialog box.
10. Finally, we need to add a QuickKey to call the LogOnAOL sequence. A MenuAction QuickKey gives us a means of testing to see that we are in the Finder and invoking the sequence. Select MenuAction from the Extensions menu. When the editor opens, click the Select Menu button and then select About this Macintosh (or About the Finder) from the Apple menu. The menu and menu item names are added to the dialog box. Select the Decision radio button if it isn't already selected and enter "LogOnAOL" in the Pass QuickKey text box. Enter the name of a sound QuickKey in the fail QuickKey. This will alert you to any problems with the macro.

NOTE: You leave both Don't Care buttons selected because the About this Macintosh menu item is never checked and is always enabled.

MenuAction Extension

Name:

© 1991 Simeon Leifer All Rights Reserved

Menu Name: \mathbb{A}
Item Name: About This Macintosh...

☐ Is Checked ☐ Is Not Checked ☒ Don't Care
☐ Is Enabled ☐ Is Not Enabled ☒ Don't Care
☒ Decision Pass QuickKey:
☐ Jump Fail QuickKey:
☐ Wait...

Figure 8-50: The settings for the MenuAction QuickKey.

11. Name the MenuAction QuickKey and click OK to close it and save your changes. Your final sequence should look like the one in Figure 8-51. Give it a different name such as CollectCISMail, so you can distinguish it from the original Get&SendMail sequence. Click OK to close the Sequence editor and save your changes. Then click OK to close QuickKeys.

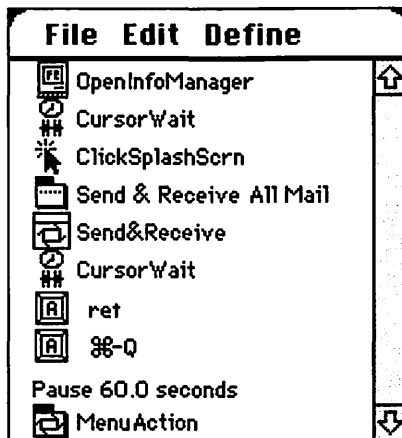


Figure 8-51: The final CollectCISMail sequence.

We have finished our modifications to the Information Manager and are ready to modify a copy of the America Online SignOff sequence so it will shut down your Macintosh after it has collected your mail.

1. Open QuickKeys and open the America Online 1.0 keyset.
2. Open the SignOff sequence.
3. Select Pause from the Define menu. When the dialog box opens, enter 45 in the seconds box and click the OK button to close it.
4. Select Shut Down from the Specials menu. When the editor opens, click OK to close it and save your changes. Now your sequence should look like Figure 8-52.
5. Click OK to close the sequence editor and save your changes. Then click OK to close QuickKeys.

Through a few modifications, you have chained several sequences together so they can automatically collect your electronic mail and shut down your Macintosh.

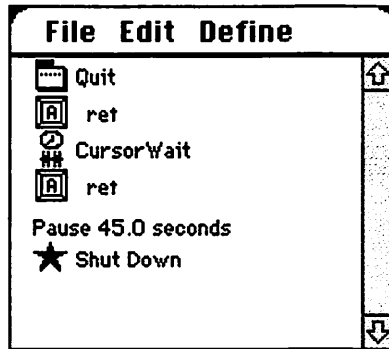


Figure 8-52: The modified SignOff sequence.

Variations: You could change these sequences in several different ways to make them more functional for your programs and hardware. For instance, you might want to put your Macintosh on a timer if you don't really want to leave your Macintosh running until it logs on and gets your electronic mail. One such timer is the PowerKey from Sophisticated Circuits. The PowerKey is a multiple outlet box that plugs into one of the ADB ports on your Macintosh. These are the Apple Desktop Bus ports that let you connect your keyboard and mouse to your Macintosh. When you plug all of your peripherals into the PowerKey outlets, they are automatically turned on or off whenever you turn your Macintosh on or off. A control panel that comes with the PowerKey lets you set times to start up or shut down your Macintosh or activate a keystroke, an external, or a QuicKey. You could let the PowerKey play your CollectCISMail QuicKey whenever it starts up your Macintosh. To do this, select QuicKeys... from the pop-up menu. A dialog box opens where you can select the QuicKey you want PowerKey to play (Fig. 8-53).

Once you set the QuicKeys event in the PowerKey cdev (Fig. 8-54), you will need to set a power on event to turn on your Macintosh before the QuicKey plays. You may find it simpler to use the PowerKey instead of setting the Timer option in the QuicKeys window since you set both the start up time for your Macintosh and the time the CollectCISMail sequence plays in the same dialog box.

With the PowerKey, you can start the Macintosh early in the morning and run the sequences that collect your electronic mail. When the final America Online sequence runs, PowerKey shuts off your peripherals with your Macintosh.

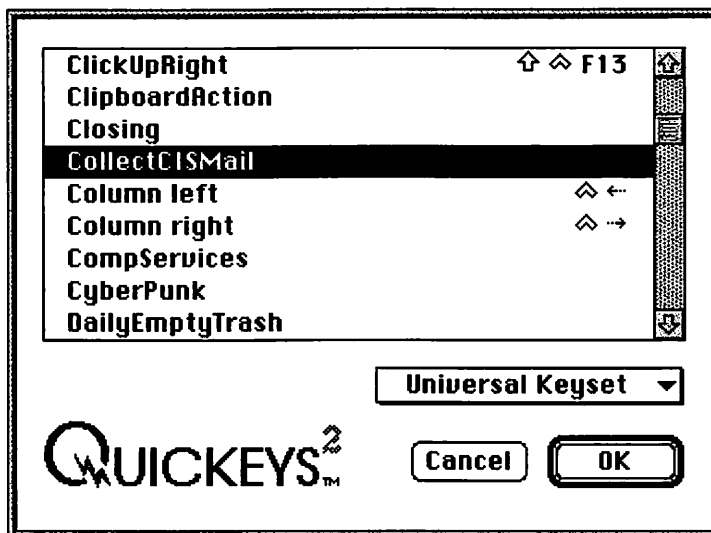


Figure 8-53: The QuickKeys selection dialog box in the PowerKey control panel device.

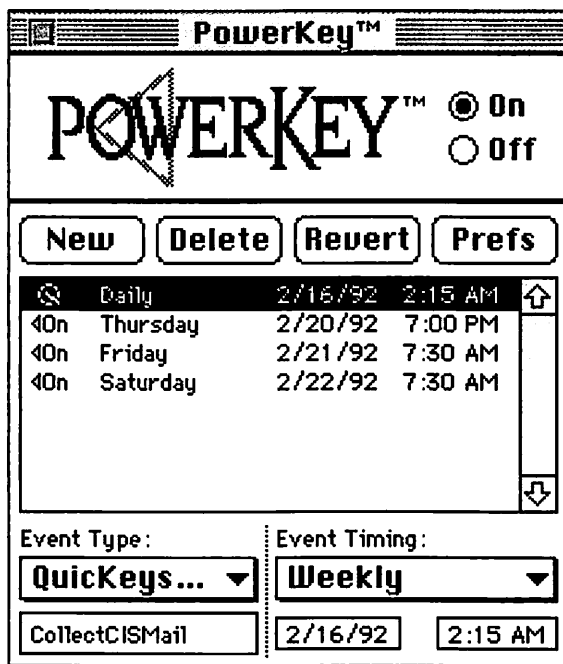


Figure 8-54: The PowerKey control panel.

You might prefer to have printed copies of your electronic mail waiting for you when you get up in the morning. You can modify the CollectCISMail sequence so that it calls a PrintMessages sequence before it plays to the LogOnAOL sequence (Fig. 8-55).

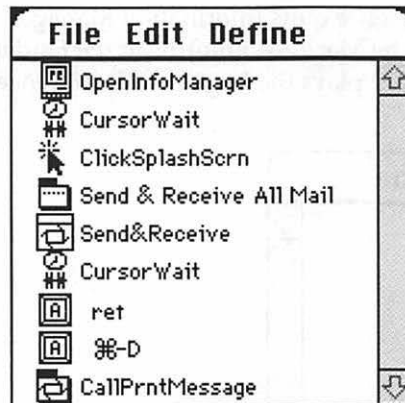


Figure 8-55: The modified CollectCISMail sequence. Notice that it calls the PrintMessages

You would replace the ⌘-Q alias with a ⌘-D to disconnect from CompuServe. You would also need to add a MenuAction QuickKey to the end of the sequence that would check to see if the Information Manager's In Basket menu is enabled (Fig. 8-56). If so, it calls the PrintMessages sequence.

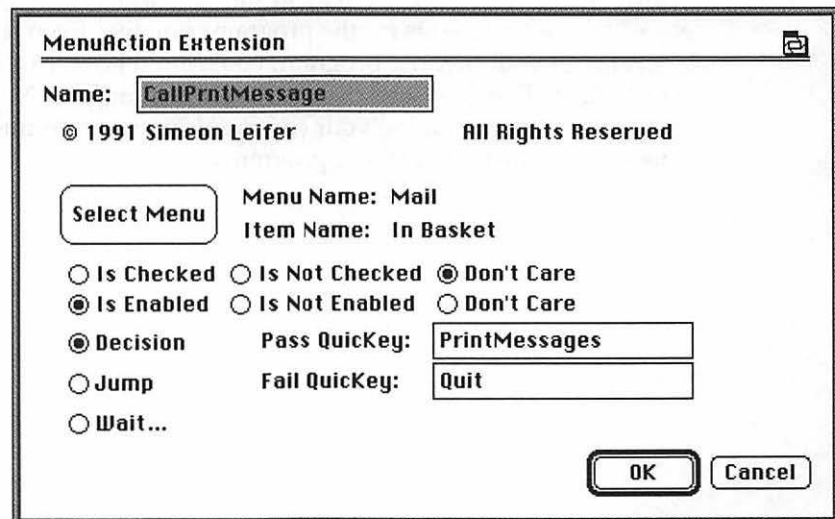


Figure 8-56: The MenuAction QuickKey that calls the PrintMessages sequence.

The PrintMessages sequence (Fig. 8-57) opens the In Basket window and then selects all of the messages in the in basket with a **⌘-A**. A **⌘-P** opens the Print dialog box, and a **Return** clicks the Print button. The WaitTilInBasket WindowAction QuickKey pauses the sequence until the printing dialog box closes and the In Basket window comes to the front. Then the sequence quits Information Manager. A pause QuickKey ensures that the Macintosh returns to the Finder before the CallLogOnAOL QuickKey plays the LogOnAOL sequence.

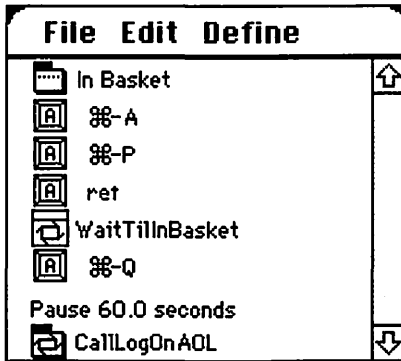


Figure 8-57: The final PrintMessages sequence.

What Next?

While the programs I have discussed in this chapter may not be the programs you use every day, you should check the companion disk to see if there are keysets for the programs you use. Even if I have failed to talk about your favorite program, you should be able to apply the techniques I've described to create your own macros. Now that we've talked about automating your communications programs, let's take a look at automating database programs.

9

CHAPTER

QuicKeys and Databases

Most relational databases are based on programming languages, and the databases you create with them can be designed to suit the needs of the user. For the end user, QuicKeys can simplify entering information and moving information between the database and other programs. For the database programmer or designer, QuicKeys can automate creating the database. QuicKeys is even more helpful with off-the-shelf database applications. You can use it to automate generating reports, add keyboard shortcuts, and set up a search.

In this chapter, I have created most of my examples in FileMaker Pro 2.0, but the macros are applicable to most database programs.

Automating Database Creation

In most of the chapters of this book, I have talked about how QuicKeys can make your use of programs more efficient. In Chapter 7, I discussed ways that QuicKeys could help you create and script stacks. While QuicKeys can help you automate data entry in a database program, you can use it very effectively while you are creating a database. Saying that some macros are better for creating databases than entering information in a database or manipulating its contents is a rather fine distinction. You will probably find these macros useful even if you only enter information in a database.

Adding Keyboard Shortcuts

As with any program, database programs don't always have keyboard shortcuts for the commands you use the most. While FileMaker Pro has a number of keyboard shortcuts, it is no exception. I generally find

myself reaching for the mouse to click the Part Label and Status Panel icons at the bottom of the FileMaker Pro window so I can see more of my database layout on screen (Fig. 9-1). Instead of fumbling for the mouse, you can create click QuickKeys to click these icons.

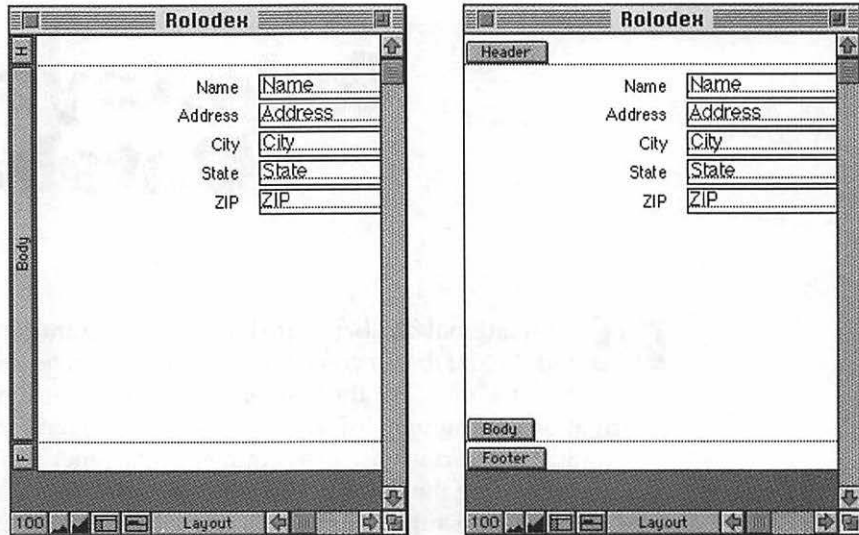


Figure 9-1: The Part Labels in the document window displayed vertically (left) and horizontally (right).

Before we begin: Open a FileMaker Pro document and select Layout from the Select menu. You need to do this to see the Part Label icon. You should also open QuickKeys and select FileMaker Pro from the pop-up menu.

Create the macro:

1. Select Record one QuickKey from the QuickKeys menu.
2. Click the Part Label control icon in the bottom-left corner of the FileMaker Pro document window (Fig. 9-2).

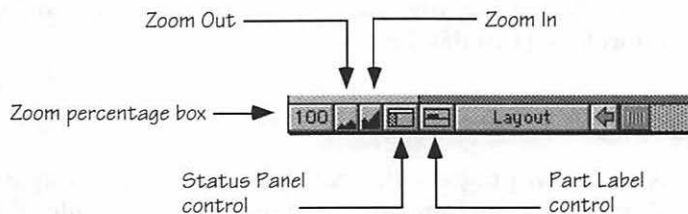


Figure 9-2: FileMaker Pro's Zoom, Status Panel, and Part Label controls. You will only see the Part Label control when you are in the Layout view.

- When the Click editor opens, you'll see that QuickKeys has measured the click from the top-left corner of the FileMaker Pro window (Fig. 9-3).

Click

Name: Keystroke:

Click: From: (78,301) from top-left corner
To: (0,-1) from current location

Window: Window #1 from front

Control area: None

Click time(s)

Hold down: ☐ ⌘ ☐ Shift ☐ Option ☐ Control

Timer Options ☐ Include in QuickKeys menu

Figure 9-3: The Click editor after you have clicked the Part Label icon.

- Click the Click button to open the Click Location editor. When it opens, you'll see that the top-left radio button in the small document window is selected. Click the bottom-left radio button (Fig. 9-4).

Click Location

Click Relative to:
☒ Screen
☐ Document

Drag Relative to:
☐ Screen
☐ Document

Mouse position
 H= Y=

Mouse position
☒ **Mouse position**
 H= Y=

Figure 9-4: The Click Location settings for the TogglePartLabel macro.

NOTE: You want to change the click location settings so QuickKeys clicks the correct part of the document window. If QuickKeys measured the click from the top-left corner, it would not click the icon whenever the document window was resized. QuickKeys might click within the window or outside of the window. By selecting the bottom-left radio button in the document window, you cause QuickKeys to measure the click from that corner. Now whenever you resize the window, QuickKeys will always click the icon. If you make macros for each one of the control icons, you will need to change the click settings in each one.

5. Click OK to close the Click Location editor.
6. Name your macro something descriptive, like TogglePartLabel, and assign a keystroke.
7. Click OK to close the Click editor.

While the QuickKeys window is still open, you can select Click from the Define menu to create another QuickKey. If you decide not to, you'll need to click OK to close QuickKeys and save your changes.

Variations: Closing the Status Panel can make even more room on screen (Fig. 9-5). You can make click QuickKeys for the Status Panel, Zoom Out, and Zoom In icons. If you are working on a small screen, the TogglePartLabel and ToggleStatus macros allow you to quickly close the Status Panel and change the Part Label display so you can see more of the layout on screen.

You might also want to create macros to click the Zoom Out and Zoom In icons. These let you quickly get a close up view of the document or zoom out to see more of the document on screen at once.

NOTE: You can never make assumptions about how a program behaves. When we made the ZoomIn and ZoomOut macros for Claris' sResolve spreadsheet in Chapter 5, you may recall that we had to add two CursorWait QuickKeys to accommodate the cursor changing to a spinning beach ball. The cursor in Claris' FileMaker Pro does not change to a spinning beach ball when you click on the Zoom controls.

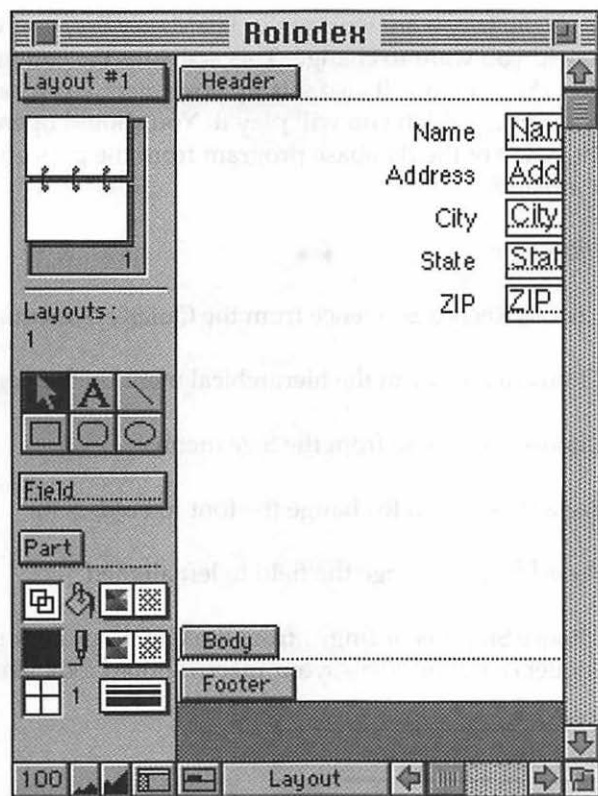


Figure 9-5: The document window with the Status Panel displayed. The document windows in Figure 9-1 have the Status Panel hidden.

You could also make a click QuickKey for the Zoom Percentage icon. It allows you to toggle between your current view and 100% view. In other words, if you have zoomed out 50% to see more of your document, you can switch to 100% by clicking the Zoom Percentage icon. Clicking the Zoom Percentage icon a second time toggles you back to 50%. You'll find a TogglePercent macro in the FileMaker Pro keyset on the enclosed disk.

Setting Up Fields

When you are designing the layouts for your database and reports in FileMaker Pro, you can specify several different attributes for the fields. For example, you can set the font, size, style, alignment, line spacing, and text color in text fields. If you have several fields that you want to format with the same attributes or settings, you can make short work of the changes with QuickKeys.

Before we begin: If you are working in FileMaker Pro, you will need to select a field you want to change. This activates the commands in FileMaker Pro's menus. It also sets up your macro so it begins recording at the point from which you will play it. You should open QuickKeys and select the name of the database program from the pop-up menu if you haven't already.

Create the macro:

1. Choose Record Sequence from the QuickKeys menu.
2. Choose a font from the hierarchical menu in the Format menu.
3. Choose a font size from the Size menu.
4. Press **⌘-Shift-B** to change the font to bold.
5. Press **⌘-[** to change the field to left aligned.
6. Choose Stop Recording... from the QuickKeys menu. When the Sequence editor opens, your macro should look like Figure 9-6.

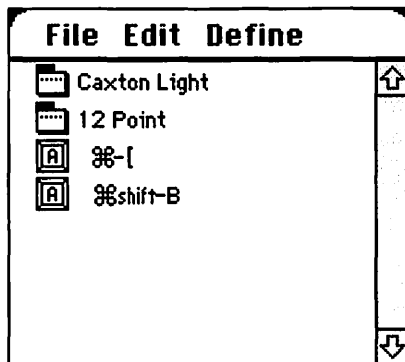


Figure 9-6: The final SetUpField macro.

7. Name your macro and assign a keystroke. Click OK to close the Sequence editor and then click OK to close QuickKeys and save your changes.

It might be tempting to open FileMaker Pro's Text Format dialog box (Fig. 9-7) and make all of your format selections there. While it may seem like the logical thing to do since you can change all of the settings in one dialog box instead of making several menu selections, QuickKeys

has trouble making menu selections from the pop-up menus. If you record the menu selections you made above from the pop-up menus in the dialog box, QuickKeys creates a series of clicks instead of menu QuickKeys. Why is this a problem? It is a little difficult to explain, but it is worth the trouble to understand.

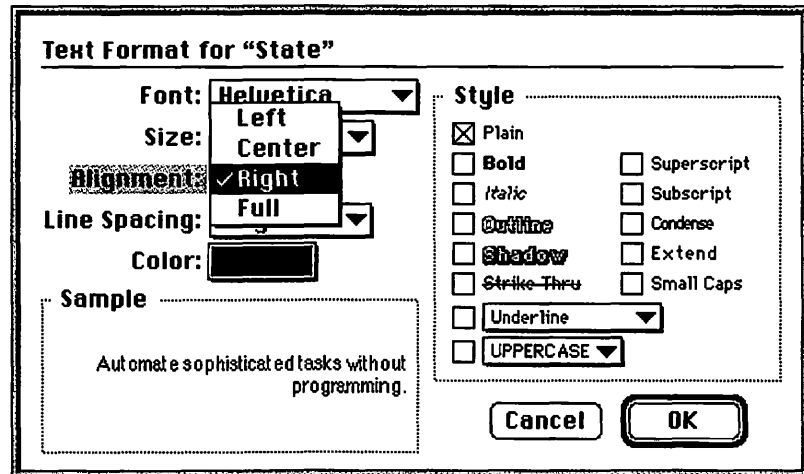


Figure 9-7: FileMaker Pro's Text Format dialog box which contains pop-up menus for Font, Size, Alignment, Line Spacing, and Color.

Click QuickKeys, as you may recall, are recorded as the distance from the beginning click to the end of the drag. When you play the click QuickKey, it always clicks and drags the same distance. The pop-up menus in the Text Format dialog box retain the command you choose. When you first choose a font, you may start with Helvetica and drag to Times. The next time you play your macro—even though your click begins at the same point in the dialog box—it will select a font that is the same distance from Times as Times was from Helvetica.

Yet another reason for choosing the commands from the menus is that you will not see the menu choices when you play the macro. You will see all of the selecting actions if you record the macro in the Text Format dialog box.

Variations: You can create several variations for this macro. If you have groups of text fields that you want set up with different attributes, you can create a macro for each one. You can also create macros that set the attributes for number, date, time, and picture fields. For instance, you could create a macro that sets a date field to automatically enter the date a record is modified and prevent that date from being changed (Fig. 9-8).

If the settings for a field are particularly complicated and you must enter them more than once, you should create a macro.

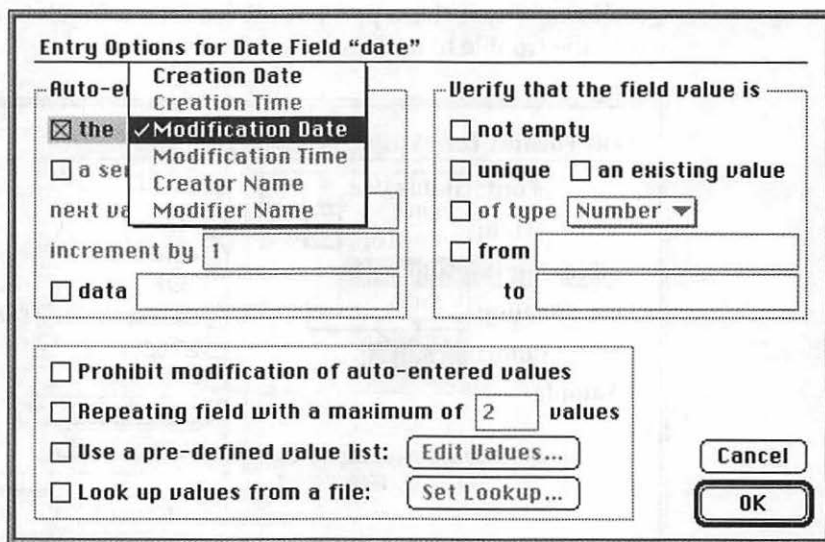


Figure 9-8: FileMaker Pro's Date Format dialog box.

If you routinely create databases with number fields that are set up for currency, you could make a macro to enter the same settings whenever you add a new number field. The real value of a SetUpNumber macro is that it remembers your settings exactly and enters the settings faster than you can. To create a SetUpNumber macro, you would record the following steps.

1. Choose Record Sequence from the QuickKeys menu.
2. Choose Number Format from the Number menu.
3. When the Number Format dialog box opens (Fig. 9-9), select the Format as decimal number radio button.
4. Select the Use thousands separator checkbox.
5. Select the Notations checkbox. If the Currency radio button isn't selected, click it.
6. Select the Fixed number of decimal digits checkbox.
7. Type "2" so the macro automatically enters the correct number of digits.

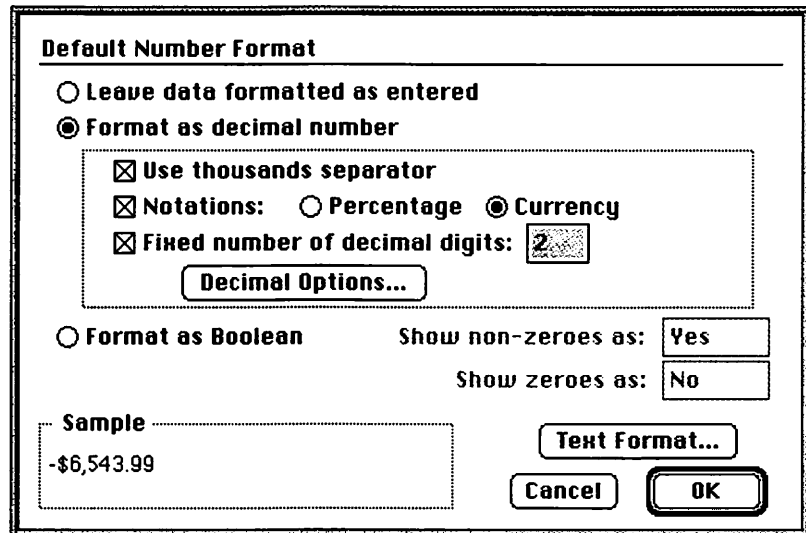


Figure 9-9: FileMaker Pro's Number Format dialog box with the settings already entered for the SetUpNumber macro.

8. Press **Return** to close the dialog box and save your changes to the number field.
9. Press **⌘-]** to format the field to right alignment.
10. Choose Stop Recording... from the QuickKeys menu. When the Sequence editor opens, your macro should look something like the one in Figure 9-10.

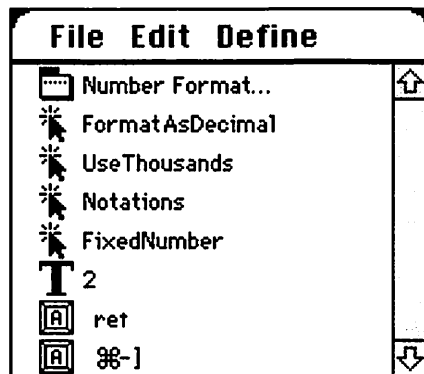


Figure 9-10: The final SetUpNumber macro. I've edited the Click QuickKey names to make the macro more readable.

11. Open the second click QuickKey (the UseThousands QuickKey in Fig. 9-10) and click the Control Area button.
12. When the Control Area dialog box opens, click the Only click if button is off radio button (Fig. 9-11). With this option selected, QuickKeys won't select the checkboxes if they have already been selected. This ensures that QuickKeys will always format the field correctly and won't unselect the Use Thousands checkbox.

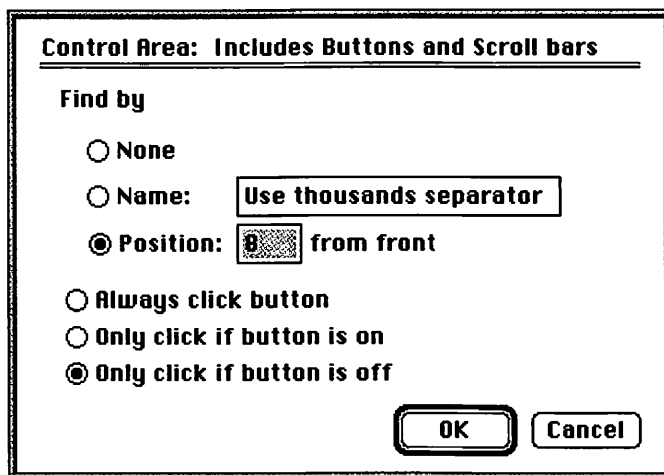


Figure 9-11: The Control Area settings for the UseThousands QuickKey.

13. Click OK to close the Control Area dialog box and click OK to close the Click editor.
14. Change the third and fourth click QuickKeys (the Notations and FixedNumber QuickKeys in Fig. 9-10) so they match the Control Area settings for the UseThousands QuickKey.
15. Name your macro and assign a keystroke.
16. Click OK to close the Sequence editor and then click OK to close QuickKeys.

Paging Through Database Records

FileMaker Pro has a number of keyboard shortcuts for commands and has commands that let you browse through individual records. You can press **⌘-Tab** to move to the next record or **⌘-Option-Tab** to move to

the previous record. However, FileMaker Pro doesn't have commands for moving to the first or the last record in a database, although you can click at the very top and bottom of the Bookmark's path to move to the first or the last record (Fig. 9-12).

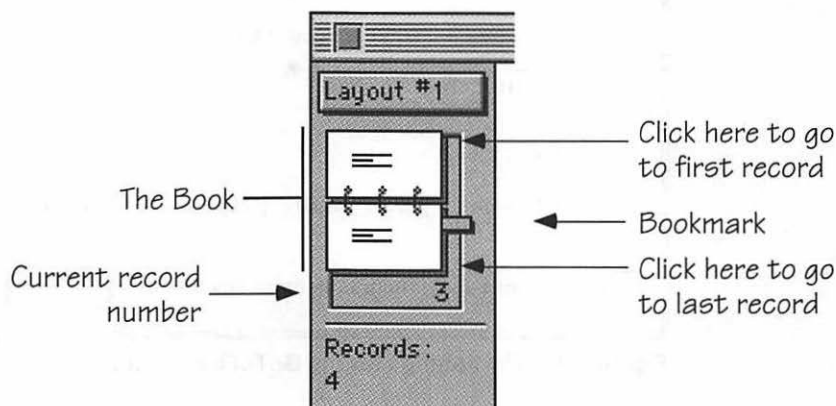


Figure 9-12: The Book icon in FileMaker Pro's Status Panel.

You can also click the current record number beneath the Book and enter the specific record to jump to. By creating a few macros, we can enhance FileMaker Pro's shortcuts for paging through database records.

Before we begin: If you are creating the navigation macros for FileMaker Pro, you will need to display the Status Panel.

Create the macro:

1. Select Record one QuicKey from the QuicKeys menu.
2. Click above the Bookmark next to the upper-right corner of the Book's top page.
3. When the Click editor opens, you can leave the settings as they are (Fig. 9-13). Give your QuicKey a name and assign a keystroke. You might want to assign **Home** as the keystroke if you have an extended keyboard.

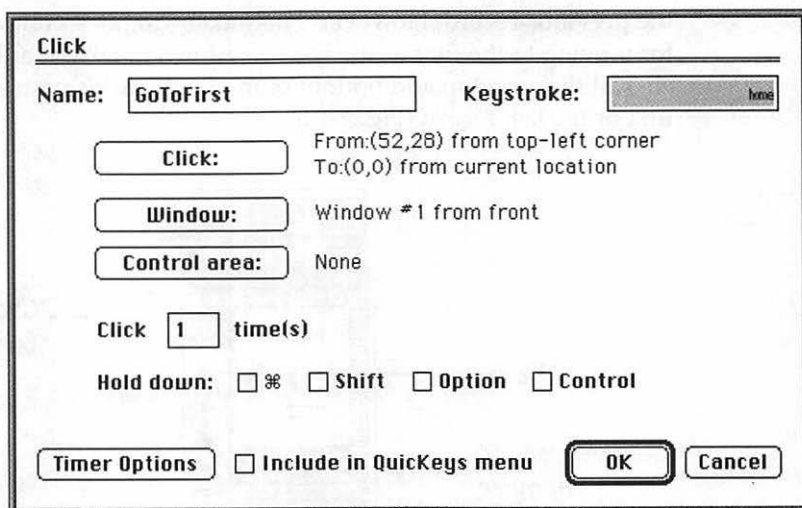


Figure 9-13: The settings for the GoToFirst macro.

NOTE: If you assign Home to the QuicKey, a dialog box may appear warning you that the keystroke has already been assigned to another QuicKey in the Universal keyset (Fig. 9-14). You will only see this dialog box when an application-specific keystroke conflicts with a keystroke in the application or Universal keyset. As I've mentioned elsewhere, it is okay to double-assign keystrokes when you want the keystroke to do something different in a specific program. When you double-assign keystrokes, QuicKeys will always perform the macro associated with the keystroke in the program, not the macro associated with the keystroke in the Universal keyset.

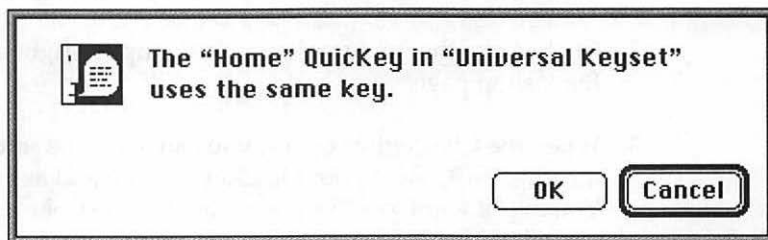


Figure 9-14: QuicKeys warns you if you assign a keystroke that you've already assigned in the Universal keyset.

4. Click OK to close the Click editor. Since you know that you want to make a macro that jumps to the last record in the database, let's start from QuicKeys.

5. Select Click from the Define menu in QuickKeys.
6. When QuickKeys closes, click below the Bookmark next to the lower-right corner of the Book's bottom page.
7. When the Click editor opens, name your QuickKey and assign a keystroke. You might want to assign **[End]** if you have an extended keyboard.

NOTE: If you use the **[Home]**, **[End]**, **[PgUp]**, and **[PgDn]** keys for navigating in other programs, you could reassign FileMaker's shortcuts for moving to the previous and next records so they form a consistent group of shortcuts. The simplest way to do this is create two alias QuickKeys. One would send **[⌘]-[Tab]** to FileMaker whenever you pressed the **[PgDn]** key (Fig. 9-15), and the other would send **[⌘]-[Option]-[Tab]** whenever you pressed the **[PgUp]** key.

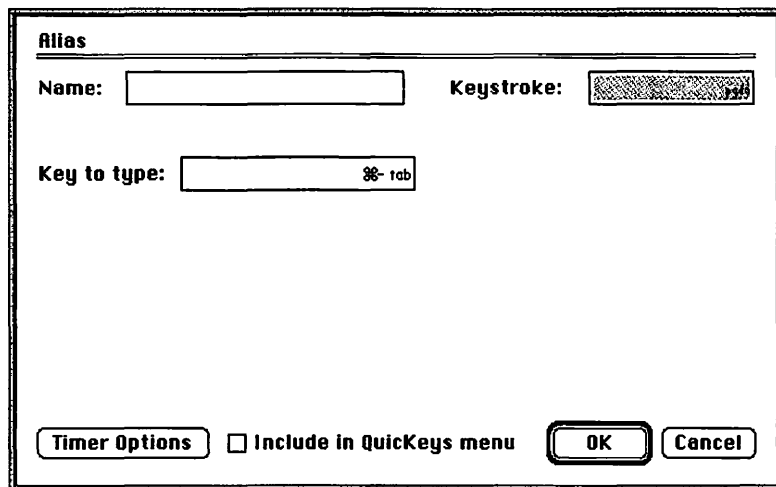


Figure 9-15: The alias QuickKey to move to the next record in a FileMaker database.

8. Click OK to close the Click editor. You might want to create a macro that lets you jump to a specific record number in the database. As with the preceding macro, you can create this from QuickKeys.
9. Select Click from the Define menu.

10. When QuickKeys closes, click on the current record number below the Book icon (Fig. 9-12). You will notice that the number is highlighted.
11. When the Click editor opens, name the macro something descriptive and assign a keystroke.
12. Click OK to close the Click editor. Click OK to close QuickKeys and save your changes.

Variations: If you tend to work with the Status Panel closed to save space, you may want to make a sequence version of these navigational macros that displays the Status Panel, jumps to the last record in the database, and then hides the Status Panel. You can do this by combining the macros we've already created for toggling the Status Panel and jumping to the first and last records in a database.

Another variation would be to create macros that navigate between the different layouts in a database (Fig. 9-16). FileMaker Pro already has a group of shortcuts for switching between the Browse, Layout, Preview, and Find views. **⌘-B** switches to the Browse view, **⌘-L** switches to the Layout view, **⌘-U** to the Preview view, and **⌘-F** to the Find view. If you are designing the layouts and reports in a database, however, you will also benefit from having keyboard shortcuts that switch between layouts. For instance, you could make macros to select the layouts you are working in while you are creating a database. If you are copying portions of one layout to another, the keyboard shortcuts would save you the time it takes to select a layout from the pop-up menu.

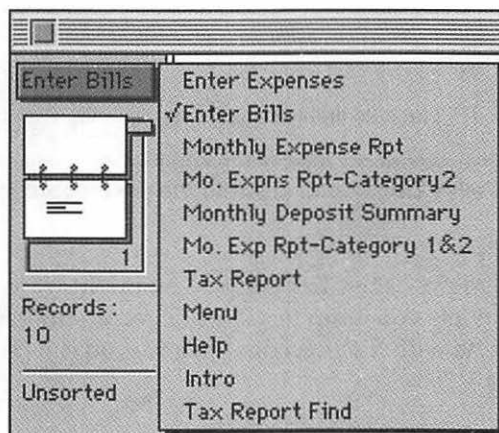


Figure 9-16: The Layout pop-up menu in FileMaker's Status Panel. It lists all of the layouts in a database.

Aligning Objects

Aligning fields, labels, and graphics in a database and report layout can be very tedious work. Fortunately, some programs have commands that make aligning objects easier. FileMaker Pro has an Alignment dialog box (Fig. 9-17) that works like most alignment commands in graphics programs.

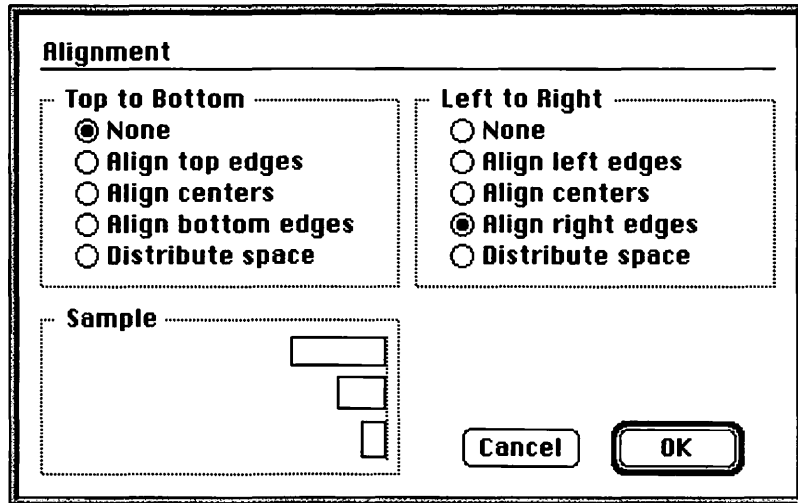


Figure 9-17: FileMaker Pro's Alignment dialog box.

You select the objects you want to align and then open the Alignment dialog box and select the radio buttons to align the objects. As you select radio buttons, the Sample area at the bottom of the dialog box shows you dynamically how the objects will be aligned. The Align command in the Arrange menu works in conjunction with the last settings in the dialog box. If, for instance, you align the right edges of the objects with the Align command... command, the Align command will align the right edges of objects until you change the settings in the dialog box.

I have found that I usually align objects by their right or left edges, so I created sequences for aligning objects by the different edges with a keystroke.

Before we begin: You will need to select the objects you want to align. If you are working in FileMaker Pro, you will need to switch to the layout view to select the objects.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Select Alignment... from FileMaker's Arrange menu.
3. Click the Align left edges radio button.
4. Press **Return** to close the dialog box and to save your changes.
5. Select Stop Recording... from the QuicKeys menu. When the Sequence editor opens, your macro should look like Figure 9-18.

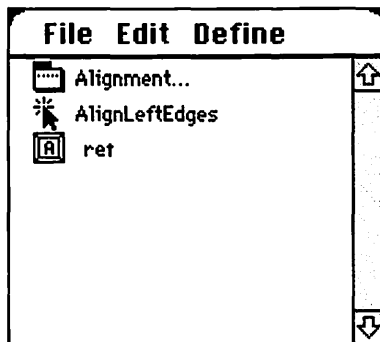


Figure 9-18: The Final AlignLeftEdges macro. As usual, I've named the click QuicKey to make it more readable.

6. Name your macro and assign a keystroke.
7. Click OK to close the Sequence editor and then click OK to close QuicKeys and save your changes.


Variations: You will find macros in the FileMaker Pro keyset that align objects by the right edges as well as by their top and bottom edges. You may want to make macros for any combination of alignment settings you use on a regular basis.

Automating Data Entry

In addition to making it easier and more automatic to create databases, QuicKeys can simplify data entry. For example, you can create macros for any information you find yourself entering regularly. If you routinely search for the same company's information in your database, you can create a macro to set up the search and enter enough information to locate the company's records.


Entering an Address



Whenever you enter the same information on a regular basis, you should consider making macros to enter that information for you. For instance, you might want to create a macro that enters the correct state abbreviation. If the database is used by a salesperson for order entry, macros that enter the state abbreviation would speed the operator's work and make it more accurate. Or, you might want to enter the addresses of your top twenty clients. In FileMaker Pro, you can create a pop-up menu that contains those twenty clients, but it would take longer to select a client's address from one of the twenty options than it would to enter it with a keystroke.

Before we begin: If you are creating a macro that automatically enters the address in separate fields of your database, you will probably want to create it within QuickKeys instead of recording it. You may have noticed that whenever you record something QuickKeys interprets as a text QuickKey, it adds all of your keystrokes to that text QuickKey until you press a -key combination.

Create the macro:

1. Open QuickKeys.
2. Select your database program from the pop-up menu.
3. Select Sequence from the Define menu.
4. When the Sequence editor opens, select Text... from the Define menu. When the Text editor opens, enter the name of the business. Click OK to close the Text editor.

If your database doesn't open a new record with the insertion point in the first field, you may need to enter a  or some other keystroke before you can enter the business' name.

5. Click the Literal button and press . This moves you to the address field.
6. Select Text... from the Define menu and enter the address of the business when the Text editor opens. After you have entered the address, click OK to close the Text editor.
7. Click the Literal button and press  to move to the city field.

8. Select Text... from the Define menu. Enter the name of the city. Click OK to close the Text editor.
9. Click the Literal button and press **Tab** to move to the state field.
10. Select Text... from the Define menu and enter state abbreviation. Click OK to close the Text editor. Your macro should look like Figure 9-19.

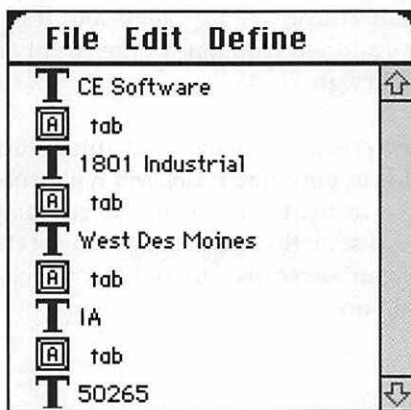


Figure 9-19: The final AutoAddress macro.

11. Name your macro and assign a keystroke. Then click OK to close the Sequence editor.
12. Click OK to close QuickKeys and save your changes.

Variations: You can expand this macro to do any of your data entry work for you. If you work in one program but keep your database open in the background, you can add steps to the macro to bring the database to the foreground, switch to the data entry mode, and then enter the address.

Preparing a Search

As you are working in a database, you may discover that you search for the same set of information or group of records on a regular basis. You may regularly search for the same clients' invoices in an order tracking database. If you do, you can create a macro that sets up the search for you. For example, the macro can invoke the Find command, tab to the field in which you are going to search, and then fill in the city, state, or name information for your search.

Before we begin: Open the database for which you would want to create an automated search. I have a simple FileMaker Pro rolodex database that I'll use for this example (Fig. 9-20). I'll be searching for all of the businesses in my database that are in San Francisco with 800 phone numbers.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. Press **⌘-F** to switch to FileMaker Pro's Find view.
3. Press **Tab** three times to move the insertion point to the City field. Type the name of a city.
4. Press **Tab** three more times to move to the Phone#1 field and enter 800 (Fig. 9-20).
5. Press **Return** to activate the search.

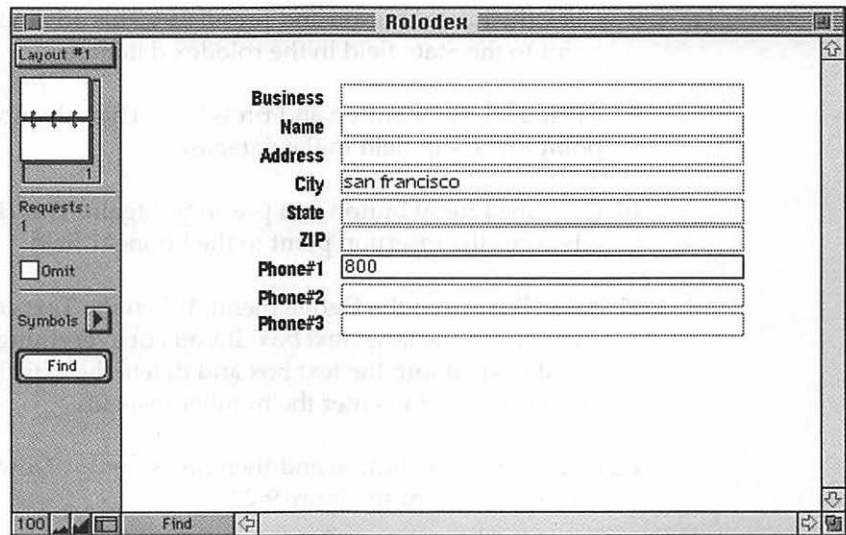


Figure 9-20: The automated search set up in the rolodex database.

6. When the search is complete, select Stop Recording... from the QuickKeys menu. The Sequence editor will open. Your sequence should look like Figure 9-21.

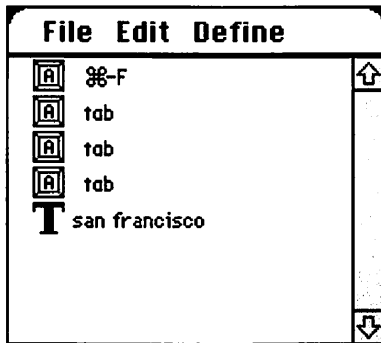


Figure 9-21: The unedited SetUpSearch macro.

7. Open the text QuickKey. You'll see that the three **Tab** presses, the 800, and the **Return** have been added in it. Select everything after the name of the city and press **Backspace** to remove it from the QuickKey. We will create literal and text QuickKeys to replace the information we are deleting. Doing this will make the QuickKey easier to change.
8. Click the Literal button and press **Tab**. This advances the insertion point to the State field in the rolodex database.
9. Click the Literal button and press **Tab**. This advances the insertion point to the Zip field in the database.
10. Click the Literal button and press **Tab** again. This final **Tab** advances the insertion point to the Phone#1 field.
11. Select Text from the Define menu. When the Text editor opens, enter 800 in the large text box. If you cut everything in step 7, you could paste it into the text box and delete the extra **Tab** presses, but it is simpler to enter the number instead.
12. Click the Literal button and then press **Return**. Now your macro should be similar to Figure 9-22.
13. Click OK to close the Sequence editor and then click OK to close QuickKeys and save your changes.

Variations: If your database is large and you know it will take several minutes before the search is complete, you could add a few additional steps to the end of the sequence so QuickKeys alerts you when the search is done. You would need to add a CursorWait QuickKey to check for the

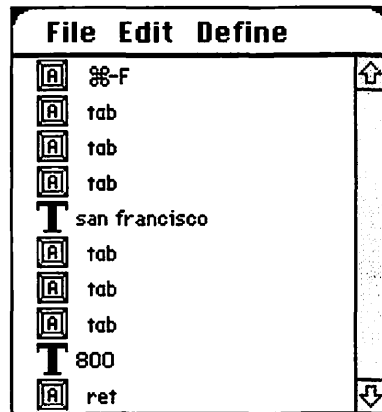


Figure 9-22: The completed SetUpSearch macro.

completion of the search. Then you would want to add a sound or a message QuickKey to get your attention.

To modify the extension SetUpSearch macro so it will pause until a search is over and then alert you, make a copy of the macro and rename it something descriptive. Then you will need to add the steps we just discussed.

1. Make certain that the insertion arrow is at the bottom of the sequence.
2. Select CursorWait from Extensions in the Define menu. When the CursorWait editor opens, select Is and Arrow from the pop-up menus and then click OK to close the extension.

NOTE: You could select *Other...* from the pop-up menu in the CursorWait extension and select the special cursors that FileMaker Pro uses during a search (the last two on the top row in Figure 9-23). However, FileMaker Pro alternates between these two cursors a number of times during a search, and it is much more reliable to wait for the Arrow cursor to reappear signaling the end of the search.

3. Select Sound from Extensions in the Define menu. When the Sound editor opens, select a sound from the pop-up menu. Click OK to close the editor.

Or, you might prefer to display a message instead of a playing sound. Select Message from Extensions in the Define menu. When

the Message editor opens, type your message in the Message to display text box (Fig. 9-24). Select the Show OK checkbox. This option adds an OK button to the Message dialog box so you can close it. Select the Continue automatically checkbox. You may want to enter a different number of seconds in the text box. This option causes the Message box to close after 10 seconds whether or not you click the OK button.

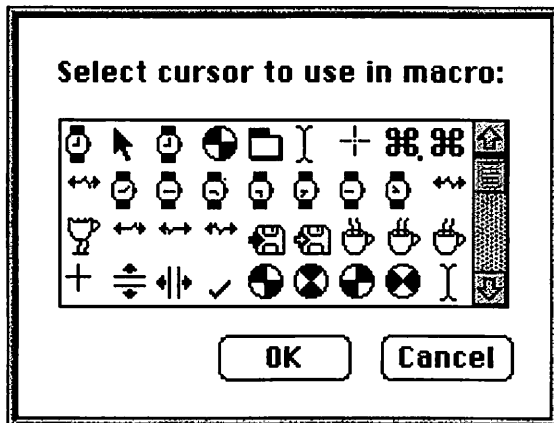


Figure 9-23: The Other... cursor dialog box in the CursorWait extension. In addition to the standard cursors, you can see the specialized cursors FileMaker Pro uses.

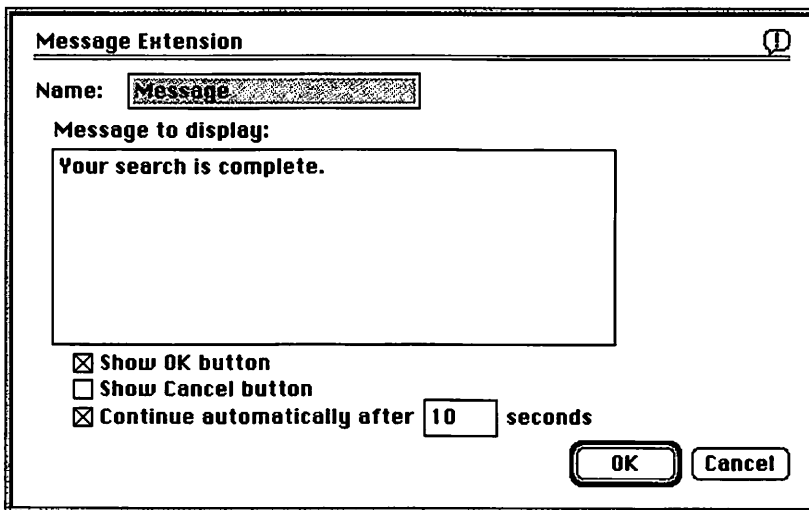


Figure 9-24: The Message extension settings for the SetUp&Alert macro.

- Now that you have entered all of your settings in the Message editor, click OK to close it. Your macro should look something like Figure 9-25.

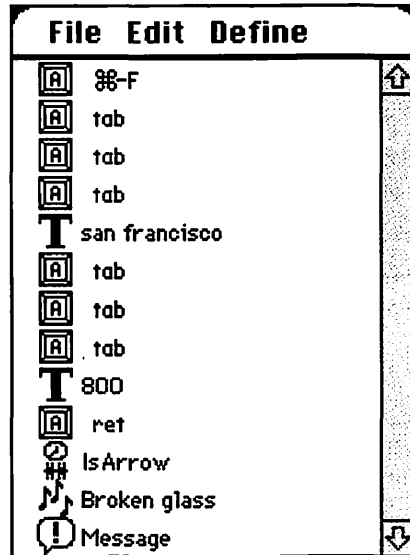


Figure 9-25: The final SetUp&Alert macro. Broken Glass is the sound I selected. I've also edited the name of the CursorWait QuickKey so I can tell what it is at a glance.

- Click OK to close the Sequence editor. Then click OK to close QuickKeys and save your changes.

If you have a database with all of your invoices in it, you can create a macro that gives you a quick count of the number of invoices for an area or region. If you check the Symbols pop-up menu in FileMaker Pro 2.0, you'll see that you can search for a range (Fig. 9-26). If you are using FileMaker Pro 1.0, you'll find a Range button in the Status Panel when you select the Find command.

To create a macro to search for a region by zip code, you would tab to the Zip field and enter the first zip code in the region. You would then enter the range symbol. You could select it from the pop-up menu, but your macro would fail whenever the Status Panel was hidden. After the range symbol, you would enter the last zip code in the region and then activate the search.

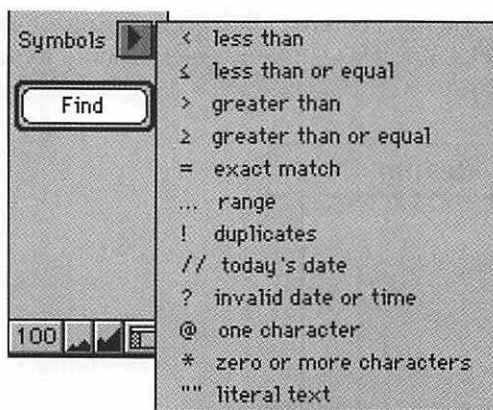


Figure 9-26: The Symbols pop-up menu in FileMaker Pro 2.0.

Automating Conversions

If you routinely export the records from your database to use in your word processor, you can create a macro that automatically converts your database for you. For instance, you may create a tab-delimited version of your database for use in a mail merge letter.

Before we begin: You should have the database from which you want to export the records open. We'll record the macro and then make some adjustments to it.

Create the macro:

1. Select Record a Sequence from the QuickKeys menu.
2. Select Export Records... from the Import/Export menu in FileMaker Pro's File menu. A dialog box opens where you can name your file and choose a format (Fig. 9-27). Let's leave the File Type pop-up menu set for Tab-Separated Text. Doing this allows any word processor to open the file.
3. Enter a name for the file and press **Return**.
4. FileMaker Pro's export dialog box opens now (Fig. 9-28). If you like, you can change the order of the fields you export or deselect the fields you don't want to export. Leave the settings as they are and press **Return**. FileMaker begins exporting the records to the new file.

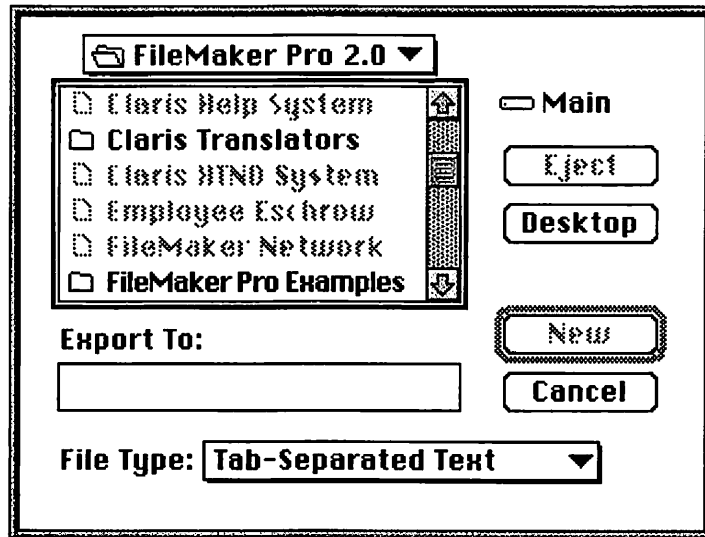


Figure 9-27: FileMaker Pro's Export dialog box.

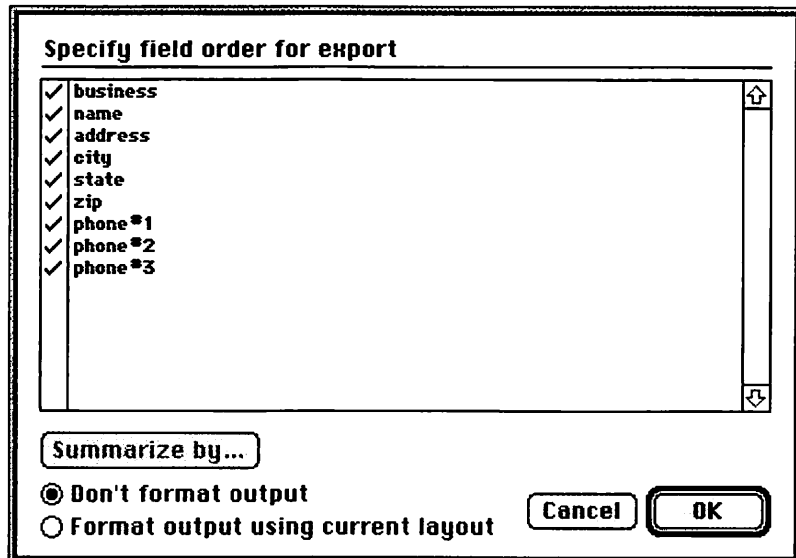


Figure 9-28: FileMaker Pro's export field order dialog box.

5. After FileMaker is through exporting the records, select Stop Recording... from the QuickKeys menu. When the Sequence editor opens, your macro should look like Figure 9-29. Now let's make a few changes to the macro so it waits for the dialog boxes to appear.

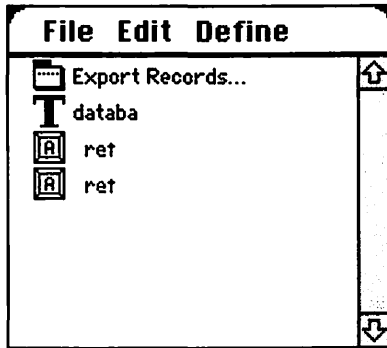


Figure 9-29: The unedited AutoExport macro.

6. Move the insertion arrow below the Export Records... menu QuickKey and select Wait... from the Extensions menu. When the Wait editor opens, select Dialog in front from the Wait until... pop-up menu. This pauses the macro until the export dialog opens. Click OK to close the Wait editor.
7. Move the insertion arrow below the name you typed for the exported records file.
8. Let's add a date-stamp to the end of the file name. Select one of the date options from the Date/Time... hierarchical menu. When the Date/Time editor opens, either change the settings or click OK to accept your settings and save the changes.
9. Select Location from the Extensions menu. When the Location editor opens, click the Select Location button. When the open dialog box appears, navigate to the folder where you want to save the file and click the Select button. You'll notice that the path to the file appears below the Set Standard File Location radio button. Click OK to close the Location editor.
10. Now move the insertion arrow between the two Return presses, and select Wait... from the Extensions menu. When the Wait editor opens, select Dialog in front from the Wait until... pop-up menu. This pauses the macro until the export field order dialog box appears. Click OK to close the Wait editor.
11. Move the insertion arrow below the last Return and select CursorWait from the Extensions menu. When the CursorWait editor opens, select Is and Arrow from the pop-up menus. This pauses the macro until all the records are exported and the cursor returns to an arrow. Click OK to close the CursorWait editor.

12. Now let's set a sound to alert us to the fact that the export is complete. Select Sound from the Extensions menu. When the Sound editor opens, select a sound from the pop-up menu. Click OK to close the Sound editor.

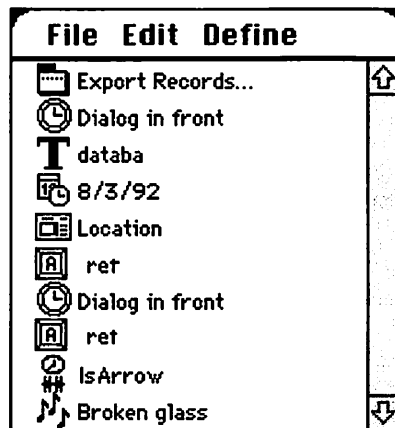


Figure 9-30: The final AutoExport macro.

13. Your macro should look similar to Figure 9-30. Name your macro and assign a keystroke. Click OK to close the Sequence editor. Click OK to close QuickKeys and save your changes.

Variations: An obvious variation of this macro would be to import the records from a tab-delimited document. If you frequently brought a text file containing the records from your company's mainframe over to your Macintosh, for example, and then imported those records into your Macintosh database, you could automate the import process in the same way you automated the export process in the macro above.

To create this macro, you would select Import Records... from the Import/Export... hierarchical menu in FileMaker Pro's File menu (Fig. 9-31). Because you must select the file to import in the FileMaker Pro's import dialog box, you will need to create a folder on your hard disk where you always place the file. You can then use Location to open the folder. If you don't place any other files in the folder, the file is automatically selected in the import dialog box, and you only have to press **Return** to open the file.

Once FileMaker Pro opens the file in the import dialog box, it displays an import field order dialog box (Fig. 9-32). It is a good idea to have the order of the fields in your mainframe records match the order of the

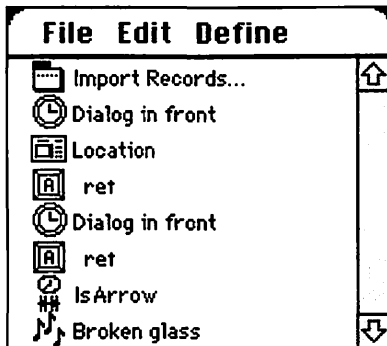


Figure 9-31: The AutoImport macro.

fields in your Macintosh database records. If you do, you can press **Return** to begin the import process whenever this dialog box appears.

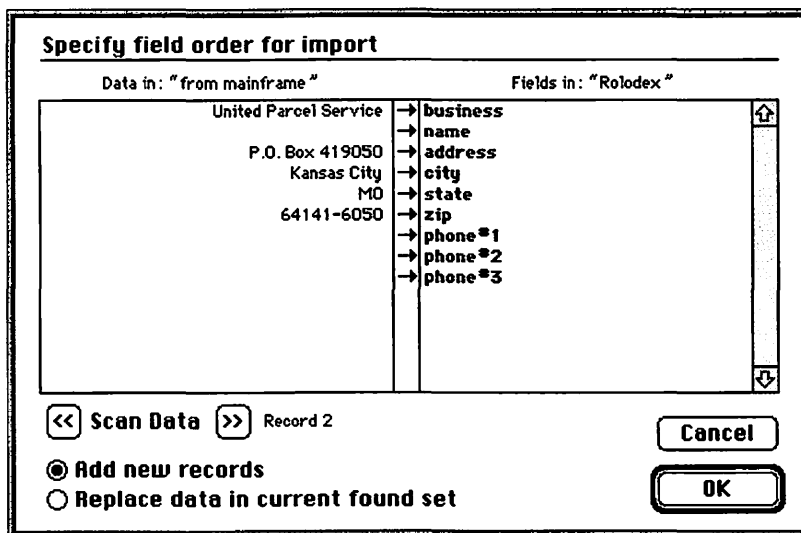


Figure 9-32: FileMaker Pro's import field order dialog box.

Now What?

Now that you have some ideas for automating your database, we'll move on to automating the Finder. In the next chapter, we will create macros that empty the trash, print files while you sleep, automate file sharing, and rebuild the desktop. You will also find macros for finding and backing up modified files and for quitting the Finder under System 7. I have also included keyboard shortcuts for the Finder's commands and tossed in a few additional tips for using QuickKeys. If you are ready, then let's dig in.

10

CHAPTER

Finder Tricks

Few users realize how much time they spend in the Macintosh Finder. If you run MultiFinder or System 7, you may find that you spend less time in the Finder transferring between applications, but you still use it for copying, moving, and deleting files and folders and, under System 7, sharing files. The Finder is the command center for such operations. Apple has sold millions of Macintosh computers because the Finder makes file maintenance tasks simple. Users don't have to know the arcane commands for copying files or changing directories, humble themselves by asking computer experts, or waste their time by looking up commands in a manual.

But for all of the simplicity the Finder brings to users, it lacks any means for automating the tasks it simplifies. There aren't any commands for finding all of your recently changed files and copying them to another disk or for printing a number of files at once. This chapter shows you how to customize and automate the Finder to perform these tasks and many more.

Automating the Finder

Speed isn't always about having the fastest Macintosh. You can make the Finder faster by running it on an '040-powered Macintosh, but the human element can make even the fastest Macintosh slower. You can always make the Finder faster by letting go of the mouse and automating your tasks. As you have probably noticed after running some of the macros from the earlier chapters of the book, your Macintosh can perform tasks much more quickly than you can. Even when it must perform more steps to accomplish a task than you have to, the Macintosh is usually faster. Sometimes it is even more accurate.

Keyboard Shortcuts

The Finder does have a number of keyboard shortcuts. You can create a new folder ($\text{⌘}+\text{N}$), eject a disk ($\text{⌘}+\text{E}$), get info about a file ($\text{⌘}+\text{I}$), and duplicate a file ($\text{⌘}+\text{D}$). But I have always found that it never had shortcuts for the commands I use the most (Restart, Shut Down, and Sharing...). Mouse-phobe that I am, I have macros for the Empty Trash, Restart, Shut Down, and Sharing... commands. In the past, I have added the shortcuts to the Finder using ResEdit, but—as I mentioned in Chapter 3—I seemed to be adding the shortcuts every time I had to reinstall the system software. A much better approach is to create Menu/DA QuickKeys instead of “hacking” keyboard shortcuts into the system software using ResEdit.

Before we begin: Since we want these QuickKeys to be Finder-specific, open QuickKeys and select Finder from the pop-up menu. Let’s create Menu/DA QuickKeys for the Restart, Shut Down, Empty Trash, and Sharing... commands. We’ll want to create these within the QuickKeys editor so we don’t accidentally invoke the commands for which we are making keyboard shortcuts. To create the shortcut for the Finder’s Sharing... command, you will need to have System 7 installed.

Create the macro:

1. With QuickKeys still open, select Menu/DA from the Define menu. A dialog box appears prompting you to select a menu item (Fig. 10-1).

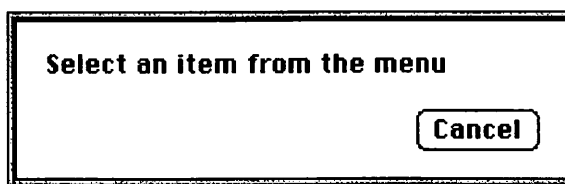


Figure 10-1: The Menu/DA QuickKey prompt dialog. If you change your mind about creating the QuickKey, click the Cancel button.

2. Select Restart from the Special menu. When the Menu editor opens, leave all of the settings the same (Fig. 10-2). You want QuickKeys to look for the command by name under the Special menu and you want it to match the command exactly. These settings prevent any unexpected surprises.

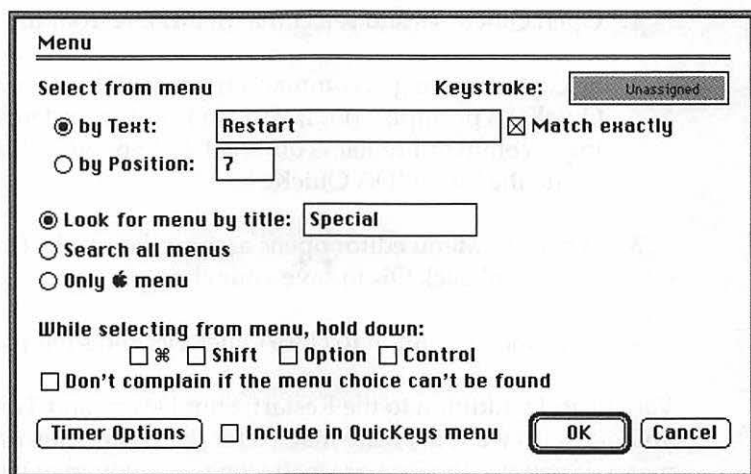



Figure 10-2: The Menu editor for the Restart QuickKey.

3. Assign a keystroke (I've chosen ⌘ -Option-R as a simple mnemonic) and click OK to close the editor and save your changes. Now let's create a Shut Down QuickKey.
4. While QuickKeys is still open, select Menu/DA from the Define menu and then select Shut Down from the Special menu.
5. When the Menu editor opens, leave all of the settings the same. Assign a keystroke (I've chosen ⌘ -Option-S as a simple mnemonic) and click OK to close the editor and save your changes.
6. Click the OK button to close QuickKeys and save your changes.

Now, let's create a QuickKey shortcut for the Empty Trash command.

1. Open QuickKeys and select Menu/DA from the Define menu.
2. Select the Empty Trash command when QuickKeys prompts you.
3. When the Menu/DA editor opens, assign a keystroke (I've chosen ⌘ -T).
4. Click the OK button to close the Menu editor and save your changes.

To create a QuickKey shortcut for the Sharing... command, you will need to have System 7 installed.

1. Open QuickKeys and select the Menu/DA from the Define menu.
2. Select the Sharing... command from the File menu when QuickKeys prompts you. If you are running System 7, the Sharing... command remains dimmed, but you will still be able to create the Menu/DA QuickKey.
3. When the Menu editor opens assign a keystroke (I've chosen -S) and click OK to save your changes.
4. Click the OK button to close QuickKeys and save your changes.

Variations: In addition to the Restart, Shut Down, and Trash commands, you may also want to create macros for the commands under the View menu. If you use the icon or small icon view, you may have noticed that it can take several seconds to open the Get Info window whenever you need to check on the size or modification date of a file. A much quicker means to find the file information is by selecting the by Size or by Date options from the View menu. With menu QuickKeys, you can create keyboard shortcuts that make changing views even quicker.

Of course, you could use the Restart and Shut Down specials instead of creating menu QuickKeys. The macro wouldn't really function the same way, since those specials restart or shut down your Macintosh from any program. You could also expand the function of these macros so that they close and save all of your files or empty a temporary trash folder before they shut down your Macintosh. We could do the same with the sharing macro. You could extend it by creating a macro that opens the Sharing window for the folder or disk and then sets default preferences. We'll create macros for both of these variations in this chapter.










Trash Old Files and Shut Down

Whenever you launched a program under System 6 and earlier, the Finder automatically emptied the Trash. Under System 7, the Trash began to behave like its real-life namesake. It no longer empties itself, but is emptied only when you select the Empty Trash command. Instead of using the Trash, you might prefer to make a folder that is a temporary receptacle for stuff to be trashed. You can then create a macro that automatically runs on a regular basis and empties the contents of the folder.

Before we begin: You could create the macro to open a folder on the desktop and drag its contents into the Trash, but it would be more likely

to fail or throw away the wrong folder or files. To make the macro reliable, you should have a program that can search for and delete files. You can let the program search for the folder that contains the files you want to trash and then delete them. For the purposes of this example, I'll be using DiskTop, but you can just as easily use any of the other file manipulation programs like DiskTools Plus, On Disk, or MasterFinder. You will also need to create a folder into which you drop your files and place some files you want to delete in it.

Create the macro:

1. Select Record Sequence from the QuickKeys menu.
2. If you are using a later version of DiskTop, select DiskTop Find from the Apple menu. DiskTop comes configured with the --F shortcut assigned to the DiskTop Find command, and you could incorporate that shortcut into the sequence using an alias QuickKey.
3. When the DiskTop Find dialog appears, enter the name of your temporary folder. If you have more than one hard drive mounted, you might want to select the drive name in the Select Drive(s) to Search list. This shortens the amount of time it takes to search for the folder.
4. Press  to begin the search. While DiskTop is searching for the folder, you may want to time how long it takes.
5. When DiskTop finds the folder, press -R to add the folder to DiskTop's Retain list.
6. Press the  to select the folder in the Retain window.
7. Press  to open the folder.
8. Press -A to select all of the files in the folder.
9. Press --D to delete all of the files in the folder.
10. Select Stop Recording from the QuickKeys menu. When the Sequence editor opens, we can make a few changes to the macro.
11. Move the insertion arrow between the menu and the text QuickKeys at the top of the sequence and select WindowAction from the Extensions menu.

12. When the WindowAction extension opens, click the Check Window Name checkbox to deselect it. Click the Check Window Type checkbox to select it. Click the Wait... radio button to select it (Fig. 10-3).

This causes QuickKeys to wait until the DiskTop Find dialog appears. Otherwise, the macro continues to play while the dialog box is opening and will never work properly.

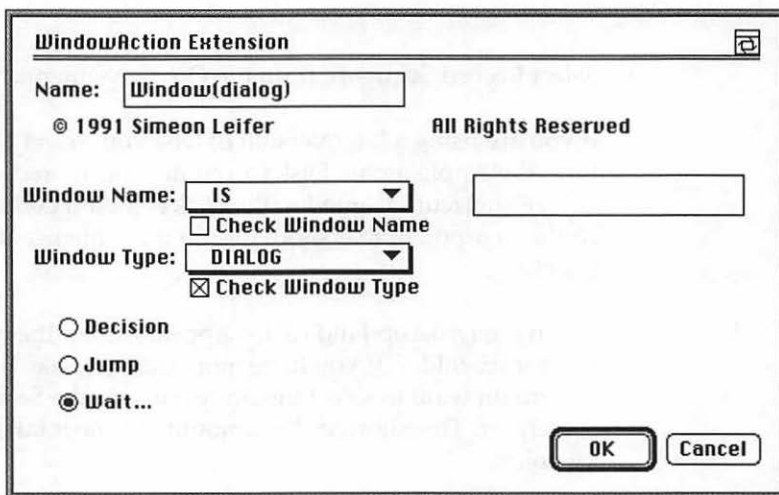


Figure 10-3: The WindowAction settings for the Empty&ShutDown macro.

13. Now move the insertion arrow between the `Return` and the `⌘-R`.
14. Select Pause from the Define menu and enter the number of seconds that it took DiskTop to find the temporary trash folder. This step also ensures that the macro pauses long enough to let your Macintosh work. Your completed macro should look similar to the one in Figure 10-4.

TIP: Because DiskTop's search times may vary, be sure to give DiskTop enough time to find your folder. You may want to add a pause and wait for user QuickKey here instead of trying to guess the amount of time it will take.

15. Click OK to close QuickKeys and save your changes

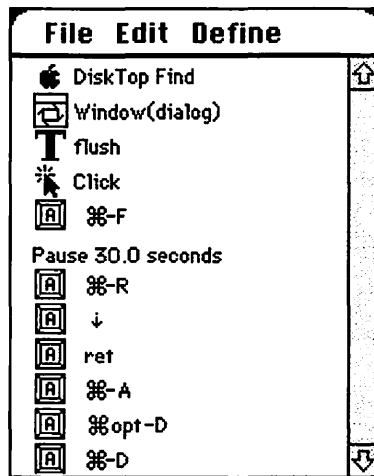


Figure 10-4: The completed Empty&ShutDown macro.

Variations: If you wanted this macro to shut down your Macintosh, you could add a shut down special QuicKey to the end of the macro. For variation, you might modify the macro so that it only looks for files of a certain type and deletes those. For example, you could create a macro that deletes all of the files with a .hqx, .cpt, .sea, or .sit extension.

Empty the Trash

If you are running System 7, you can empty your trash without returning to the Finder. Using QuicKeys and IAC, you can send an Empty Trash Apple Event to the Finder.

Before we begin: You must have System 7, CEIAC, and the Apple Events extension installed. If you are running System 7, CEIAC and the Apple Events extension are automatically installed when you install QuicKeys. Select Universal from the pop-up menu so the macro is available from any program.

Create the macro:

1. Open QuicKeys.
2. Select Apple Events from the Extensions menu.
3. When the Apple Events editor opens, select Finder from the Send Event to pop-up menu (Fig. 10-5).

You'll notice that all of the programs you are currently running are listed at the top of the pop-up menu. The Remote Application... command opens a dialog box that allows you to send an Apple Event to a program running on another Macintosh. You are only able to send Apple Events to remote programs if the other Macintosh is running System 7 and has Program Linking turned on in the Sharing Setup control panel. The Other... command displays an open dialog box that allows you to select a program from disk that isn't currently running.

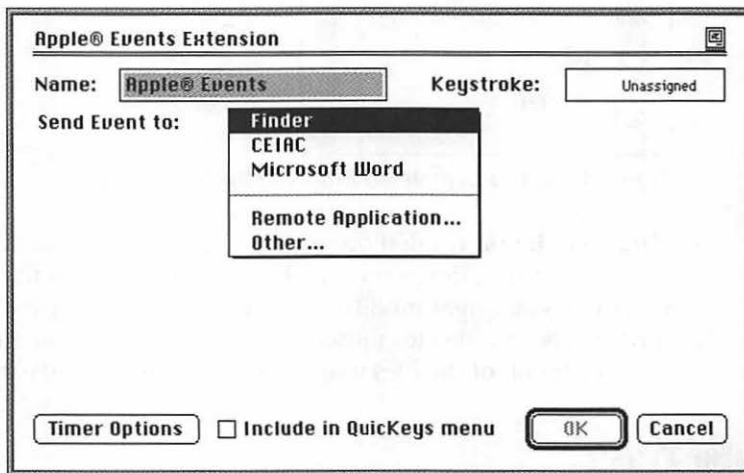


Figure 10-5: Select Finder from the Send Event to pop-up menu.

4. Select Custom Event from the Event pop-up menu (Fig. 10-6). The Core events the program supports are listed at the top of the pop-up menu. In this case, the Finder supports the Print Documents and the Quit Application events. If the program has an aete resource, you can select Lookup from Target..., and QuickKeys will display a list of the custom events that the program supports.

We are creating a custom event because the Finder, while supporting the Core Apple Events, does not have an aete resource. This means we can use the Lookup feature of the Apple Event extension and must enter the event's Class and ID. The Class and ID text boxes appear at the bottom of the Apple Event editor.

5. Type "FNDR" in the Class text box. This specifies the class of the Apple Event you are going to send as Finder.

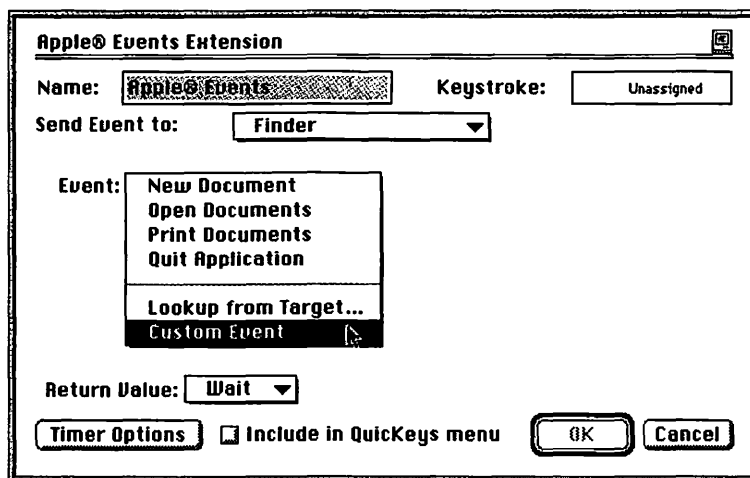


Figure 10-6: Select Custom Event from the Event pop-up menu.

6. Type “empt” in the ID text box. This specifies the ID of the Apple Event you are going to send as empty, which tells the finder to empty the trash (Fig. 10-7).

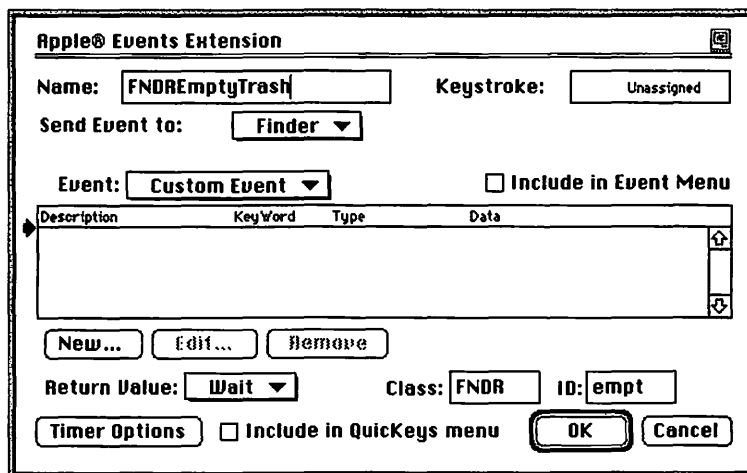


Figure 10-7: The Apple Events extension configured for the EmptyTrash macro.

7. Name your macro and assign a keystroke.
8. Click OK to close the Apple Event editor.
9. Click OK to close QuickKeys and save your changes.

Variations: If you are running System 7 and want to empty your trash on a regular basis but don't want to use one of the public domain programs or a commercial program like TrashMaster (which runs in the background and empties your trash), you can make this a timed QuickKey. To do this, follow these steps:

1. Click beneath the clock and next to the QuickKey in the QuickKeys editor (Fig. 10-8). Or you can open the QuickKey and click the Timer Options button at the bottom of the Apple Events editor.

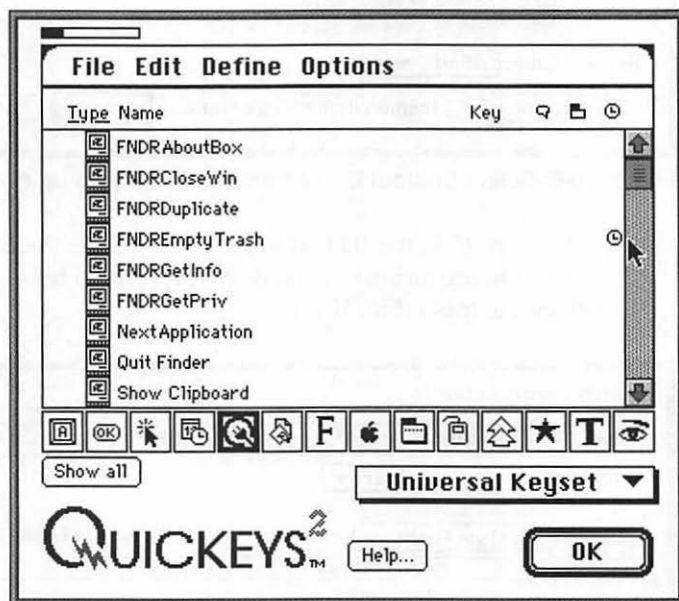


Figure 10-8: Click next to the QuickKey and beneath the clock icon to set the time you want the QuickKey to run.

2. When the Timer options dialog opens (Fig. 10-9), click the At checkbox.
3. Then click the Repeating every checkbox and enter the amount of time you want QuickKeys to wait before it plays the QuickKey.
4. Click OK to close the Timer Options dialog.
5. Click OK to close QuickKeys and save your changes.

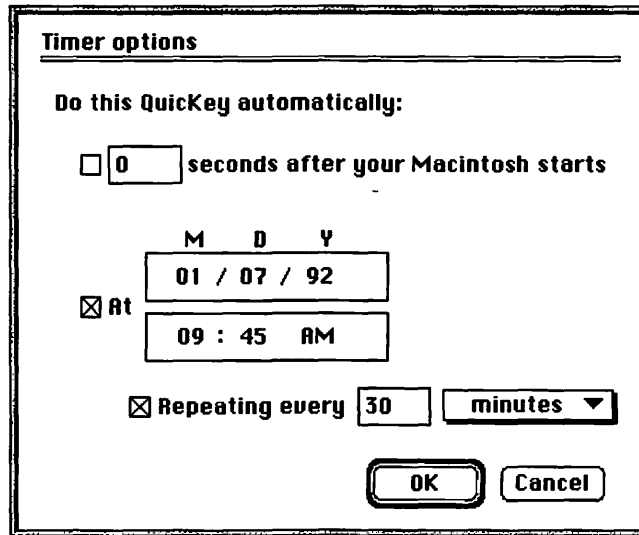


Figure 10-9: The Timer options dialog lets you set the starting time and frequency for a QuickKey to run.

Even though this QuickKey isn't as fancy as some of the public domain and commercial offerings, it does have several advantages. It empties the Trash on a regular basis, you don't have to add another extension or INIT to your System folder, and it doesn't cost you anything.

Printing Files While You Sleep

Nothing is more infuriating than having to wait on a printer. While spoolers can free you from waiting until the print job is completed, only putting a printer on every desk frees you from waiting on a co-worker's print job. You can avoid the printer bottleneck by printing your files during non-prime time, that is, after everyone has gone home. You can create a macro to do this using the Print command in the Finder Events extension.

Before we begin: To take advantage of Finder Events, you'll need to be running System 7 and have CEIAC and the Finder Events extension installed. Open QuickKeys and select Universal from the pop-up menu so you can play the macro from within any program.

Create the macro:

1. Open QuickKeys if you have closed it and select Sequence from the Define menu.

2. Select Finder Events from the Extension menu. When the Finder Events editor opens, select Print from the pop-up menu (Fig. 10-10).

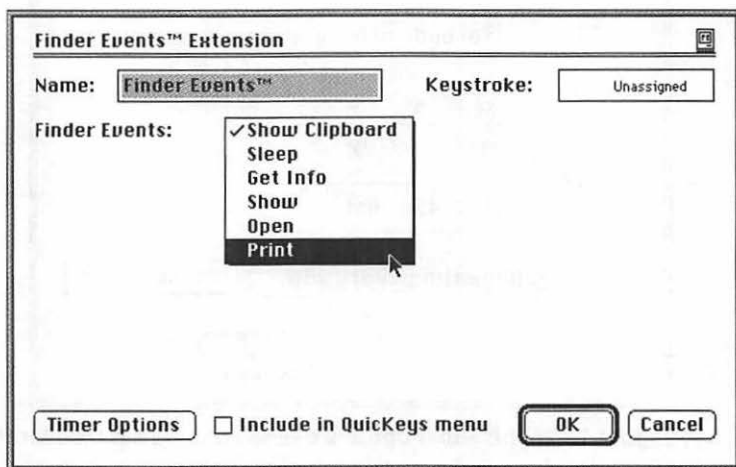


Figure 10-10: Selecting the Print command from the Finder Events extension.

3. A dialog box opens where you can specify the documents you want to print. Select the name of the documents in the top scrolling list and click the Add button (or you can double-click the document's name).

When you have selected all of the documents, click the Done button to close the dialog and save your changes. The Finder Events extension now contains a list of the documents you selected (Fig. 10-11). You do not have to worry about selecting the program that created the document because the Finder will take care of that for you. However, you can only select documents created by the same program. If you select documents created by different programs, the next step in the macro will not click the Print button. Click OK to close the extension and save your changes.

4. Now select WindowAction from the Extensions menu. When the extension opens, click the Check Window Name checkbox to deselect it. Then click the Check Window Type checkbox and click the Wait... radio button. When you play the macro, these settings will pause the macro until the Print dialog box opens. Click OK to close the extension and save your changes.

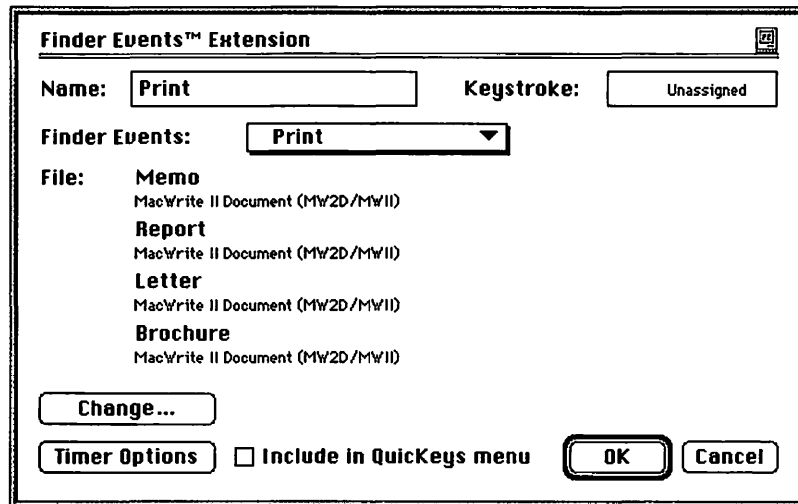


Figure 10-11: The Finder Event extension after you have selected your documents.

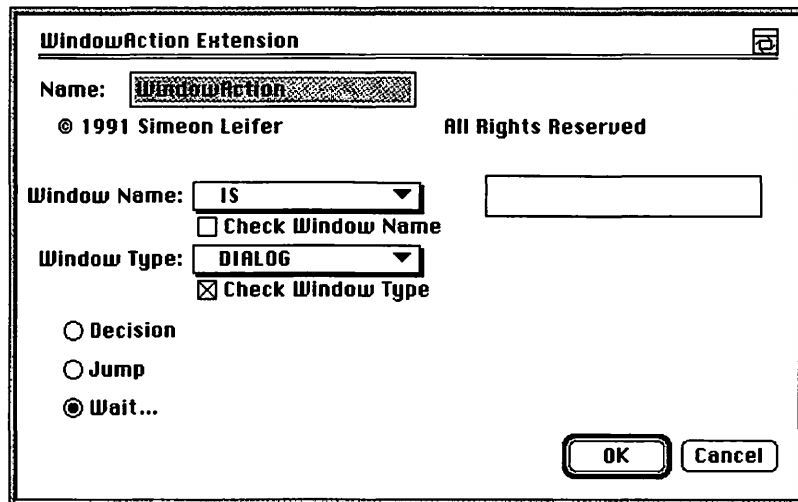


Figure 10-12: The settings for the WindowAction extension.

5. Click the Literal button in the Sequence editor. When you are prompted to press a keystroke, press **Return**. The **Return** clicks the Print button whenever the Print dialog appears.
6. To set a time for the macro to run and print the documents, click Timer Options button. When the Timer options dialog opens, you can set the time as you did in the EmptyTrash macro above (Fig. 10-9).

7. Your completed macro should look like Figure 10-13. Assign your macro a name and click OK to close the Sequence editor. If you plan to use this macro in the early morning hours, you do not need to assign a keystroke.
8. Click OK to close QuickKeys and save your changes.

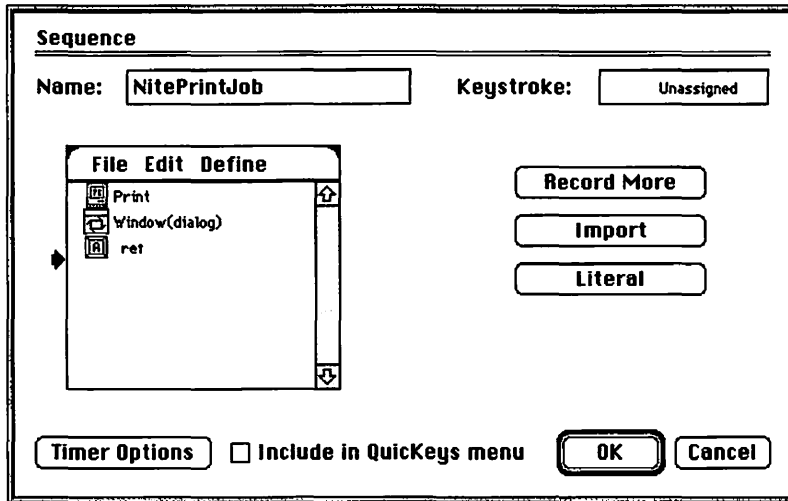


Figure 10-13: The completed NitePrintJob macro.

Use and variations: Sometimes a program does not need the **Return**; it will automatically start printing the document if you have issued the Print command from the Finder (which is really all the Print Finder Event is). If your program starts printing without the **Return**, you can reduce the number of steps in your macro. While MacWrite II requires the **Return**, your program may not—a little experimentation is in order.

If you don't want to leave your Macintosh on all night and you own a PowerKey from Sophisticated Circuits, you could set the PowerKey to start your Mac and then play the NitePrintJob macro. To do this, you select QuickKeys... from the Event Type pop-up menu. A scrolling list of your QuickKeys opens. When you select your macro, it appears in a box below the pop-up menu (Fig. 10-14).

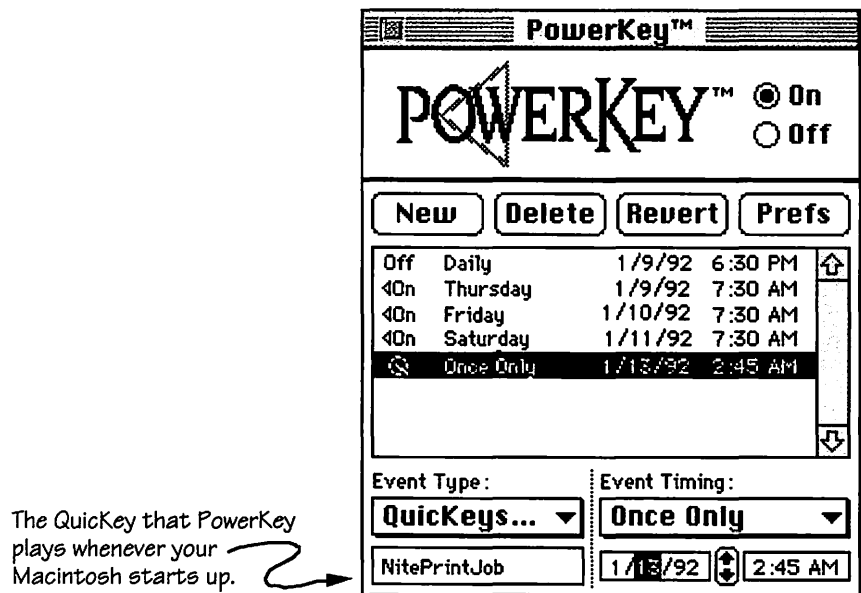


Figure 10-14: The PowerKey control panel with the NitePrintJob macro set to run whenever your Macintosh starts up.

Automating File Sharing

Peer-to-peer file sharing is one of the great additions in System 7; it allows you to share your hard disk and files without having a Macintosh dedicated to running AppleShare. However, leaving file sharing on all of the time can seriously degrade the performance of your Macintosh. The Start and Stop File Sharing QuicKeys available in the System 7 Specials extension make it more convenient to use file sharing. You can run your Macintosh with file sharing turned off and turn it on with a keystroke whenever you want—if your office mate needs a file from your hard disk, for instance. You can press the keystroke you've assigned to the Start File Sharing special. While this is a quick way to turn on peer-to-peer file sharing, you don't always know when file sharing has actually started up.

You can expand the functionality of the Start File Sharing QuicKey by combining it with the Finder's Sharing... command. Instead of just having **⌘-S** open the sharing dialog box, you can create a sequence that starts up sharing, opens the Sharing Setup control panel so you can monitor sharing startup, and then switches back to the program you were working in.

Before we begin: You should be running System 7 and have CEIAC, System 7 Specials, and Finder Events extensions installed. You should create users and groups following the instructions in your System 7 manuals. Also, you should enter your owner name, password, and Macintosh name in the Sharing Setup control panel (Fig. 10-15). If you haven't done this, warning dialog boxes will appear and cause your macro to fail. If that isn't a good enough reason to set it up in advance, let me remind you that you really don't want to turn on file sharing and expose your Macintosh to the world without including a password.

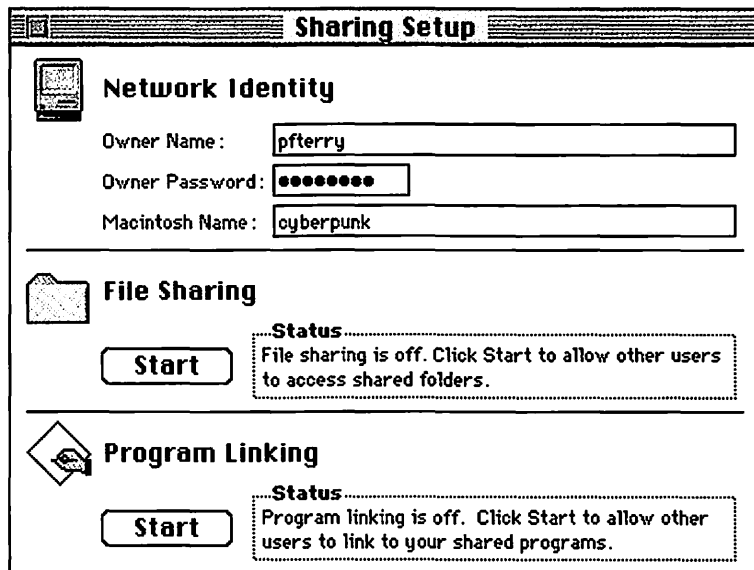


Figure 10-15: The Sharing Setup control panel in System 7.

You'll need to open QuickKeys and select Universal from the pop-up menu.

Create the macro:

1. Open QuickKeys if you have closed it and select Sequence from the Define menu.
2. When the Sequence editor opens, select System 7 Specials from the Extensions menu.
3. When the System 7 Special extension opens, select Start File Sharing from the pop-up menu (Fig. 10-16).

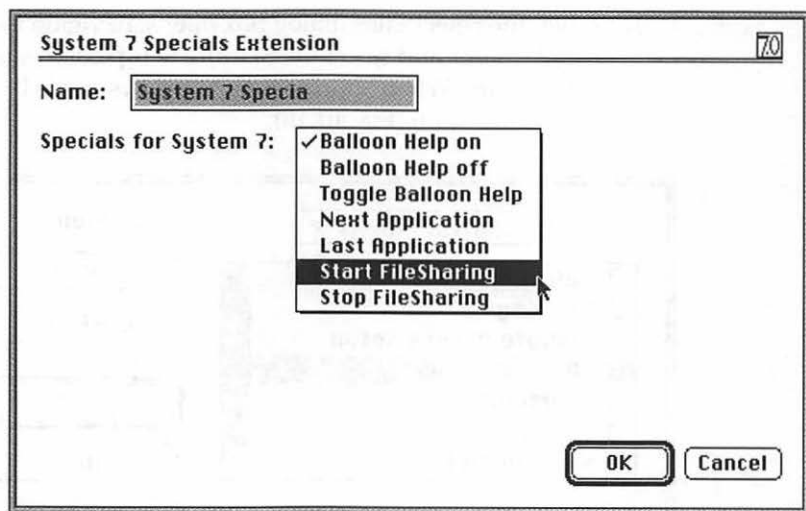


Figure 10-16: Selecting the Start File Sharing command in the System 7 Specials extension.

4. Click OK to close the System 7 Specials extension and save your changes.
5. Select Finder Events from the Extensions menu.
6. When the Finder Events editor opens, select Open from the pop-up menu (Fig. 10-17).

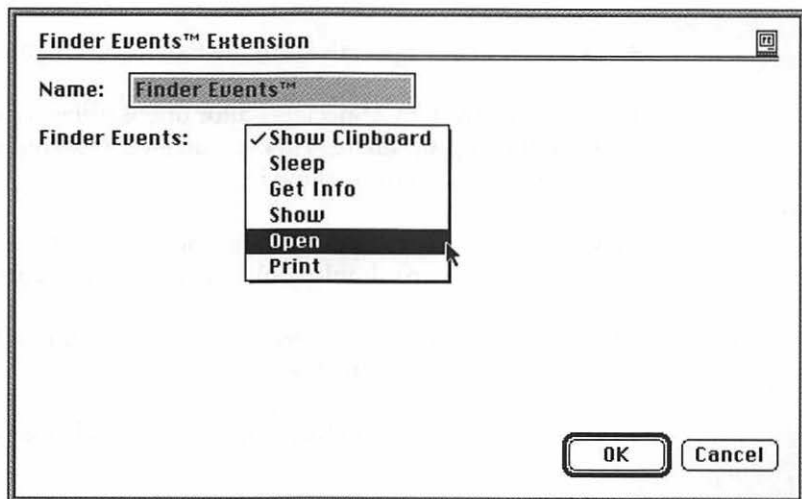


Figure 10-17: Selecting the Open command in the Finder Events extension.

7. When the Select Files dialog box opens, navigate to your Control Panels folder and select the Sharing Setup control panel. Click the Add button. When Sharing Setup appears in the bottom list, click the Done button (Fig. 10-18).

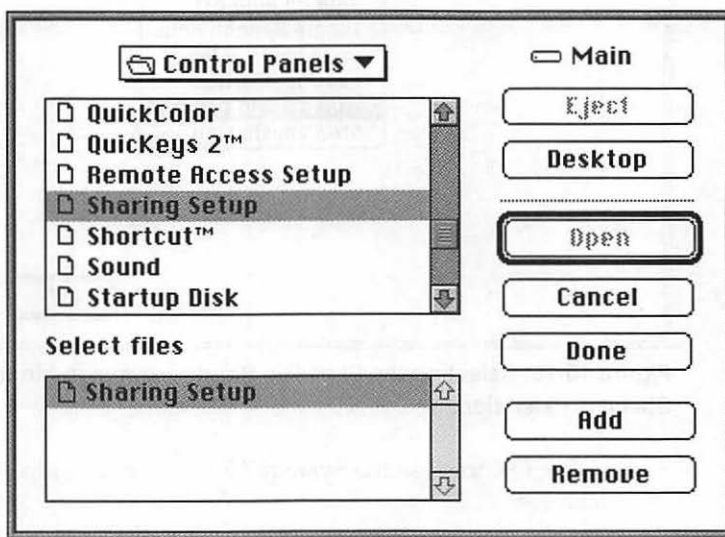


Figure 10-18: The Select Files dialog box for the Finder Events extension.

8. Close the Finder Events extension to save your changes. When the macro runs, this Finder Event will open Sharing Setup so you can monitor file sharing progress while it is starting up.
9. Select System 7 Specials from the Extensions menu again.
10. When the System 7 Specials editor opens, select Last Application from the pop-up menu. This special switches you back to the last program you were working in.
11. Close the System 7 Specials editor to save your changes. The completed macro should look like the one in Figure 10-19.
12. Name your macro and assign it a keystroke. Close the Sequence editor to save your changes.
13. Click OK to close QuickKeys and save your changes.

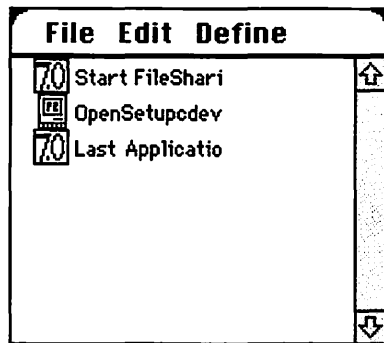


Figure 10-19: The completed MonitorStartup macro.

Use and variations: When you play this macro, you can switch between the program you are working in and the Finder until you see that sharing has started up. If you have a larger monitor, you could add a click QuickKey to the end of the macro that resizes the front window so you can see the File Sharing control panel. This would allow you to monitor the startup progress of file sharing without switching between the program you are working in and the Finder. Of course, you could let QuickKeys not only start up file sharing, but also enter default access privileges. Our next macro automates this entire process.

Setting File Sharing Privileges Automatically

As we mentioned above, you probably don't turn on file sharing until you need to share files. This improves the performance of your Macintosh and conserves memory. We can speed up the sharing process with a macro that is responsive to the current state of file sharing on your Macintosh. In other words, the macro recognizes when you do not have file sharing turned on and branches to start it up. We can also make the macro recognize when you do have file sharing turned on and branch to open the sharing window for the folder or disk you have selected.

Before we begin: As with the sharing macro above, this macro requires you to be running System 7 or later, to have entered the information in the Sharing Setup control panel, and to already have users and groups created.

If you have sharing turned off, you may want to turn it on now. Since this is a branching macro, we'll create it in three stages. First we'll create the portion of the macro that checks to see whether or not sharing is turned on and then branches. Then we'll create the two different

branches: one to check if file sharing is on and turn it on if it isn't and the other to set access privileges. Now open QuickKeys, and select Finder from the pop-up menu to save the macro in the Finder keyset.

Create the macro:

1. If you still have QuickKeys open, select Sequence from the Define menu.
2. When the Sequence editor opens, click the Import button. A scrolling list of QuickKeys appears on the right side of the Sequence editor.
3. Scroll until you see the Sharing Menu QuickKey you created above.
4. Select the Sharing Menu QuickKey and click the Copy button to import it into your sequence (Fig. 10-20).

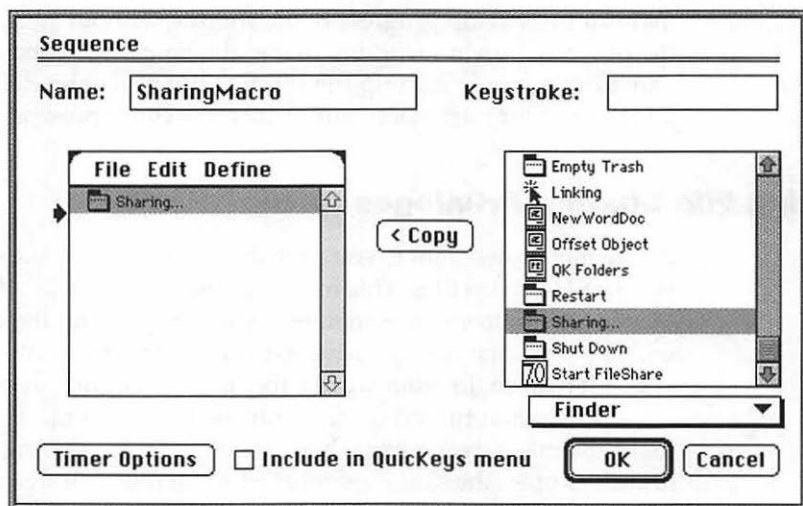


Figure 10-20: Importing the Sharing QuickKey you created in the previous example.

5. Select WindowAction from the Extensions menu. This portion of the macro recognizes the current file sharing conditions on your Macintosh and branches to play the appropriate QuickKey.
6. When the WindowAction extension opens, click the Check Window Name checkbox to deselect it then click the Check Window Type checkbox to select it.

The reason for these settings is that we don't want QuicKeys to look for the name of a window. Every access privilege window will have a different title depending on the name of the folder or disk you select to share. What we want QuicKeys to do is distinguish between an access privilege window and the dialog box that tells us file sharing has not been turned on (Fig. 10-23). To this end, you'll need to make sure that Not Dialog is selected in the pop-up menu (Fig. 10-21).

7. Click the Decision radio button. This tells WindowAction that it is going to make a decision based on whether a window or a dialog box appears. If a window appears, WindowAction plays the pass QuicKey. If a dialog box appears, WindowAction plays the fail QuicKey. In other words, WindowAction is watching for the access privileges window.
8. Now enter the names of the macros to which you want the SharingMacro to branch. For the purposes of this example, enter "SetPrivileges" in the Pass QuicKey text box and "StartSharing" in the Fail QuicKey text box.

The Set Privileges macro sets default privileges in the access privileges window. The StartSharing macro starts file sharing and opens the Sharing Setup control panel.

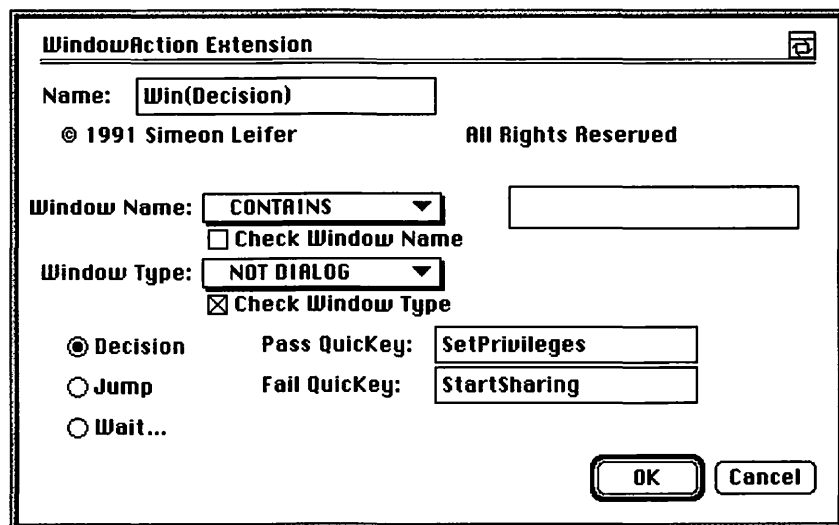


Figure 10-21: The WindowAction extension settings for the SharingMacro.

9. Once you have configured the WindowAction extension, click OK to save your changes. Your completed SharingMacro should look like Figure 10-22.

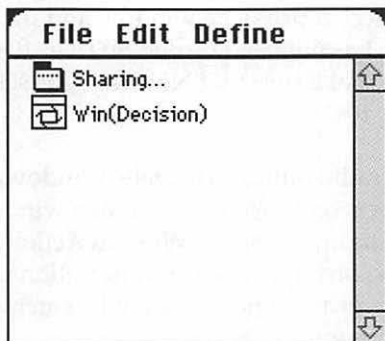


Figure 10-22: The final SharingMacro.

Now that we have created the branching portion of the macro, we need to create the two macros that the SharingMacro calls. First, we'll create the Set Privileges macro which is played if you have sharing turned on and the access privileges windows opened.

1. Select the folder or disk that you want to share.
2. Select Record Sequence from the QuicKeys menu.
3. Select Sharing from the File menu or press the keystroke you assigned above. The access privileges window for the folder appears (Fig. 10-23).
4. Click the Share this item checkbox. The Owner and User/Group pop-up menus and the other checkboxes are enabled.
5. Click the Everyone Make Changes checkbox to deselect it. This ensures that whoever connects to your Macintosh and opens the folder won't be able to make changes to the files and folders in it.
6. Click the Make all... checkbox. This sets the access privileges for every folder within the folder that you are sharing.
7. Click the Close box for the sharing window. A dialog box appears that asks if you want to save the changes made to access privileges (Fig. 10-24).

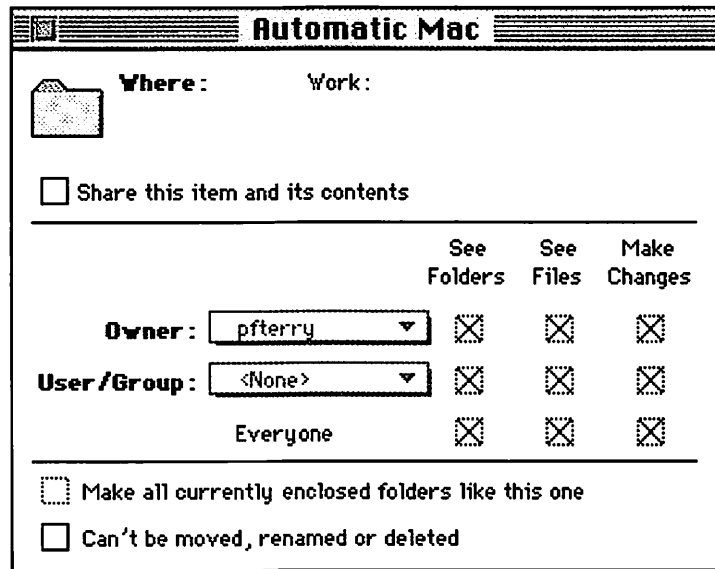


Figure 10-23: The access privileges window for the Automatic Mac folder.

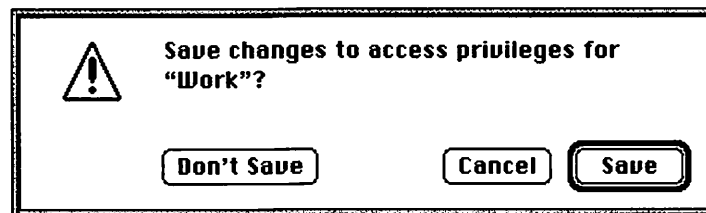


Figure 10-24: The save changes dialog box.

8. Press **Return** to save your changes. Another dialog box appears asking if you want to change the privileges for all of the folders (Fig. 10-25).

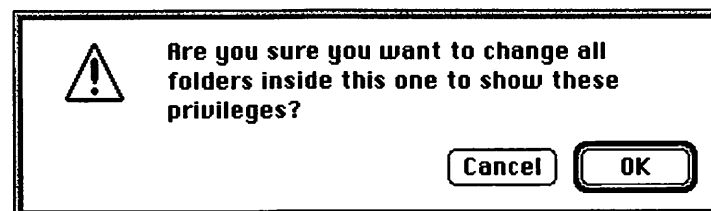


Figure 10-25: The change all folders dialog box.

9. Press **Return** to save the changes for all of the folders.

10. Select Stop Recording... from the QuickKeys menu.
11. When the Sequence editor opens, add one-second pauses between the click QuickKeys. This makes the macro play more reliably. Add the pauses by moving the insertion arrow between the clicks and selecting Pause from the Define menu.
12. Name the macro SetPrivileges. Make certain that the spelling matches what you entered in the Pass QuickKey text box of the WindowAction extension. If it is wrong, this macro will not be played. You do not need to assign a keystroke since the WindowAction plays the macro for you.
13. Click OK to close the Sequence editor and save your changes. The completed macro should look like Figure 10-26.

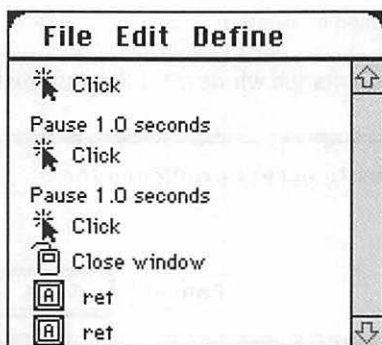


Figure 10-26: The completed SetPrivileges macro.

Now we need to create the StartSharing macro. This macro is played when you don't have file sharing turned on and the Finder gives you the option of opening the Sharing Setup control panel (Fig. 10-27).

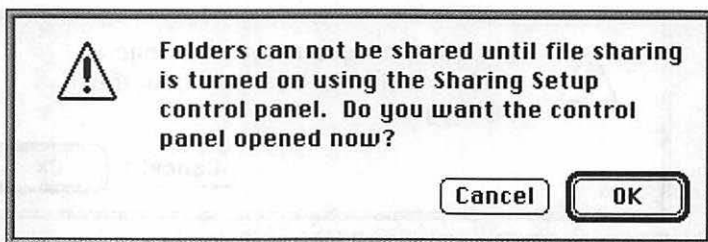


Figure 10-27: If you do not have file sharing turned on, the Finder asks you if you would like to open the Sharing Setup control panel.

1. Turn off file sharing if it is still on.
2. Open QuickKeys and select Sequence from the Define menu.
3. When the Sequence editor opens, select System 7 Specials from the Extensions menu.
4. Select Start File Sharing from the pop-up menu and then click OK to close the editor and save your changes. If the dialog box appears, the SharingMacro branches to this StartSharing macro and starts up file sharing. We could click the OK button and then click the Start button in the Sharing Setup window when it opens, but this method takes one less step and is more reliable.
5. Click the Literal button and press `Return` when you are prompted. Click OK to close the editor and save your changes. When the macro plays, this QuickKey clicks the OK button in the dialog box. We can skip this step since the System 7 special above turns on file sharing, but the Sharing Setup dialog box tells you when sharing is on so you don't have to keep checking on its progress.
6. Your completed macro should look like the one in Figure 10-28.

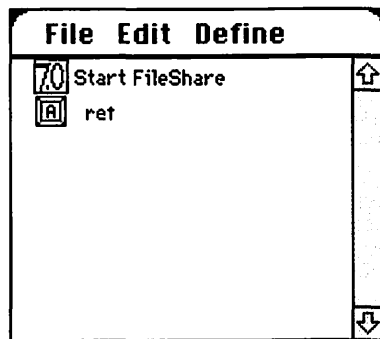


Figure 10-28: The completed SharingSetup macro.

7. Click OK to close QuickKeys and save your changes.

Use and Variations: If file sharing has to start up when you activate the SharingMacro, you will need to press the keystroke you assigned to the macro a second time. Doing so activates the SetPrivileges macro after file sharing has started.

Rebuilding the Desktop





Rebuilding the invisible Desktop file is an exercise that every Macintosh owner should practice on a semi-regular basis.* It compresses the file and improves the Finder's performance. For example, you may notice that it takes the Finder longer to display the disk and file icons when you haven't rebuilt the desktop file. You may also notice that the Finder displays the wrong icons or generic icons for a program and its documents when the desktop needs to be rebuilt.

Unfortunately, rebuilding the Desktop can take several minutes, longer if you have large drives or partitions mounted. I have two large drives split into four partitions, so the Finder's "Are you sure you want to rebuild the desktop file" dialog box appears four times after some very long pauses while the Finder compresses the Desktop file. I often begin rebuilding the Desktop and wander off to take care of other business, but I forget to come back and click the OK button for each partition. Since I really don't like to wait on my Macintosh, I created a macro that runs the Macintosh on autopilot and rebuilds the Desktop without my baby-sitting.

Before we begin: If you haven't already done so, open QuickKeys and select Finder from the pop-up menu.

Create the macro:

1. Select Sequence from the Define menu. When the Sequence editor opens, select Repeat from the Extensions menu.
2. In the Repeat extension, select the Begin Repeat button. This starts the repeat process for the sequence. You can now enter the number of times you want the sequence to repeat in the Repeat items text box. You can enter the number of drives or partitioned volumes you have attached. Since the Finder will ask you if you want to rebuild the desktop for each drive, you want the sequence to click the OK button in each dialog box. Another possibility is to enter a number but have the Repeat extension ask you how many times to play the sequence. If you often have several different drives or partitions mounted, this gives you a chance to enter the exact number of times for the sequence to repeat. Click OK to save your changes.

*You can rebuild the Desktop file by holding down - after restarting your Macintosh and before the Finder desktop appears. If you are not running MultiFinder or System 7, you can rebuild the desktop file by holding down - after you quit a program, but before the Finder appears.

3. Select WindowAction from the Extensions menu in the Sequence editor. Deselect the Check Window Name checkbox. Select the Check Window Type checkbox and make sure Dialog is selected from the pop-up menu. Then click the Wait... radio button. These settings cause QuickKeys to wait until a dialog box appears. Click the OK button to save your settings and close the extension.

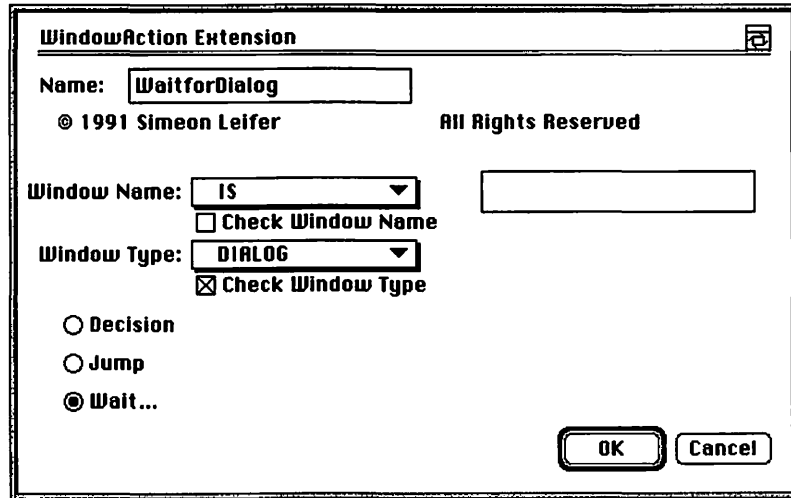


Figure 10-29: Setting the WindowAction extension to wait for the rebuild desktop dialog box.

4. Click the Literal button in the Sequence editor and press `Return`. The WindowAction extension in the previous step causes QuickKeys to wait until the dialog box appears that asks you if you want to rebuild the desktop. This `Return` literal selects the OK button in that dialog box.
5. Now that you have all of the elements in the macro that wait for the dialog box to appear and select the OK button, you are ready to end the repeat. Select Repeat from the Extensions menu. When the extension appears, select the End Repeat radio button.
6. Your Rebuild Desktop sequence is complete. You can assign a keystroke and save it. Of course, you don't have to assign a keystroke. I haven't assigned a keystroke to my RebuildDesktop macro. Instead, I use the QuickReference Card to activate it.
7. Click OK to close QuickKeys and save your changes.

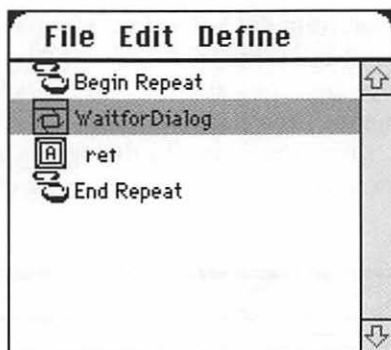


Figure 10-30: The final RebuildDesktop macro.

Use and variation: Because you must initiate rebuilding the desktop by holding down the **Option** and **⌘** keys whenever you turn on your Macintosh, you have to jump start this macro. Once you bring up the "Are you sure you want to rebuild the desktop file" dialog box by holding down the **Option** and **⌘** keys, you can activate the macro, enter the number of times you want it to repeat, and walk away from your Macintosh. The wait in the RebuildDesktop sequence immediately checks for the dialog box and clicks the OK button. Then it settles back and waits until the dialog box appears again. The macro will continue clicking the OK button until the desktop files on all of your hard drives or partitions have been rebuilt.

Quitting the Finder

How frequently have you had to quit a program to free up additional memory that you could have gained by quitting the Finder? Or have you ever fired up ResEdit or another application and been prevented from modifying your Finder because it was already running? Your only recourse seemed to be rebooting from a different system disk or making a copy of the Finder and modifying the copy. One little known fact about System 7 is that you can quit the Finder and then run it later. This frees up valuable RAM and it also makes it possible to work on the Finder without rebooting your Macintosh.

To do this, you may want to have a program that allows you to restart the Finder. OnCue II, HandOff II, or MultiMaster all let you do this. If you don't mind quitting all running programs to restart the Finder, you won't need the program to launch the Finder. However, this means that you will have to quit CEIAC and you won't be able to send any additional Apple Events until you restart your Macintosh.

Before we begin: We will create a macro to send a Quit Application Apple Event to the Finder. When you are ready to launch the Finder, you can either quit all currently running programs or launch it using one of the programs mentioned above. You should be running System 7 and should have CEIAC and Apple Events installed. Open QuickKeys and select Universal from the pop-up menu.

Create the macro:

1. Open QuickKeys if you've closed it.
2. Select Apple Events from the Extensions menu.
3. When the extension opens, select Finder from the Send Event to pop-up menu (Fig. 10-31).

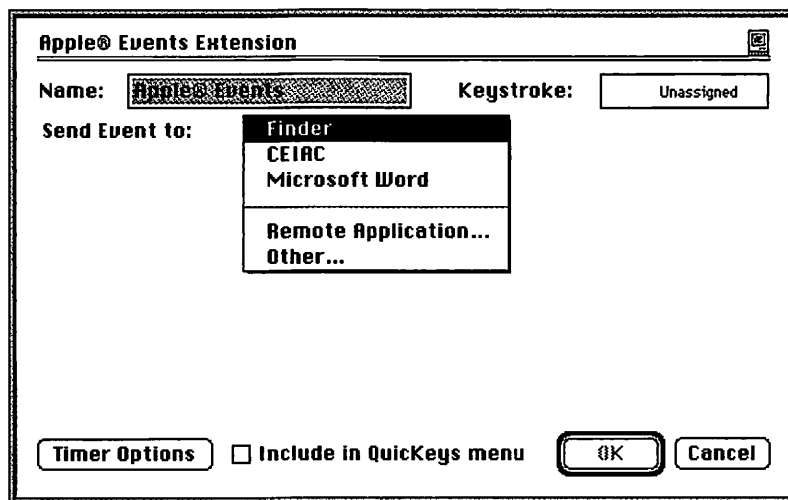


Figure 10-31: Select the program to receive the Apple Event.

4. Then select Quit Application from the Event menu (Fig. 10-32). You'll notice that QuickKeys automatically enters the event information for you.
5. Name your QuickKey and assign a keystroke. Click OK to close the extension.
6. Click OK to close QuickKeys and save your changes.

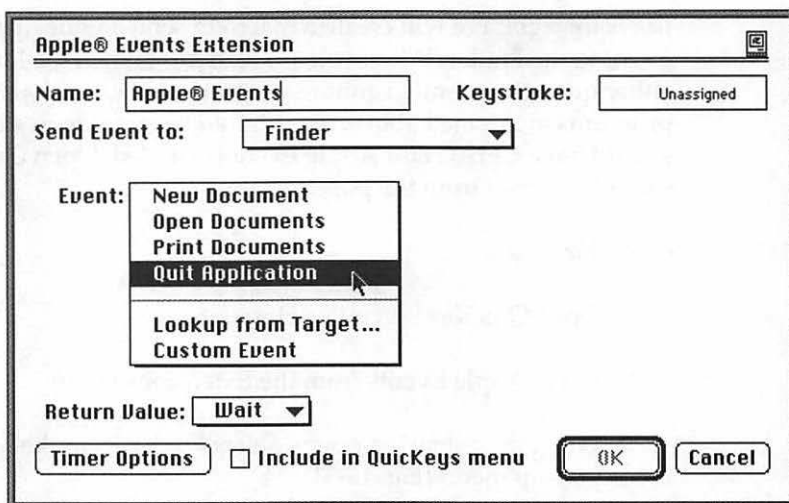


Figure 10-32: Selecting the event to send to the Finder.

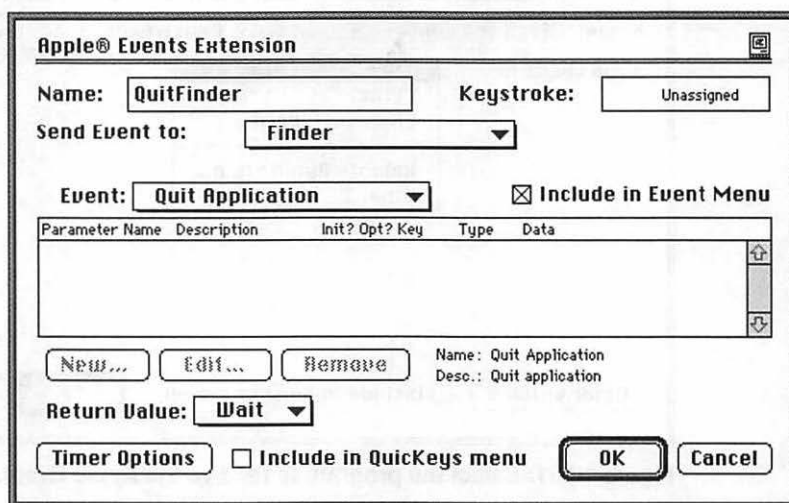


Figure 10-33: The completed QuitFinder macro.

Use and variations: You could combine this macro with the RebuildDesktop macro above if you want to avoid restarting your Macintosh. Once you have quit the Finder, you could restart the Finder with one of the programs mentioned above. Also, you could create a Quit Application QuickKey for any program.

While the advantages of the Quit Application Apple Event might not be obvious, it can reduce steps in more completed macros. For example,

you can create a macro that cycles through all of your open programs and quits each one before it shuts down your Macintosh. If you know which programs are running, you can reduce the switching and quitting to a single Quit Application QuicKey for each program.

Managing Files

As I mentioned at the beginning of this chapter, you do most of your file management in the Finder. Unlike the DOS system, however, the Finder doesn't have a scripting language that makes managing files routine. If you need to back up files or jump immediately into a folder, you have to find and copy the files you want to back up (unless you buy a program that does it for you). If you want to open a folder that is buried several layers deep in the hierarchy, you have to open each folder on the way to that one particular folder. Fortunately, QuicKeys makes it possible for you to overcome these limitations. In this section, we'll discuss macros for copying files and opening folders without digging.

Showing Buried Folders

If you have a great number of files and folders on your hard disk you probably know how difficult it can be to navigate the Finder. If you have a small screen, you certainly know how problematic it is to double-click your way through the layers of folders to get to the files you want to see. Holding down the **Option** key to close the parent folder when you double-click on a folder helps clear the screen some. The Show Finder Event in the Finder Events extension solves this problem once and for all. It lets you select files or folders to display whenever you invoke the QuicKey. If you, for instance, often add new System extensions, you could create a show QuicKey to open your Extensions folder. I routinely copy the QuicKeys Clipboard files from my QuicKeys folder which is buried inside the Preferences folder inside the System folder, so I have a show QuicKey that opens the QuicKeys folder. You can use the Show Finder Event to give you faster access to any file or folder buried several folders deep on your hard disk.

Before we begin: Before you can create a Show Finder Event, you will need to be running System 7 and have CEIAC and the Finder Events extension installed. Open QuicKeys and select Universal from the pop-up menu. If you are in a program other than the Finder, the QuicKey will switch you to the Finder and then display the QuicKey folder.

Create the macro:

1. Open QuickKeys if you have closed it.
2. Select Finder Events from the Extensions menu.
3. When the Finder Events editor opens, select Show from the pop-up menu (Fig. 10-34).

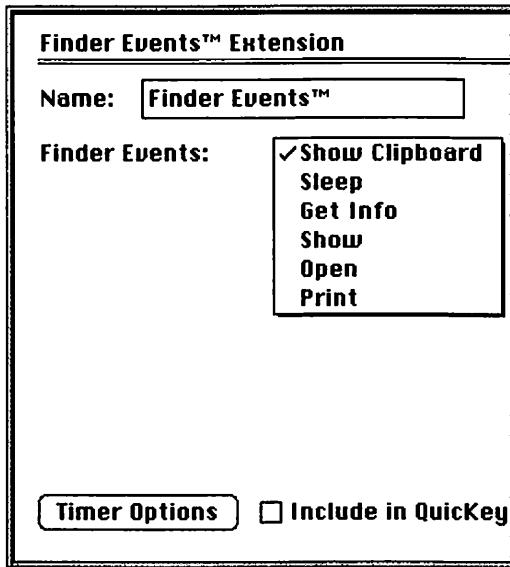


Figure 10-34: The pop-up menu of events in the Finder Events extension.

4. When the Open dialog box appears, look for the Preferences folder in your System folder. Open the Preferences folder, find and open the QuickKeys folder, and then select a file or folder in the QuickKeys folder.

The Show Event opens the folder containing the file you selected and highlights the file. If you select the QuickKeys folder, the Preferences folder opens and the QuickKeys folder is highlighted. If you want to open the QuickKeys folder with the Event, you must select a file or folder within it.

5. Click the Add button. The file or folder you've selected will appear in the Select files box at the bottom of the extension.
6. Click the Done button to close the dialog box. Notice that the file or folder you selected is now listed in the Finder Events extension

below the pop-up menu (Fig. 10-35). You can change the files or folders you have selected at any time by clicking the Change... button.

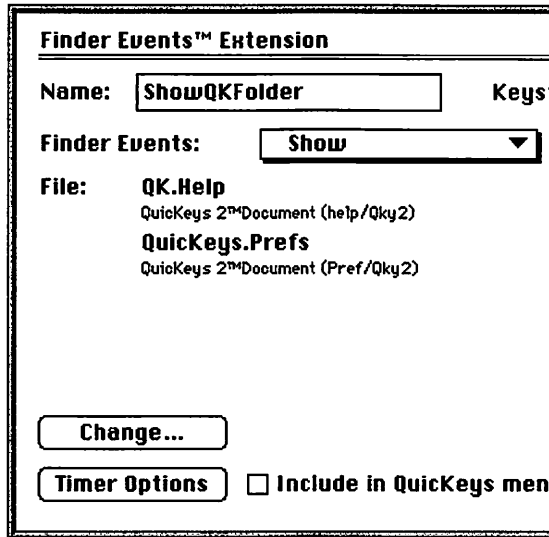


Figure 10-35: The ShowQKFolder macro after the files are selected. You can change the assigned files by clicking on the Change button.

7. Name your macro, assign a keystroke to it, and then click OK to close the extension.
8. Click OK again to close QuicKeys and save your changes.

Variations: You can create show QuicKeys for folders you frequently open that are buried deep within other folders on your hard drive. For example, if you regularly add fonts and sounds to your System file, you could create a show QuicKey for your System file. It would be similar to the macro you created above, but you would select your System file instead of the QuicKeys.Help and QuicKeys.Prefs files. When you activate the show QuicKey for your System file, it would open your System folder and automatically scroll to and highlight your System file.

Backing Up Project Files

Backups are a major headache. You shouldn't use your Macintosh without routinely backing up your data files since you would have to re-create all of your work if your hard drive failed. Unfortunately, it takes time to methodically copy all changed files or to set up a program


to make backups, and users aren't anxious to take the time to back up their files. Or they forget to copy their changed files. Have you ever wondered why the Finder can't back up your files? Assuming that you don't want to compress your files into one larger file, it would make more sense for the Finder to copy your changed files to another volume. While you can't coerce the Finder to back up your files for you, you can create a macro that copies your work files to another volume. When you are through working for the day, you can run the macro and begin cleaning up your desk while your Macintosh does the tedious job for you. This macro is easy to create if you tend to organize all of your related project files in a folder since you don't have to search your hard disk for all changed files and then copy them.

Before we begin: This macro will work under Systems 6 and 7. You should have DiskTop or a similar program, like MasterFinder, DiskTools, or On Disk. You will need to open DiskTop's Preferences dialog and select the Files and Current Location options. These settings cause DiskTop to open to the files view of the folder in which you are currently working. If you are using one of the other programs, you will need to make some modifications to the macro for it to run correctly. You should open QuickKeys and select Finder from the pop-up menu. You could make the macro Universal, but that only makes your Universal keyset larger, and you will probably run the macro from the Finder most of the time.

Create the macro:

1. Open QuickKeys and select Sequence from the Define menu.
2. When the Sequence editor opens, select Menu/DA... from the Define menu. When you are prompted to select an item from the menu, select DiskTop from the Apple menu.
3. Select WindowAction from the Extensions menu. When the WindowAction editor opens, enter DiskTop in the Window Name text box. Then click the Wait... radio button (Fig. 10-36).

Name the QuickKey to make it easier to read the macro and close the editor to save your changes. The WindowAction extension causes the macro to pause until the DiskTop window comes to the front.

4. Click the Literal button. When you are prompted, press -T. This is DiskTop's keystroke to toggle from the files view to the

DiskTop view. Because DiskTop defaults to the current folder, we need to change the view before we change the folder with the Location extension so DiskTop will open to the folder containing your files.

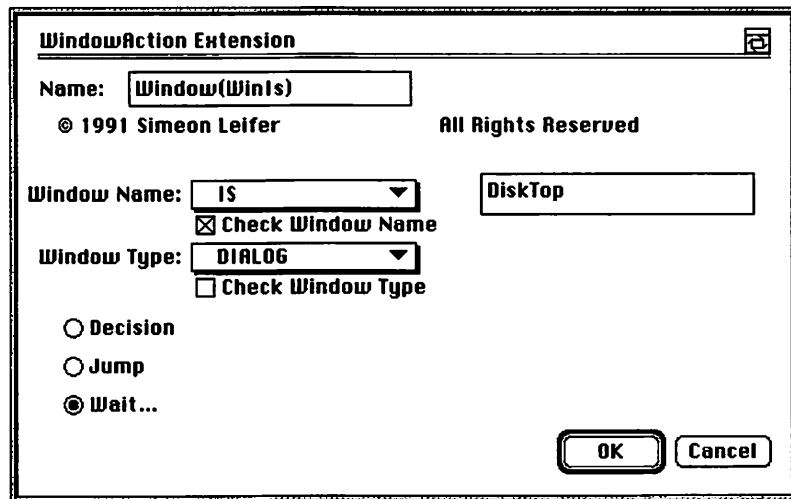


Figure 10-36: Specifying a window name for the WindowAction extension to wait.

5. Select Location from the Extensions menu. When the Location editor opens, click the Select Location button (Fig. 10-37).

A Select location dialog box opens so you can choose the folder which contains the files that you want to backup (Fig. 10-38). Be sure to enter the folder before you click the Select button (the Open button should be dimmed). When the dialog box closes, you should see the path to the folder you have just selected should appear below the Set Standard File Location button.

Click OK to close the Location editor and save your changes. When the macro runs, this sets DiskTop to open in your project's folder.

6. Click the Literal button and press **Enter** when you are prompted for a keystroke. This is DiskTop's keystroke to return to the files view in the DiskTop window. Pressing this keystroke after you have set the location displays your project folder.

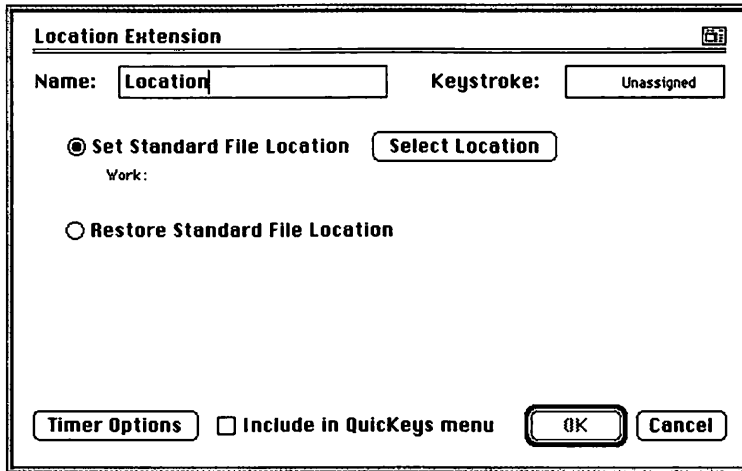


Figure 10-37: The Location editor. Notice the beginnings of a path below the Set Standard File Location radio button.

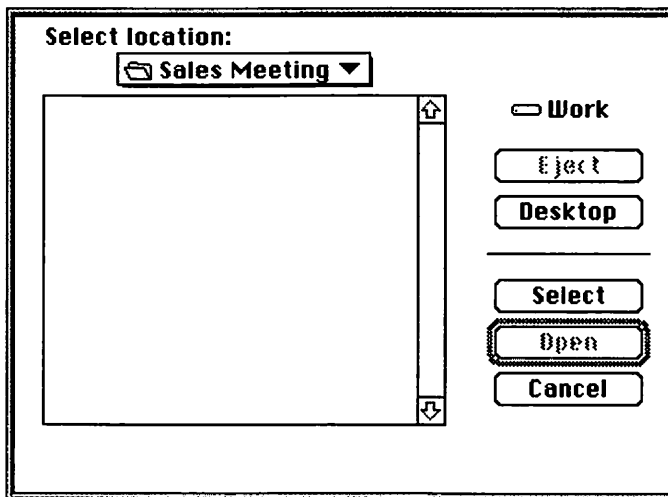


Figure 10-38: Using the Location editor to select the folder that contains the files you want to back up.

NOTE: If your project folder is not on your boot drive and your hard disk has more than one partition or you have more than one hard disk attached to your Macintosh, you may need to add a **Tab** to change DiskTop to the partition or disk the project folder is on.

- Click the Literal button. When you are prompted for a keystroke, press **⌘-A**. Now that we have DiskTop opening to the folder containing the files and folders you want to back up, the **⌘-A** selects the contents of the folder in DiskTop's window (Fig. 10-39).

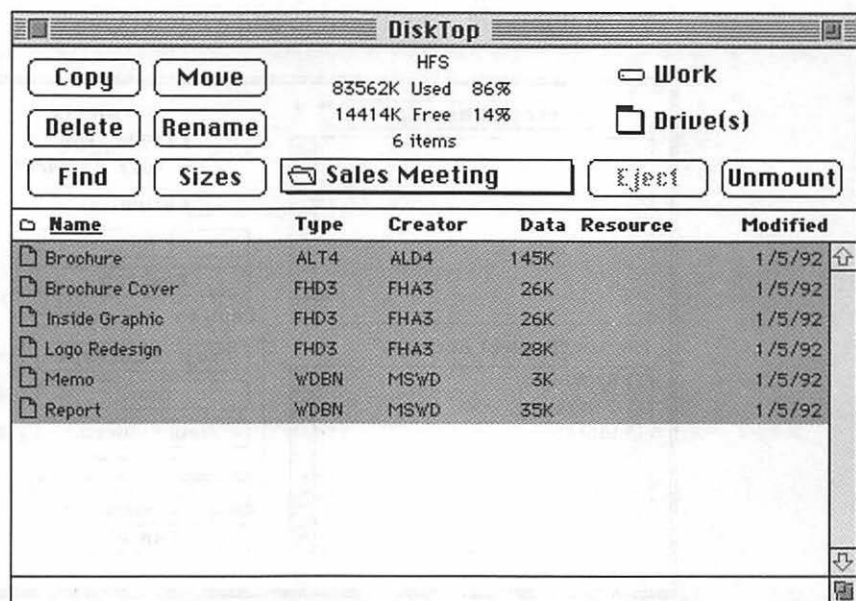


Figure 10-39: the DiskTop window open to your project folder.

- Click the Literal button again. When you are prompted for a keystroke, press **⌘-C**. This clicks the Copy button in DiskTop's window.
- Select Location from the Extensions menu again. When the editor opens, click the Select Location button and choose the destination folder for the backup files. Close the editor. At this point, when you run the macro, DiskTop opens a dialog box for you to select a destination for the files/folders you are copying. This location QuickKey selects the backup folder (Fig. 10-40).
- Click the Literal button. When you are prompted for a keystroke, press **⌘-1**. This clicks the Project Backups button in DiskTop's Copy To dialog.

You have now created the body of the macro which will jump to the location of the files, select all of them, and copy them to the new location. The first time you run this macro, it performs

exactly as you expect. But the second time you run it, DiskTop notices that the backup folder you are copying the files to also contains files with the same names. Like any well-behaved Macintosh program, DiskTop asks if you want to replace the files with the ones you are copying. To keep the macro from breaking at this point, you'll need to add a button QuickKey.

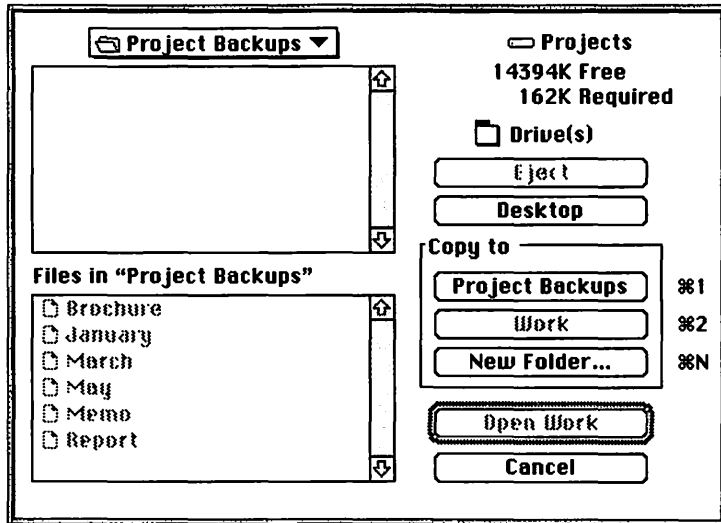


Figure 10-40: DiskTop's Copy To dialog box.

10. Select Buttons... from the Define menu. When the Buttons editor opens, enter "Replace All" in the text box and leave the Always click button selected. Close the Sequence editor.

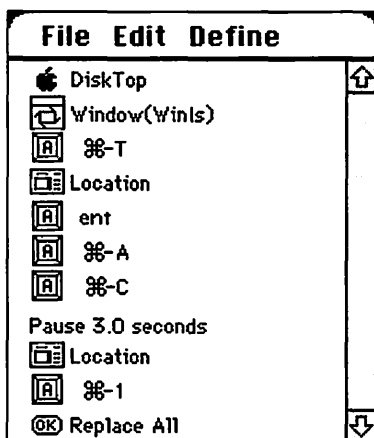


Figure 10-41: The final BackUpProject macro.

11. Click OK to close QuicKeys and save your changes.

Use and variations: If you really don't trust your Macintosh, you can add a pause before the ⌘ -A. The pause would give you enough time to interrupt the macro if DiskTop has not opened to the correct folder. If you are using DiskTop, you shouldn't have to worry about it overwriting your changed files. If it copies your files to your work folder, "Copy of" is added to the beginning of each file's name. You might also want to add a pause before the second location QuicKey. This slows the macro down and ensures that QuicKeys selects your backup folder. As I have said before, you could adjust the overall speed of QuicKeys, but I prefer to run my macros as fast as possible and make adjustments in the individual macros when QuicKeys needs to slow down.

Two variations on this macro immediately suggest themselves. You could have the macro create incremental backups of your work. That is, instead of overwriting your previous backup files, you could create a folder with the current date and copy the files to it. This would allow you to keep the daily or weekly changes to your files. If you need to recover the previous week's version of a file or a graphic, it would be available in one of your incremental backups. You only need to add a few steps to the current macro to make it incremental.

1. Move the insertion arrow between the second location QuicKey and the ⌘ -1 alias.

You'll notice that in DiskTop's Copy To dialog box there is a New Folder button. We insert steps in our macro to click the New Folder button, create a new folder, and then name the folder before the files are copied.

2. Select Alias from the Define menu. When the editor opens, press ⌘ -N. This keystroke clicks the New Folder button. Close the editor.
3. Select Text... from the Define menu. When the Text editor opens, enter a name for the folder followed by a trailing space. We'll enter "Backup " for the purposes of this macro. The trailing space is necessary to separate the name from the date.
4. Select Date/Time... from the Define menu. When the editor opens, select one of the Date formats from the pop-up menu. You might want to use the mm/dd/yy format to keep the name short. Click OK to save your changes.

At this point when the macro is running, QuickKeys has opened the New Folder dialog box and entered the name and date. The dialog box should look similar to the one in Figure 10-42. Now we need to enter a **Return** to click the Create button. DiskTop automatically enters the new folder so your files are copied to it.

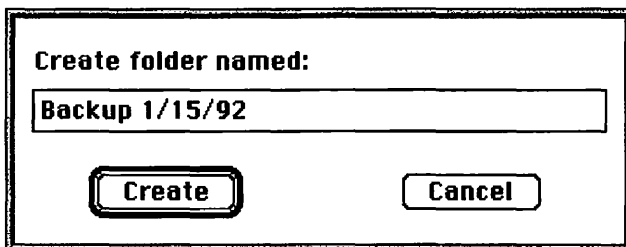


Figure 10-42: The New Folder dialog box after the text and time/date QuickKeys have run.

5. Select Alias from the Define menu. Hold down the mouse button and press **Return** when the editor opens. Click OK to close the editor.

The **⌘-1** from the original macro clicks the Copy To button in the dialog box (Fig. 10-40 above) and initiates the copy process.

6. Remove the Replace All Button QuickKey from the original macro. Now that the files are copied to a new folder, DiskTop's warning dialog won't appear. The completed macro looks like Figure 10-43.

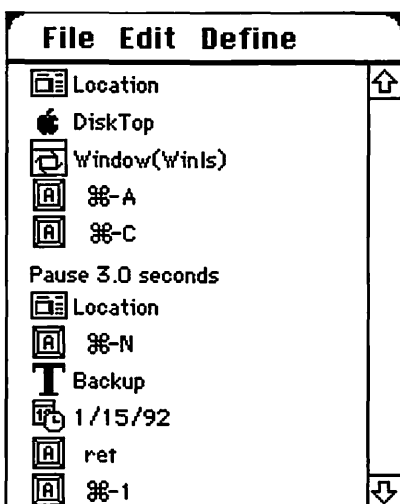


Figure 10-43: The final IncrementBackUp

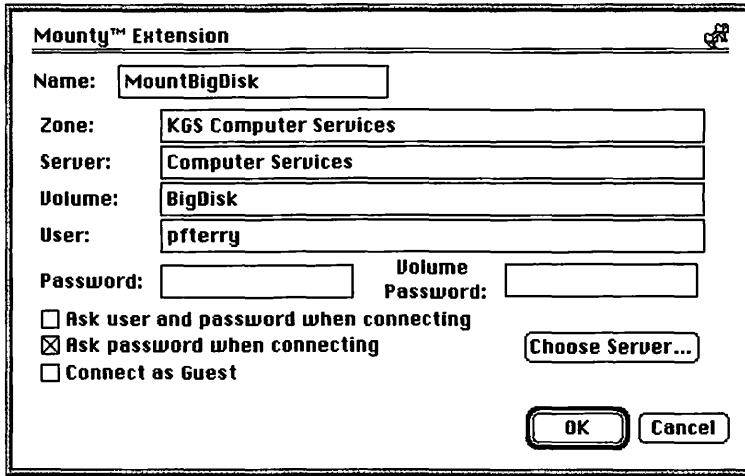
A few words of warning: while extensions and INITs are wonderful in the ways they allow you to modify your system, you have to remember that they can intervene in your macros in ways that you may not anticipate. If you are using On Disk, which is part of On Cue II, you will need to modify the macro to work with the New Folder command which On Disk adds to Save dialog boxes. Shortcut and Super-Boomerang also add a New Folder command to Save dialog boxes.

The second variation on the macro is letting it mount the AppleShare server in your office and backing up your files to it. This adds an additional layer of security to your backups. If anything happens to your hard disk, you still have your files on the AppleShare server. You will need to add steps to the beginning and the ending of the original macro to add the additional functionality. You will also need to change the second location QuicKey.

1. Using the Chooser, mount the server volume to which you will be copying your files. If you aren't familiar with using remote or AppleShare volumes, refer to your AppleShare or System 7 documentation.
2. Open QuicKeys.
3. Open either the BackUpProject or the IncrementBackUp macro.
4. Move the insertion arrow to the beginning of the sequence and select MounTy from the Extensions menu. When the MounTy editor opens, enter the zone, server, volume, and your user names. If you are always going to sit through the beginning of the macro and you don't want your Macintosh entering your password, click the Ask password when connecting checkbox. The completed set up should look something like Figure 10-44. Close MounTy.

CAVEAT: *If your files are sensitive, you should never enter your password in a QuicKeys Sequence. When you do enter the password in a sequence, anyone can open it, find your password, and get access to your files.*

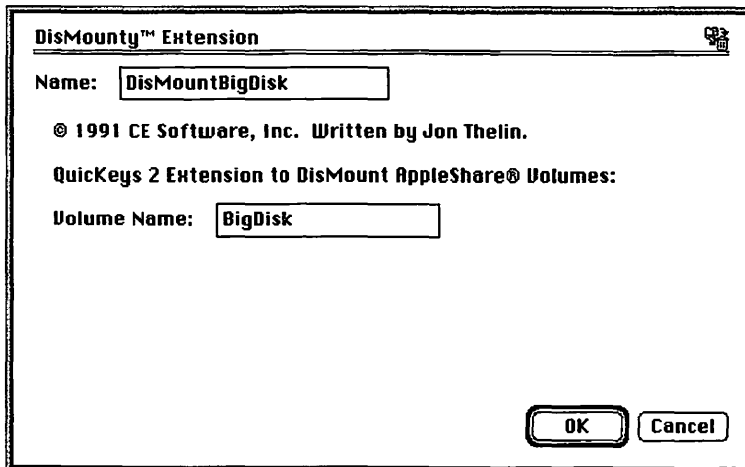
5. Open the second location QuicKey and click the Select Location button. When the Select Location dialog opens, select your backup folder on the server. Close the Location editor to save your changes.



The Mounty™ Extension dialog box is shown. It has a title bar with the text "Mounty™ Extension" and a small icon on the right. The fields are as follows: Name: MountBigDisk; Zone: KGS Computer Services; Server: Computer Services; Volume: BigDisk; User: pfterry; Password: (empty); Volume Password: (empty). There are three checkboxes: "Ask user and password when connecting" (unchecked), "Ask password when connecting" (checked), and "Connect as Guest" (unchecked). A "Choose Server..." button is to the right of the checkboxes. At the bottom right are "OK" and "Cancel" buttons.

Figure 10-44: The Mounty extension mounts a server volume without going through the Chooser.

6. Move the insertion arrow to the end of the sequence, and select DisMounty from the Extensions menu.
7. When the DisMounty editor opens, enter the volume name of your server in the Volume Name text box (Fig. 10-45). This step is more a matter of convenience than necessity. You may prefer to leave the server mounted on your desktop so you can verify that the files were backed up.



The DisMounty™ Extension dialog box is shown. It has a title bar with the text "DisMounty™ Extension" and a small icon on the right. The fields are as follows: Name: DisMountBigDisk; Copyright: © 1991 CE Software, Inc. Written by Jon Thelin.; Description: QuickKeys 2 Extension to DisMount AppleShare® Volumes:; Volume Name: BigDisk. At the bottom right are "OK" and "Cancel" buttons.

Figure 10-45: The DisMounty extension allows you to automatically unmount the server after the files have been copied.

8. Click OK to close the DisMount editor and then click OK to close QuicKeys and save your changes.

You now have macros that can back up your files and folders for you. Now you don't have any excuse not to back up your files. Right?

Backing Up Modified Files

If you don't organize your files by project, the macro above may not be the approach you want to take. But you can create a macro that uses DiskTop to search for all of the files with your most recent changes. When DiskTop finds the modified files, you can take a portion of the macro above to copy those changed files to another volume or an office server. Since it is difficult to exclude programs and system files whose modification dates change daily—like the Finder—this macro may not be the best substitute for a real backup program. But it can be a quick way to get a backup when you don't have time to run a backup program.

Before we begin: You should set up DiskTop and QuicKeys as described in the ChangeCreator sequence described in Chapter 2. You should also have DiskTop.Extras and CE Toolbox installed so the DiskTop Find command is available from the Apple menu. You should configure CE Toolbox with menus either grouped or sorted with DAs. Open QuicKeys and select Universal from the pop-up menu.

Create the macro:

1. Select Record Sequence from the QuicKeys menu.
2. Select DiskTop Find from the Apple menu or press the DiskTop's default keyboard shortcut.
3. When the Find Criteria dialog box opens, click the Modified checkbox to select it. If you are using Escapade, you can type **⌘-M**. This tells DiskTop to search for all of the files you modified on that day.
4. Press **Return** or click the Find button. This initiates the search for all modified files. Notice how long it takes DiskTop to complete the search since you will need to enter a pause later in the macro.

5. When DiskTop finishes searching for all modified files, click the Retain button to save the list of modified files to a DiskTop window. Again, you can type **⌘-R** if you are using Escapade.
6. Select Stop Recording... from the QuickKeys menu.
7. When the Sequence editor opens, you will need to make some changes to the macro to make it wait until DiskTop is through searching the hard disk.

Move the insertion arrow between the **Return** and the last click and select Pause from the Define menu. When the Pause dialog opens, enter the amount of time that you think it takes DiskTop to search your hard disk.

8. Let's add a pause at the beginning of the macro so it won't run while the DiskTop Find dialog is opening.

Move the insertion arrow between the menu QuickKey and the first click and select WindowAction from the Extensions menu. When the WindowAction editor opens, click the Check Window Name checkbox to deselect it. Then click the Check Window Type checkbox to select it. Select Dialog from the pop-up menu and then click the Wait... radio button. Click OK to close the editor.

9. At this point the macro should look similar to Figure 10-46. Give the macro a name and close the Sequence editor.

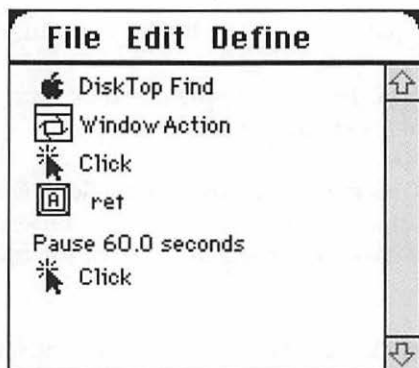


Figure 10-46: The BackUpMods macro before adding the QuickKeys from the IncrementBackUp macro.

- Now you are ready to copy part of the macro you created above. When you are back in the QuickKeys window, open the IncrementBackUp macro. Hold down the **(Shift)** key and click the WindowAction QuickKey and then drag to the bottom of the macro (Fig. 10-47).

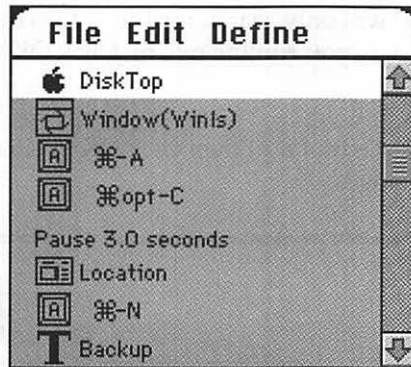


Figure 10-47: Selecting the QuickKeys in the IncrementBackUp macro.

- Select Copy from the Edit menu and click OK to close the Sequence editor. You have copied the QuickKeys that wait until the DiskTop window comes to the front, select the files in the window, create a backup folder, and copy the files to it.
- Open the BackUpMods macro and select Paste from the Edit menu. The QuickKeys are pasted from the Clipboard into the Sequence editor.
- Click OK to close the editor. Then click OK to close QuickKeys and save your changes.

Variations: Since this macro finds all of the changed files on your hard disk, you may want to add a text QuickKey that inserts a file type in the Type text box. For example, you can search only for your changed Microsoft Word files. If you use Word most of the time, this refines your search and reduces the amount of time it takes for the macro to run. If you are using DiskTop and do not see the type and creator in the files window, you will need to open the Preferences dialog and click the Technical radio button under Level.

- Move the insertion point between the WindowAction QuickKey and the click at the top of the macro.

2. Click the Literal button. When you are prompted for a keystroke, press **Tab**. The **Tab** will move the insertion point to the Type text box (Fig. 10-48). Click OK to close the editor.
3. Select Text from the Define menu. When the editor opens, type "WDBN" in the text box. This enters the file type for Microsoft Word so DiskTop will only search for Word files that have been modified on the date you run the macro. Click OK to close the editor.
4. Close the Sequence editor and then click OK to close QuicKeys and save your changes.

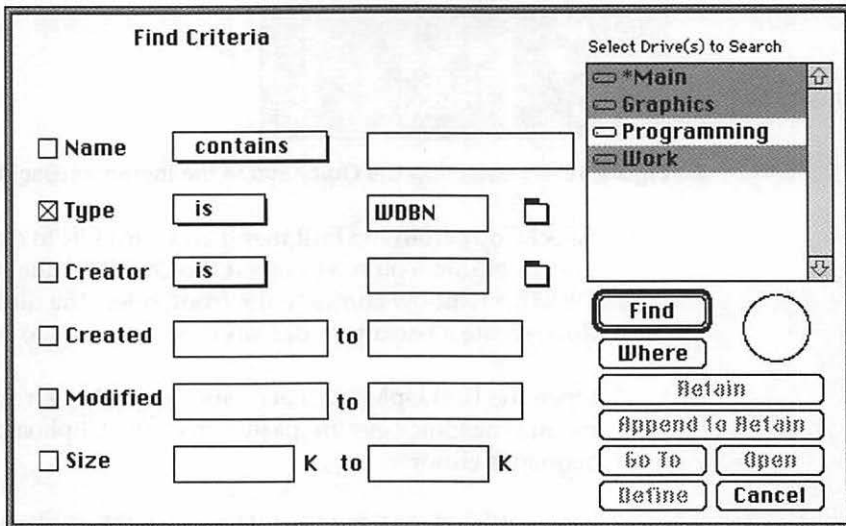


Figure 10-48: The DiskTop Find Criteria dialog with the Word file type entered in the Type text box.

You could also vary this macro by entering other dates, file type or creator information, or names in the text boxes. If you want to search for several different file types and refine each search, you can add a pause and wait for user QuicKey and let the macro finish running after you have saved the results of each search to DiskTop's Found window.

Last But Not Least

I hope some of the macros in this chapter will simplify your interaction with the Finder and automate the tasks you do every day. More importantly, I hope the macros have made you think of ways to improve your computing environment.

We have talked about System 7 features and InterApplication Communication using Apple Events in the previous chapters of the book, but you'll discover that Chapter 11 is devoted to System 7 and Apple Events. In it, we'll create macros for compressing files, looking up addresses in a database, and operating your Macintosh remotely.

QuickKeys and System 7

One of the core technologies in System 7 is Apple events. While most developers are rushing to add Apple event support to their programs, only a few programs that support Apple events are available at this writing. For instance, as I am finishing this book, only two companies have announced word processors with Apple event support, and neither of those programs is shipping. On the other hand, several programs have already incorporated support for Apple events. Canvas 3.0, as we've already seen in Chapter 7, has a rich set of Apple events, and StuffIt Lite—a shareware file compression application—has a complete set of Apple events for compressing files.

What Are Apple Events?

As I've stated elsewhere in this book, Apple events allow applications to communicate with one another on the same Macintosh computer or over a network. If a developer is really following Apple's specifications, the user interface of an application will be able to drive the application by sending Apple events to it. Apple events are grouped into functionally related sets, or suites, and Apple has compiled a listing of these suites in the *Apple Event Registry*. Every System 7 savvy program must at the very least support the Required suite of Apple events. The Required suite contains the Open Application, Open Document, Print Document, and Quit Application events.

An Apple event has several different components. Each event has a class and ID and may contain one or more parameters. As with suites, events are grouped together in classes of related events, and the event ID uniquely identifies the event. For instance, the required set of events has the suite "reqd." If you wanted to open a file with an Apple event, you

would send the Open Document event to the Finder; “aevt” is the class designation, and “odoc” is the event ID. You would also need to tell the Finder which document you wanted to open and send this information as a parameter.

The number of parameters varies depending on the Apple event. Each parameter of the event is represented by a four-character code. For example, the parameter that sends the path to a file in StuffIt Lite has “path” as its four-character code. This parameter tells StuffIt Lite which file to compress. The parameters usually indicate the object (i.e., window, document, script, etc.) to which the Apple event should apply, the data to use (e.g., a picture to paste), and the desired options when performing the event (e.g., with/without dialog boxes, page numbers to print, etc.).

When the AppleScript user programming language is released early next year, it will be the interface for Apple events, a system-wide set of commands. In the interim, the Apple events extensions in QuickKeys can already send Apple events directly to programs with event support. At best, this is a rudimentary introduction to Apple events. It should, however, be enough to get you started with the most exciting development in Macintosh computing.

The remainder of this chapter discusses how to incorporate Apple events into your sequences to make your macros even more powerful. We have already seen how easy it is to send Apple events to programs with QuickKeys and will look at examples of moving information between programs with QuickKeys. For instance, QuickKeys can greatly reduce the number of steps required to look up an address and paste it into your letter or report. Probably the most impressive use of QuickKeys is sending and receiving Apple events between programs on networked Macintoshes. This is a boon for network managers. If they have QuickKeys installed on the Macintoshes on their networks, they can significantly reduce the wear and tear on their sneakers.

Making System 7 More Automatic

By the time you get to this chapter, you should be able to think of dozens, or hundreds, of ways to make your Macintosh do more of your work for you. If you haven’t already realized it, QuickKeys and System 7 are an astounding team. Let’s look at a couple of QuickKeys that build on the power of System 7’s networking capabilities.

Startup File Sharing

As we saw in the previous chapter, you can toggle peer-to-peer file sharing off and on using the System 7 Specials extension. Unfortunately, the System 7 specials only work on the Macintosh you are currently using. If you want start up file sharing on the Macintosh in the next room—or the next building—so you can get a file from it, you have to get up and walk over to it. Or do you?

If you have the QuicKeys' Apple Events extension and CEIAC installed on both Macintoshes, you can send an Apple event to CEIAC running on the remote Macintosh to start file sharing. You can set up a macro to start file sharing on the remote computer.

Before we begin: You will need to create a user account for yourself on the remote Macintosh and turn on Program Linking in the Sharing Setup control panel (Fig. 11-1). If you do not remember how to create a user account, refer to the section about the Users & Groups control panel in your *Macintosh User's Guide* or *System 7's Macintosh Networking Reference*. You will also need to have QuicKeys and CEIAC running on the remote Macintosh, and CEIAC will also need to be in foreground mode. You will need to create System 7 specials on it to toggle file sharing on and off like we did in Chapter 10; we are going to send Apple events to play them.

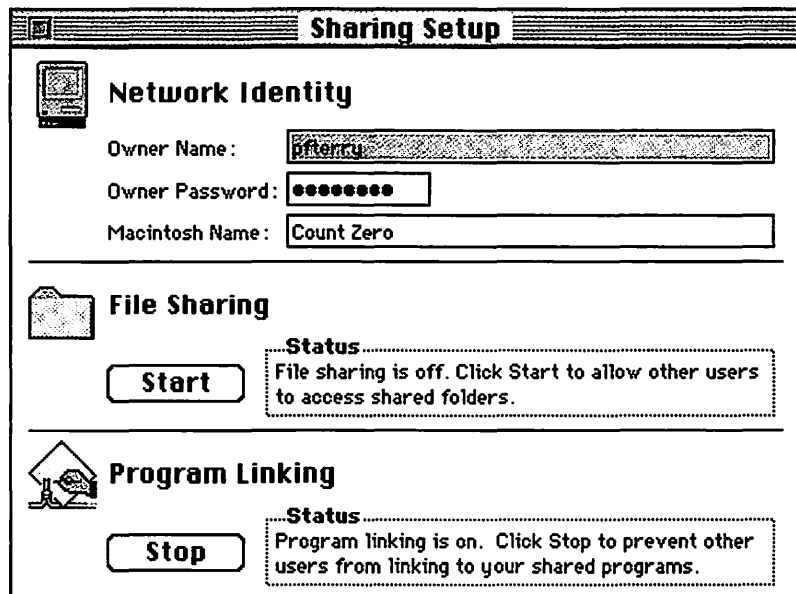


Figure 11-1: The Sharing Setup control panel.

Create the macro:

1. Open QuickKeys and select Universal from the pop-up menu. We are selecting the Universal keyset so the macro will be available no matter what program you are in.
2. Select Apple Events from the Extensions menu. When the Apple Events editor opens, select Remote Applications from the Send Event to pop-up menu (Fig. 11-2). The Send Events pop-up menu contains a list all of the programs that are currently running on your Macintosh.

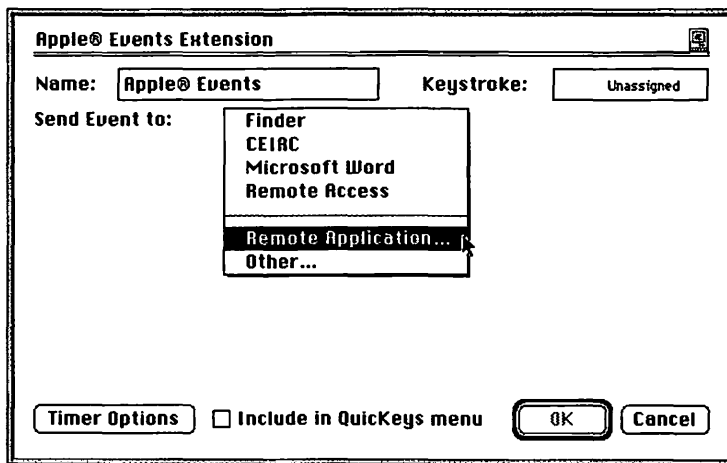


Figure 11-2: The Send Event to pop-up menu in the Apple Events extension.

After you select Remote Application... the Program-to-Program Communication (PPC) Toolbox opens which lists your AppleTalk zones, the Macintoshes available in those zones, and the programs running on those Macintoshes (Fig. 11-3). You can press **Tab** to move among the three windows in the dialog box.

3. Select a zone from the bottom scrolling list. All of the Macintoshes available in that zone will appear in the Macintoshes scrolling list.
4. In the top-left scrolling list, select the Macintosh to which you want to connect. All of the programs running on the remote Macintosh you have selected appear on the right.

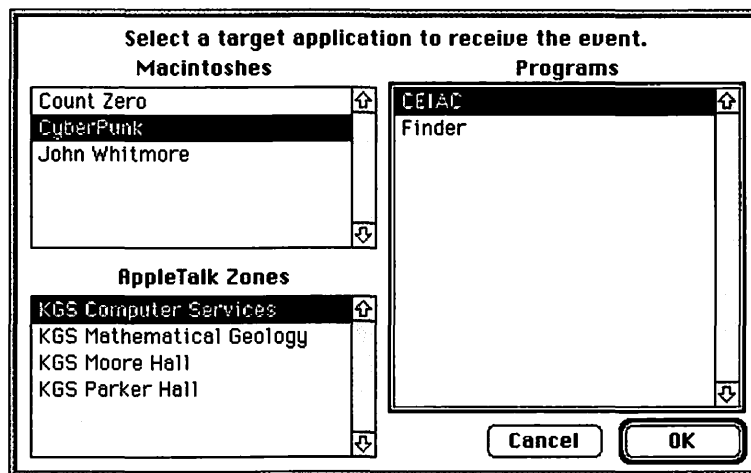


Figure 11-3: The System 7 Program-to-Program Communication Toolbox.

5. Select CEIAC from the list of programs on the left, and then click OK or press **Return**. The dialog box opens with another Event pop-up menu. You'll notice that the Macintosh you are linking to and its zone address show up in the dialog box (Fig. 11-4).

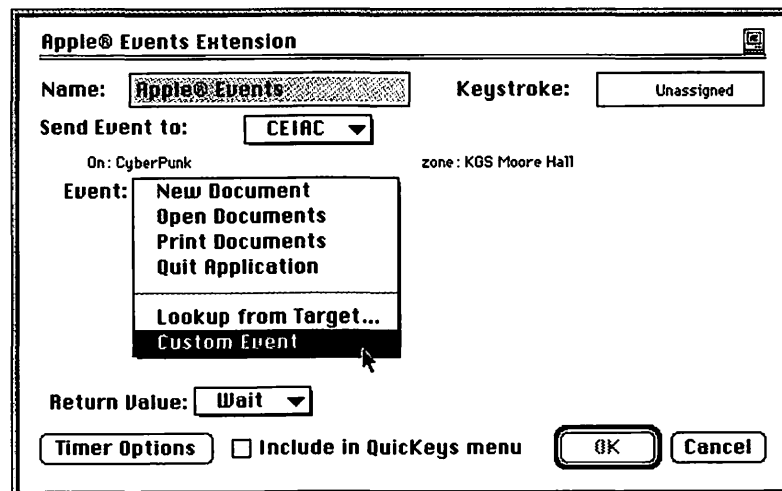


Figure 11-4: The Event pop-up menu in the Apple Events editor.

6. Select Custom Event from the bottom of the Event pop-up menu. We select Custom rather than Lookup from Target because CEIAC doesn't have the aete resource that allows QuickKeys—and other programs—to get information about the suites of Apple

events that CEIAC supports. Notice that the Custom Event option adds a scrolling list, three buttons, and Class and ID text fields to the bottom of the dialog box (Fig. 11-5).

NOTE: CEIAC 2.0 has an *aete* resource.

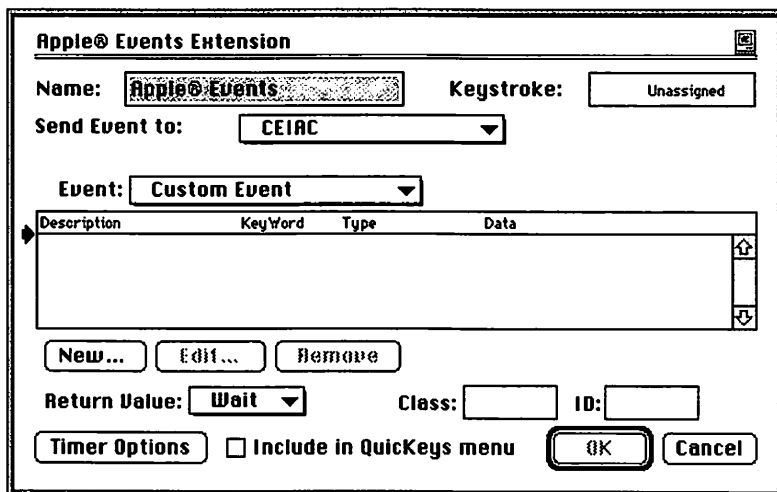


Figure 11-5: The Apple Events editor after selecting Custom Event from the pop-up menu.

7. Click the New button to open the parameters dialog box (Fig. 11-6).

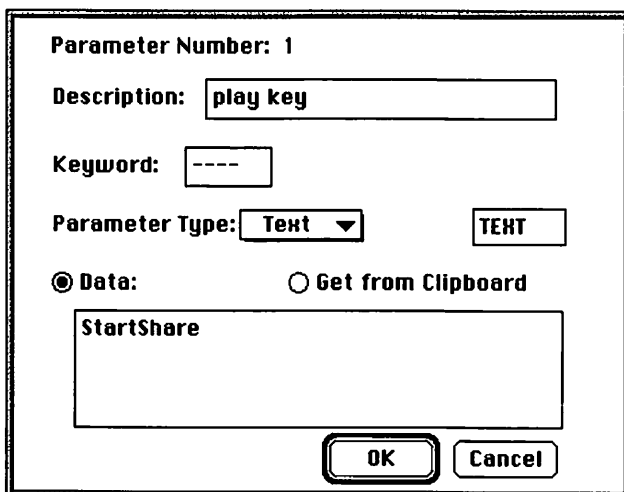


Figure 11-6: The parameter settings for the StartSharing macro.

8. When the dialog box opens, enter a description for the Apple event, enter "----" for the keyword, select Text from the pop-up menu, click the Data radio button, and enter the name of the QuickKey on the remote Macintosh.
9. Click the OK button or press **Enter** to close the parameters dialog and save your changes.

NOTE: Do not press **Return** in the parameter dialog box. It enters a **Return** after the macro name and causes the macro to fail.

10. When the parameter dialog closes, type "QKy2" in the Class box and "QPNm" in the ID box (Fig. 11-7). The Class tells CEIAC that the event is in the QuickKeys suite and the ID tells CEIAC that it should send the play by name event to QuickKeys on the remote Macintosh.

NOTE: Apple events are case sensitive.

Apple® Events Extension

Name: **StartSharing** Keystroke: **Unassigned**

Send Event to: **CEIAC**

On: cyberpunk zone: KGS Moore Hall

Event: **Custom Event** ☒ **Include in Event Menu**

Description	Key Word	Type	Data
play key	----	TEXT	StartShare

New... **Edit...** **Remove**

Return Value: **Wait** Class: **QKy2** ID: **QPNm**

Timer Options ☐ **Include in QuickKeys menu** **OK** **Cancel**

Figure 11-7: The StartSharing macro with Class and ID information entered.

11. Select Display from the Return Value pop-up menu in the lower-left corner of the dialog box (Fig. 11-8). This allows QuickKeys to tell you whether or not the QuickKey has been played successfully on the remote Macintosh. If it has, a dialog box appears on your Macintosh with the name of the QuickKey in it (Fig. 11-9).

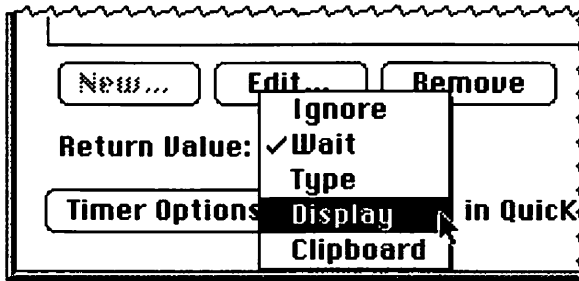


Figure 11-8: The Return Value pop-up menu allows you to select how QuickKeys treats any reply it gets from the program to which you are sending the Apple event.

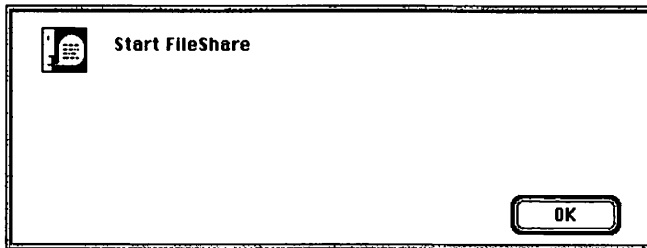


Figure 11-9: The Display Return Value option causes QuickKeys to display any information it receives from the remote Macintosh.

12. Now that you have entered all of the settings for the Apple event, name your QuickKey and assign a keystroke. Click OK to close the Apple Events editor and then click OK to close QuickKeys and save your changes.

Use and variation: When you play the StartSharing macro, QuickKeys sends a request to the remote Macintosh to link to CEIAC. CEIAC comes to the foreground on your Macintosh, and a dialog box appears requesting your user name and password (Fig. 11-10). Enter those and click OK. When you look in the Chooser in a minute or so, you should see that the remote Macintosh's name has been added to the list of AppleTalk servers.

An obvious variation on this macro would be to stop file sharing. The settings would be exactly the same, but you would enter StopSharing (or the name that you have given the macro on your remote Macintosh) in the text box at the bottom of the Parameter dialog box (Fig. 11-6).

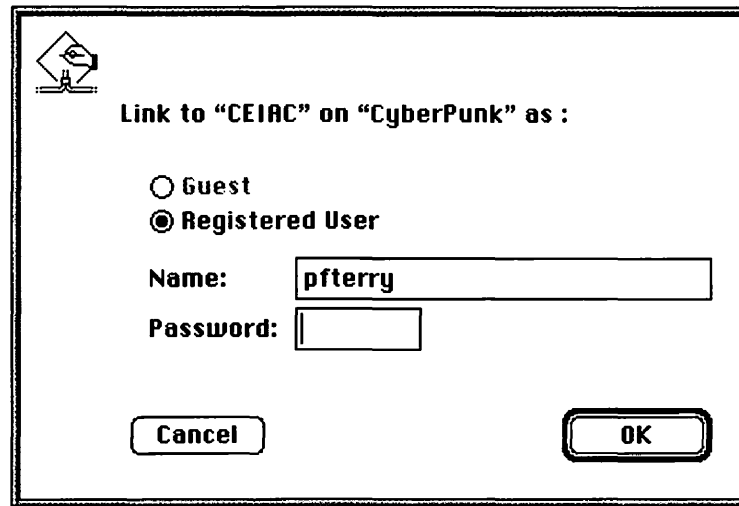


Figure 11-10: The Link dialog box that requests your name and password.

You could also use this QuickKey in a sequence with the Chooser or the mounty QuickKey to reduce the number of actions necessary to connect to another Macintosh. That is, you could send the Apple event to start up file sharing, wait for the sharing to start up, then open the Chooser in preparation for mounting a disk. Or, you could also add a mounty QuickKey to mount the disk. To make this sequence, you will need to turn on file sharing on the remote Macintosh and then follow these steps:

1. Open QuickKeys and select Universal keyset from the pop-up menu.
2. Scroll until you see the StartSharing QuickKey you created above. Select the QuickKey and then select Copy from the Edit menu.
3. Select Sequence from the Define menu.
4. When the Sequence editor opens, select Paste from the Edit menu. The QuickKey will be pasted into the Sequence editor.
5. Select Pause from the Define menu. When the dialog box opens, enter 60 in the seconds box. You may need to enter either fewer or more seconds depending on your Macintosh and your network. Click OK to close the dialog box.

6. Select Mounty from the Extensions menu. When the Mounty editor opens, click the Choose Server button. A dialog box opens showing the available zones and servers. Your Macintosh's zone is selected (Fig. 11-11). Select the zone the remote Macintosh is in (if different from the one selected), select the name of the remote Macintosh, and then click OK or press **Return**.

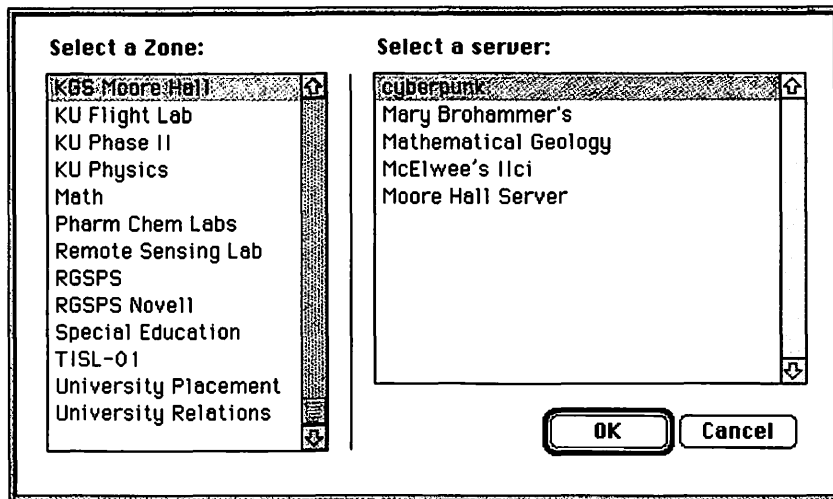


Figure 11-11: The Mounty Choose Server dialog box.

7. The Mounty editor now has most of the information in it to mount the server of the remote Macintosh (Fig. 11-12). Enter the remaining information.

NOTE: *You should not enter your password in the Mounty editor unless you know that your Macintosh is secure and others cannot use it to access your files.*

8. Click OK to close the Mounty editor. Your sequence should be similar to Figure 11-13.
9. Name your Sequence and assign a keystroke. Click OK to close the Sequence editor and click OK to close QuickKeys and save your changes.

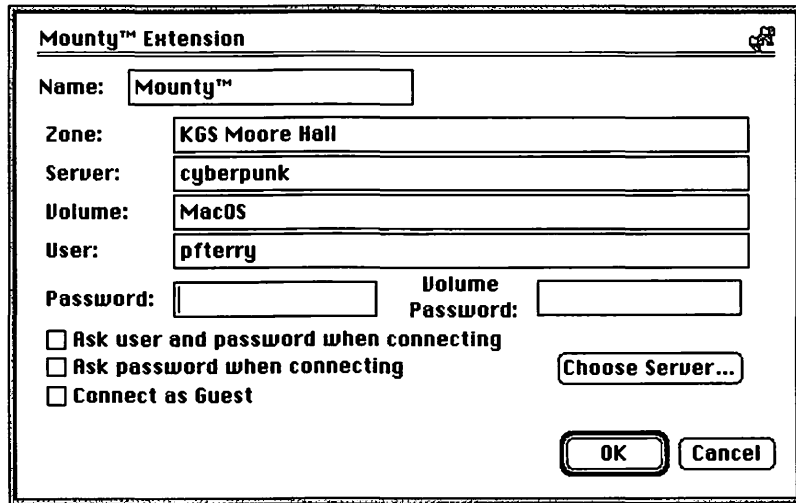


Figure 11-12: The Mounty editor with the remote Macintosh information entered.

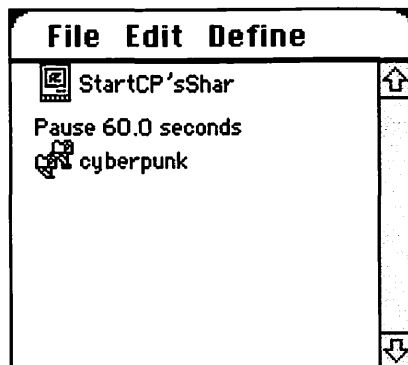


Figure 11-13: The final SharingSequence macro.

Now you have a macro that starts file sharing on a remote Macintosh and mounts the disk for you automatically. This macro allows you to leave file sharing turned off to reduce the activity on the remote Macintosh and then turn it on when you need it.

Another variation would be to team this macro with AppleTalk Remote Access. You could call your office network using AppleTalk Remote Access and turn on file sharing from your home Macintosh or your PowerBook. You might create a complementary macro that shuts down your office Macintosh when you are through using it. You would need to create a Shut Down special on your office Macintosh. In the

parameter dialog box, you would enter the name of the Shut Down QuickKey in place of the StartSharing macro.

Automate Publish and Subscribe

Another core technology in System 7 is Publish and Subscribe. The Publish and Subscribe features allow you to publish all or a portion of a document. Other members of your work group are then able to subscribe to your document, read it, and make changes. If you are working in a group and share the same project files on a regular basis, you can create a macro that publishes your document. On the other Macintosh, you can create a macro that sets up the subscriber. For instance, if you are part of a team that is generating a company report or brochure, you might want to publish the chart you've created from the quarterly sales figures. The person who is creating the report or brochure by combining someone else's text and your chart would be able to subscribe to your published chart and incorporate it into the brochure.

Before we begin: You should have Claris Resolve installed. You will also need a worksheet in which you've created a chart that you can publish. Because Resolve does not have an aete resource, we will have to create a custom event that sends a do Script Event to Resolve. Since Resolve has a built-in scripting language, we can send it a script to publish the chart.

Create the macro:

1. Open QuickKeys and select Apple Events from the Extensions menu.
2. When the Apple Events editor opens, select Resolve from the Send Events to pop-up menu.
3. Select Custom from the Events pop-up menu.
4. Click the New... button to add a new parameter to the list.
5. Select your new parameter and click the Edit button to open the parameter dialog box.
6. Enter "Do Resolve script" or something equally descriptive in the Description box.
7. Click in the Keyword box and enter "- - - -" for the parameter's keyword.

8. Select Text from the Type pop-up menu.
9. Click the large text box at the bottom of the dialog box and enter the script:

```
select chart 1
create publisher "Resolve edition"
```

The first line of the script selects the chart in your Resolve worksheet though you could have the script select any part of the worksheet. The second line of the script creates the published edition in the same folder as Resolve. Your parameter settings should look like Figure 11-14.

Parameter Number: 1

Description: Do Resolve script

Keyword: ----

Parameter Type: Text

☒ Data: ☐ Get from Clipboard

select chart 1
create publisher "Resolve edition"

OK Cancel

Figure 11-14: The parameter settings for the ResolveDoScript macro.

10. Click OK or press **Enter** to close the parameter dialog box.
11. When the parameter dialog box closes, enter "misc" in the Class box and "dosc" in the ID box in the lower-right corner of the Apple Events editor (Fig. 11-15). This class and ID tell CEIAC to send the text in the parameter dialog box to Resolve as a script.

NOTE: The Data text box in the parameters dialog box can only hold 255 characters.

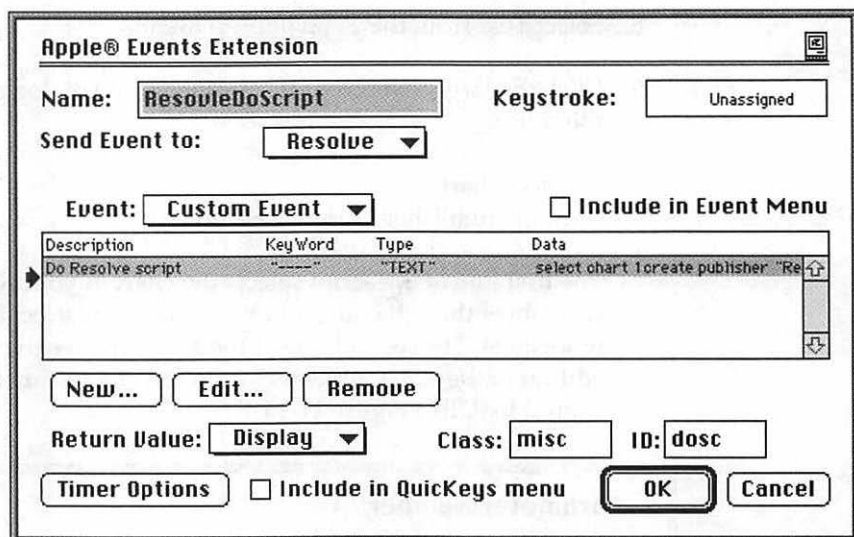


Figure 11-15: The final ResolveDoScript macro.

NOTE: If you are using Resolve 1.0v1-v3, your script can only be 255 characters long. If you are using Resolve 1.1, your script can be up to 64K. However, the Data text box can only hold 255 characters.

12. Name your macro and click OK to close the Apple Events editor and save your changes. Click OK or press **Return** to close QuickKeys.

Use and variations: Now you can publish the chart in your worksheet at a keystroke, and the others in your work group will be able to subscribe to the chart and incorporate it into their documents. If the person incorporating the chart is using a program that supports Apple events, they can create a macro that automatically subscribes to your published chart.

The Future of Macintosh Computing

Well, maybe it's an overstatement to say that Apple events are the future of Macintosh computing, but I don't think so. Apple events and InterApplication Communication really are the future of Macintosh computing. They allow you to retrieve information from one program and incorporate it into a document in another program. And QuickKeys is the prime mover for Apple events. We've spent most of the book

looking at ways QuickKeys can make tasks simpler by selecting menu items or by making all of your option selections in dialog boxes. But Apple events allow you to get behind a program's interface and communicate directly with the program. QuickKeys is the ideal tool to drive those programs. By virtue of being a cdev, QuickKeys is available anywhere; you don't have to switch to another program and run a script.

But there is much more. The hidden power of InterApplication Communication and Apple events is that they work just as well over a network as they do on a stand-alone Macintosh. Any macro that you create to perform a task on your Macintosh can just as easily be sent over the network to another Macintosh. You can retrieve information from your co-worker's Macintosh or a server Macintosh or you can send data to an idle Macintosh for processing.

Look Up an Address

Remember the macro that we created in Chapter 5 to look up an address in Dynodex (the address database with sophisticated printing capabilities)? If you have Dynodex 3.0, you can look up an address and paste it into your letter without ever leaving your word processor. You can send an Apple event to Dynodex that searches for whatever you have on the Clipboard. You can then send a second Apple event to Dynodex that copies the current record, reformats it as a label, and then pastes it into your document at the insertion point. Sound complicated? It really isn't. Let's create the two macros now so you can see just how easy this is.

Before we begin: You will need to have Dynodex 3.0 installed and running. If you have an earlier version of Dynodex, you will need to upgrade it before you can create this macro because versions earlier than 3.0 do not have Apple event support.

Create the macro:

1. Open QuickKeys and select Universal keyset from the pop-up menu. Since we will want to play this macro from a word processor, we'll save it in the Universal keyset where it is always available. As an added bonus, we will also be able to play it from other programs.
2. Select Apple Events from the Extensions menu.
3. When the Apple Events editor opens, select Dynodex 3.0 from the Send Event to pop-up menu.

4. Select Custom from the Event pop-up menu. The scrolling parameter list is added to the middle of the Apple Events editor.
5. Click the New button. When the parameter dialog box opens, enter "String to find" in the Description box (Fig. 11-16). Click the Keyword box and enter "FSTR" as the keyword. Select Text from the Parameter Type pop-up menu and then click the Get from Clipboard radio button. Click OK or press **Return** to close the parameter dialog box.

We could enter the name or some other bit of information we want Dynodex to search for in the Data text box, but that would limit the usefulness of the macro. By selecting the Clipboard, we can have Dynodex search through its current file for whatever we place on the Clipboard.

The image shows a 'Parameter' dialog box with the following fields and controls:

- Parameter Number:** 1
- Description:** String to find
- Keyword:** FSTR
- Parameter Type:** Text (selected in a dropdown menu)
- Get from:** TEXT (button)
- Radio buttons:** ☐ Data: ☒ Get from Clipboard
- Buttons:** OK, Cancel

Figure 11-16: The settings for the String to find parameter.

6. Click the New button. When the parameter dialog box opens, enter "Field to search in" in the Description text box (Fig. 11-17). Click the Keyword box and enter "FIEL" for the keyword. Select Text from the Parameter Type dialog box and then enter "All Fields" in the Data text box. Click OK or press **Return** to close the parameter dialog box. These settings cause Dynodex to search through all of the fields in the current file.

Parameter Number: 2

Description: Field to search in

Keyword: FIEL

Parameter Type: Text ▼ TEXT

☒ **Data:** ☐ Get from Clipboard

All Fields

OK Cancel

Figure 11-17: The settings for the Field to search in parameter.

7. Click the New button. When the parameter dialog box opens, enter "Whole Word" in the Description box (Fig. 11-18). Click in the Keyword box and enter "WWRD" for the keyword. Select Integer from the Parameter Type pop-up menu and then enter "1" in the Data text box. Click OK or press **Return** to close the parameter dialog box. These settings cause Dynodex to look only for a whole word. If we wanted Dynodex to look for a part of a word, we would enter a "0" in the Data text box.
8. Click the New button. When the parameter dialog box opens, enter "Match Upper/Lowercase" in the Description box (Fig. 11-19). Click the Keyword field and enter "CASE" as the keyword. Select Integer from the Parameter Type pop-up menu and then enter "0" in the Data text box. These settings tell Dynodex not to match upper- and lowercase. If we wanted to match upper- and lowercase, we would enter a "1" in the text box.
9. Now enter "DYNO" in the Class box and "FIND" in the ID box in the lower-right corner of the Apple Events editor (Fig. 11-20). Now your DynoFind macro is complete.
10. Name your QuickKey and assign a keystroke. Click OK or press **Return** to close the Apple Events editor and save your changes.

Parameter Number: 3

Description: Whole Word

Keyword: WWRD

Parameter Type: Integer ▼ shor

☒ **Data:** ☐ Get from Clipboard

1

OK Cancel

Figure 11-18: The settings for the Whole Word parameter.

Parameter Number: 4

Description: Match Upper/Lowercase

Keyword: CASE

Parameter Type: Integer ▼ shor

☒ **Data:** ☐ Get from Clipboard

0

OK Cancel

Figure 11-19: The settings for the Match Upper/Lowercase parameter.

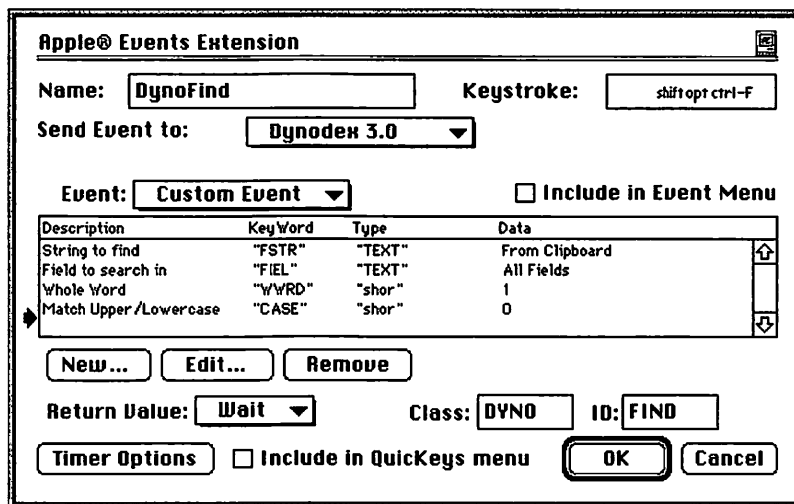


Figure 11-20: The final DynoFind macro.

You have one half of the macro pair. Now we need to create the macro that brings the address information back to your word processor.

1. Select Apple Events from the Extensions menu.
2. When the editor opens, select Dynodex 3.0 from the Send event to pop-up menu.
3. Select Lookup from Target... from the Event pop-up menu.
4. When the Events dialog box opens, select Dynodex Suite in the Events Suites list and then select Copy Label in the Events in the selected suite list. This event copies the address information from the Dynodex record just as it would appear on a label.
5. Click OK or press **Return** to close the Events dialog box and save your changes. You'll notice that this event doesn't have parameters like the previous one (Fig. 11-21).
6. Name your macro and assign a keystroke. Click OK or press **Return** to close the Apple Events editor and then click OK or press **Return** to close QuickKeys and save your changes.

Now you are ready to use your macros to look up an address in Dynodex 3.0 and paste the address into your word processor document.

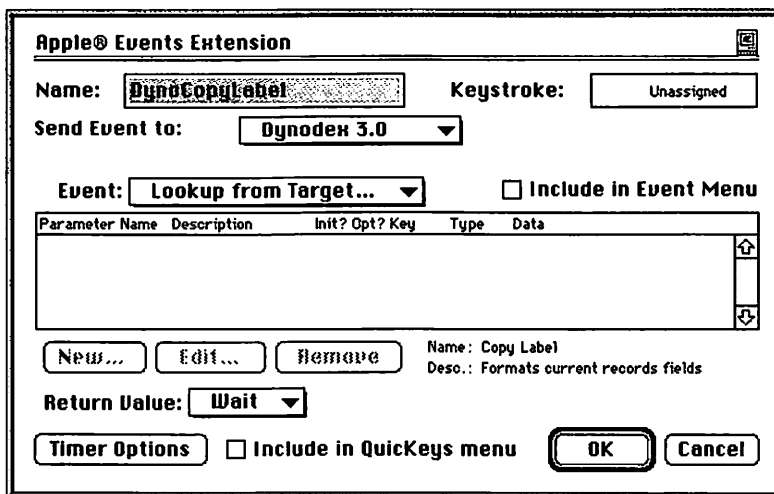


Figure 11-21: The Apple Events editor settings for the Copy Record event.

Use and variations: To use these two macros, you should have Dynodex running with the file in which you want Dynodex to search open. In the foreground, you should have your word processor loaded with a letter open. Select the name of the person for whose address you want Dynodex to search, select Copy from the Edit menu or press **⌘-C**, and then press the keystroke you've assigned to the DynoFind macro. In the background, Dynodex searches for the record that matches the contents of the Clipboard. After a moment, you can press the keystroke you've assigned to the DynoCopyLabel macro. Like magic, the person's full name and address is pasted into your letter where the name was before.

Of course, you don't have to keep Dynodex running all of the time. You could create a sequence that opens Dynodex, waits until it is completely loaded, initiates the search, and then copies the address into your letter. This sequence would use a System 7 special macro to open Dynodex and would look something like Figure 11-22.

And, as I pointed out earlier, Dynodex doesn't have to be running on your Macintosh for these macros to work. Your office or work group could use Dynodex as an address server. All of the addresses that people in the company or work group need everyday could be in a Dynodex file running on the mail or file server or another Macintosh. The Apple event would be created with the Remote Application option in the Apple Events extension. Whenever someone needs an address for a letter, they could play the DynoFind and DynoLabel QuickKeys to look up the address and paste it into their letter.

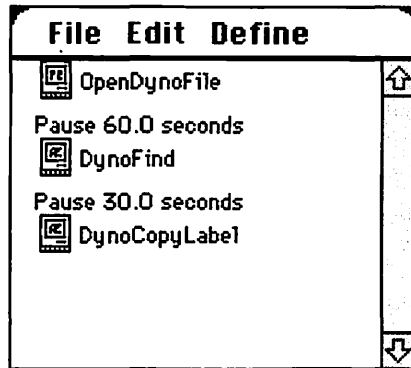


Figure 11-22: The Launch&Lookup macro.

Send Apple Events to StuffIt Lite

The popular compression program StuffIt Deluxe 3.0 and its shareware sibling StuffIt Lite support a very complete set of Apple events. Everything you can do in the programs with a mouse and a keyboard, you can also do with Apple events and without bringing the application to the foreground.

Of course, some of you may wonder why we are even bothering to create a QuickStuff macro. After all, we can use the StuffIt and UnStuffIt extensions supplied with QuickKeys and StuffIt Deluxe to stuff and unstuff files. Well, we could, but we would have to select the files we want to stuff or unstuff in the dialog box. We would also have to stop what we are doing to make the selection and wait while the files are stuffed or unstuffed. The QuickStuff macro allows us to specify the files or folders we want stuffed and then stuff them in the background without interrupting what we are doing.

Before we begin: You should have StuffIt Deluxe or StuffIt Lite installed and running. Even though StuffIt Deluxe and StuffIt Lite have an aete resource, we'll make this macro by creating custom events in the Apple Events editor. This is to avoid some potential conflicts in the way the two programs handle Apple events. While there isn't any discussion of the Apple event parameters in the StuffIt Deluxe documentation, you can get all of the information you need to create these custom events by using the Lookup from Target... command.

Create the macro:

1. Open QuickKeys and select Universal keyset from the pop-up menu. Then select Apple Events from the Extensions menu.
2. When the Apple Events editor opens, select StuffIt Lite or StuffIt Deluxe (depending on which program you are using) from the Send event to pop-up menu.
3. Select Custom event from the Event pop-up menu. A scrolling list opens in the middle of the dialog box where you can enter the Apple event parameters (Fig. 11-23).

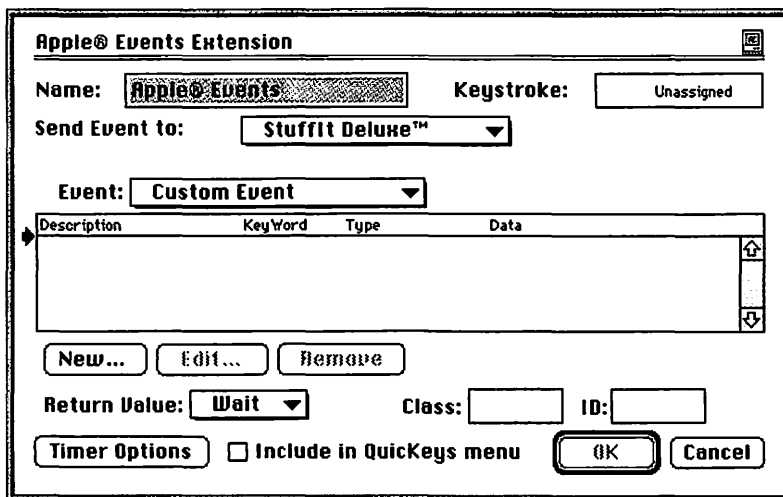


Figure 11-23: The Apple Events editor before you enter parameters.

4. Enter "SIT!" in the Class box and "QStf" in the ID box in the bottom-right portion of the Apple Events extension. Apple events are case sensitive, so be sure to enter the class and ID settings with the correct case. These settings tell CEIAC what kind of Apple event you are sending; in this case, a QuickStuff event.
5. Click the New... button to open the parameter dialog box (Fig. 11-24). Enter "pathname" in the Description box. Enter "path" in the keyword box. Select Text from the Parameter Type pop-up menu and then click the Get from Clipboard radio button. This first parameter tells StuffIt that it should get the pathname to the file or folder you want to compress from the Clipboard. Click OK to close the dialog box.

Parameter Number: 1

Description:

Keyword:

Parameter Type:

☐ Data: ☒ Get from Clipboard

Figure 11-24: The pathname parameter settings for the CreateSEA macro.

6. Click the New... button again. When the parameter dialog box opens, enter "delete original" in the Description box (Fig. 11-25). Enter "dele" in the Keyword box. Select Integer from the Parameter Type pop-up menu and enter "1" in the Data box at the bottom of the dialog. This parameter tells StuffIt that you want it to delete the original file or folder after it has stuffed it. If you decide later that you don't want StuffIt to delete the original file or folder, you can remove the parameter or enter a "0" in the text box. Click OK to close the dialog box.
7. Click the New... button and enter "make SEA" in the Description box (Fig. 11-26). Enter "sea" in the Keyword box. Select Integer from the Parameter Type pop-up menu and enter "1" in the Data box. This parameter tells StuffIt to attach the self-extractor to the archive you've just created. Click OK to close the dialog box.
8. Your completed CreateSEA macro should look like the one in Figure 11-27.
9. Name your macro and assign a keystroke. Click OK to close the Apple Events editor and click OK to close QuickKeys.

Parameter Number: 1

Description: delete original

Keyword: delete

Parameter Type: Integer (dropdown) short (checkbox)

☒ Data: ☐ Get from Clipboard

1

OK Cancel

Figure 11-25: The delete original settings for the CreateSEA macro.

Parameter Number: 1

Description: make SEA

Keyword: sea

Parameter Type: Integer (dropdown) short (checkbox)

☒ Data: ☐ Get from Clipboard

1

OK Cancel

Figure 11-26: The make SEA settings for the CreateSEA macro.

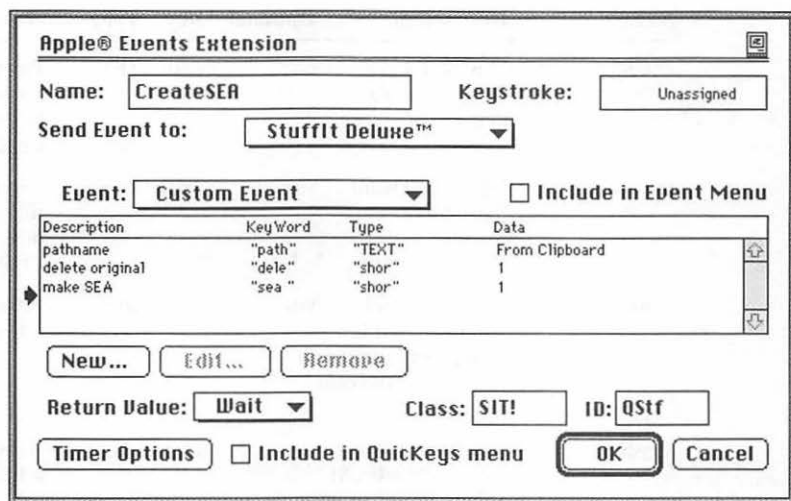


Figure 11-27: The completed CreateSEA macro.

Use and variations: To use the CreateSEA macro, you could enter the path to the file or folder (for example, hard drive:Desktop Folder:FileToStuff) in a text editor or in the Notepad desk accessory, select it, and copy it to the Clipboard. Then you would press the keystroke you've assigned to the macro. If you don't have too many windows open, you should be able to see Stuffit's status dialog box appear while the file or folder is stuffed.

In addition to the parameters we've used in this macro, the QuickStuff Apple event has several additional parameters (Table 11-1). You can enter a destination parameter so the stuffed file is saved to a different folder or disk. You would enter "dpat" in the Keyword box and select Text from the pop-up menu. You would then enter the path to the folder where you want to save the compressed file in the Data field. For example, you might enter "hard drive:Desktop Folder:Files to Upload:" making sure to separate the drive name from folder names with colons. This would save the compressed file in a folder on your desktop called "Files to Upload." Since you already have copied the path to the file you want to stuff on your Clipboard, you cannot put the destination path there too. You can also enter a rename parameter using the settings from the list in Table 11-2. This parameter would cause Stuffit to append a number to the end of the compressed file's name if it found a file with the same name. The QuickStuff Apple event also takes a version parameter that tells Stuffit to create a Stuffit 1.51 archive.

Name	Description	Optional	Key	Type	Data
Pathname	The location of the file you want to archive.	no	path	TEXT	Enter the path to the file in the Data text box or get it from the Clipboard.
Destination	Where you want the archive placed.	yes	dpat	TEXT	Enter the path to the file in the Data text box or get it from the Clipboard.
Rename	Whether or not you want StuffIt to append a number to the name of your archive.	yes	rena	shor	A 0 turns off renaming; a 1 turns it on.
Version	Causes StuffIt to create a StuffIt 1.51 archive.	yes	1.5.1	shor	A 0 turns off renaming; a 1 turns it on.
Delete Original	Causes StuffIt to delete the original file after it creates the archive.	yes	dele	shor	A 0 turns off renaming; a 1 turns it on.
Make SEA	Causes StuffIt to attach self-extracting information to the archive.	yes	sea*	shor	A 0 turns off renaming; a 1 turns it on.

*There is a space after the "a."

Table 11-1: The complete set of parameters for a QuickStuff Apple event.

Besides sending different parameters for the QuickStuff Apple event, you can create a QuickUnStuff macro. To do this, you would enter "SIT!" in the Class box and "QUSt" in the ID box. The QuickUnStuff Apple event also takes destination pathname, rename, and delete original parameters (Table 11-2). StuffIt Lite automatically appends a number to the name of a file when it finds a file with the same name. The rename parameter lets you suppress this feature when you enter a "1" in the Data box of the parameter. You will find an AEUnStuff macro on the companion disk that looks like Figure 11-28. This macro takes the pathname from the Clipboard and unstuffs the file. It appends a number if it finds another file with the same name and it deletes the compressed file when it is through.

Parameter Name	Description	Optional	Key	Type	Data
Pathname	The location of the file you want to archive.	no	path	TEXT	Enter the path to the file in the Data text box or get it from the Clipboard.
Destination	Where you want the archive placed.	yes	dpat	TEXT	Enter the path to the file in the Data text box or get it from the Clipboard.
Rename	Appends a number to the end of a file when it finds a file with the same name.	yes	rena	shor	A 0 turns off renaming; a 1 turns it on.
Delete Original	Causes Stuffit to delete the original file after it creates the archive	yes	dele	shor	A 0 turns on renaming; a 1 turns it off.

Table 11-2: The complete set of parameters for a QuickUnStuff Apple event.

Apple® Events Extension

Name: **REUnStuff** Keystroke: **opt ctrl-~**

Send Event to: **Stuffit Lite™ 3.0.1**

Event: **Custom Event** ☐ Include in Event Menu

Description	Key Word	Type	Data
pathname	"path"	"TEXT"	From Clipboard
rename	"rena"	"shor"	1
delete original	"dele"	"shor"	1

New... **Edit...** **Remove**

Return Value: **Wait** Class: **SIT!** ID: **QUSt**

☐ Timer Options ☐ Include in QuickKeys menu **OK** **Cancel**

Figure 11-28: The AEUnStuff macro.

NOTE: If you haven't registered your copy of Stuffit Lite, you will not have access to the Apple events Add Match, Clear Match, Create New Folder, Delete Selected Items, Get Archive Comment, Rename Item, Rename Selected Items, Select Archive, Set Archive Comment, Stuff Item, Translate, or UnStuff Item.

QuickKeys and Scripting Programs

I have mentioned AppleScript several times through the course of this book. It will begin an entirely new Macintosh environment, but it will not be a replacement for QuickKeys. QuickKeys is rather uniquely positioned between programs that send and receive Apple events. Most of those programs cannot directly communicate with one another, but QuickKeys can do the talking for them. AppleScript and other scripting programs have a broader ability to communicate with the system and programs that support Apple events, but you may still have to be in the program to run your scripts. QuickKeys allows you to run your scripts and scripting programs without leaving the program you are in.

UserLand Frontier

If you want an idea of AppleScript's power now, you should get a copy of Userland's Frontier. Frontier is a scripting environment that can take advantage of Apple events and let you control programs that support Apple events. Granted, its scripting language is akin to the C programming language and can be a little difficult to learn. You can accomplish tasks with Frontier that would be difficult or impossible to accomplish in QuickKeys.

So how does Frontier tie in with QuickKeys? When you pair the two, you make both programs more powerful and your macros more efficient. In the backup macro in Chapter 10, for instance, we had to use DiskTop or a comparable program to locate the most recently changed files and then copy them to another volume. QuickKeys isn't able to do this on its own. We could create a much cleaner macro by using Frontier to locate the recently changed files and copy them. It has a set of verbs built into its scripting language that makes it much easier to communicate with the Finder. Frontier even ships with a backup script. You can use the Frontier extension shipped with QuickKeys to send a script to Frontier. With it, you can use QuickKeys to invoke your backup script at a keystroke or you can have QuickKeys play the script at a specific time. You can also use CEIAC and the play by name Apple event to play a QuickKeys macro from Frontier.

ControlTower

I would be more than remiss if I didn't mention ControlTower from Simple Software. It also allows you to control programs using Apple events, and it is closer to solving the user interface problems than other programs.

You have probably realized that QuicKeys' Apple Events extension correctly reads the aete resource and forces you to enter the parameters manually in the proper order. Unfortunately, the interface is a bit arcane, and the extension truncates the text describing the Apple event. ControlTower has worked out some of these problems. It is able to show all of the Apple event description text; it walks you through creating the parameters for the Apple event; and, like Frontier, ControlTower has a connection to QuicKeys. It contains a QuicKeys Apple event suite that allows you to play macros by name.

Where to Now?

If you want to know more about Apple events, there is information available, but be warned that it is written for programmers. *Inside Macintosh, Vol. VI* has a chapter devoted to Apple events, and *InterApplication Communication*—one of the volumes in the new *Inside Macintosh* series—should contain even more information about how events are sent between programs. For information about the currently supported events, you can get the *Apple Event Registry* from the APDA (formerly the Apple Programmer's and Developer's Association). It is updated on a regular basis to include new suites of Apple events. Unfortunately, there aren't any books for the casual user who wants to use Apple events, but some should become available with the release of AppleScript.

You may have reached the end of this book, but you are at the beginning of a new era of macroing and scripting capabilities. As more developers add Apple event support to their programs, you will have more control over your Macintosh computing environment. Eventually, some of the tricks and kludges I've used to accomplish tasks with QuicKeys will be unnecessary. As you have seen in this chapter, you are able to use Apple events to query Dynodex 3.0 for an address and paste it into your word processor document without the need to switch between programs. Soon all communication between programs will be just as transparent.

So, experiment more with QuicKeys. More importantly, have fun with QuicKeys.

A

APPENDIX

Where Do I Find This Stuff?

Throughout this book I have referred to quite a bit of hardware and software. In this appendix, I have listed the names, addresses, and phone numbers of the companies that make the products I've discussed. You can buy the commercial software and hardware listed here directly from the manufacturer or publisher. However, if you purchase from one of the warehouses, you can save money. All of the warehouses have 800 numbers and most of them ship your purchases via next-day air for the same amount as most UPS ground shipments.

I've included two public domain programs, Escapade and Rebound, on the companion disk because they can make your macros more accurate. You can generally find the other shareware and public domain programs I mention on any of the electronic services. You should also check with your local user group. These groups often maintain libraries of shareware, freeware, and public domain programs that you can get for the price of a disk. The Berkeley Macintosh User's Group (BMUG) also sells disks filled with public domain and shareware programs.

Commerical Software and Hardware

Communications and Electronic Mail

America Online

List price: The software is free,
but there is a monthly charge
for the service.

8619 Westwood Center Dr.
Vienna, VA 22182
(800) 827-6364
(703) 893-6288

Apple Event Registry: Standard Suites

List price: \$85.00
Apple Programmer's and
Developer's Association
Apple Computer, Inc.
20525 Mariani Ave., M/S 33G
Cupertino, CA 95014
(800) 776-2333
(408) 996-1010

AppleLink

List price: \$70.00 (additional
monthly charges)
Apple Programmer's and
Developer's Association
Apple Computer, Inc.
20525 Mariani Ave., M/S 33G
Cupertino, CA 95014
(800) 776-2333
(408) 996-1010

AppleTalk Remote Access

List price: \$199.00
Apple Computer, Inc.
20525 Mariani Ave.
Cupertino, CA 95014
(800) 776-2333
(408) 996-1010

CompuServe Information Manager

List price: \$39.95 (additional
monthly charges)
CompuServe Information Service
5000 Arlington Centre Blvd.
Columbus, OH 43220
(800) 848-8199
(614) 457-8600

CONNECT

List price: \$150.00
CONNECT Inc.
10161 Bubba Rd.
Cupertino, CA 95014
(408) 973-0110
(800) 262-2638

Microphone II

List price: \$295.00
Software Ventures, Inc.
2907 Claremont Ave.
Berkeley, CA 94705
(510) 644-3232

Microsoft Mail

List price: \$395.00 (5 users)
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
(800) 426-9400
(206) 882-8080

Navigator

List price: \$99.95 (additional
monthly charges)
CompuServe Information Service
5000 Arlington Centre Blvd.
Columbus, OH 43220
(800) 848-8199
(614) 457-8600

PhoneNET Liason

List price: \$395.00
Farallon Computing
2000 Powell St., Ste. 600
Emeryville, CA 94608
(510) 596-9000

QuickMail

List price: \$399.00 (5 users)
CE Software
1801 Industrial Circle,
P.O. Box 65580
West Des Moines, IA 50265
(515) 224-1412
(515) 224-4534

Teleport FullFax

List price: \$295.00
Global Village Communications
1204 O'Brien Dr.
Menlo Park, CA 94025
(800) 736-4821
(415) 329-0700

VersaTerm Pro

List price: \$295.00
Synergy Software
2457 Perkiomen Ave.
Reading, PA 19606
(800) 876-8376
(215) 779-0522

White Knight

List price: \$139.00
FreeSoft Co.
105 McKinley Rd.
Beaver Falls, PA 15010
(412) 846-2700

Databases and PIMs

Dynodex

List price: \$125.00
Portfolio Systems
10062 Mille Ave., Ste. 201
Cupertino, CA 95014
(800) 729-3966
(408) 252-0420

FileMaker Pro

List price: \$299.00 (version 1.0)
List price: \$399.00 (version 2.0)
Claris Corp.
5201 Patrick Henry Dr.
Santa Clara, CA 95052
(408) 727-8227

Graphics and Page Layout

Amazing Paint

List price: \$99.95
CE Software
1801 Industrial Circle
P.O. Box 65580
West Des Moines, IA 50265
(800) 523-7638
(515) 224-1995

Aldus PageMaker

List price: \$795.00
Aldus SuperPaint
List price: \$199.00
Aldus Corp.
411 First Ave. S.
Seattle, WA 98104
(800) 333-2538
(206) 628-2320

MacDraw Pro

List price: \$399.00
Claris Corp.
5201 Patrick Henry Dr.
Santa Clara, CA 95052
(408) 727-8227

Canvas

List price: \$399.00
UltraPaint
List price: \$199.00
Deneba Software
3305 N.W. 74th Ave.
Miami, FL 33122
(800) 622-6827
(305) 594-6965

Hardware and Other Stuff

Fkey Catalog

List price: \$29.95
Multicomp, Inc.
720 Pine St., Ste. 3
P.O. Box 2761
Abilene, TX 79604
(915) 676-0844
(915) 675-5944 (fax)
(800) 541-4351 (orders)

Mac Pro Plus

List price: \$179.00
Trak Pro
List price: \$249.00
Key Tronic Corp.
P.O. Box 14687
Spokane, WA 99214
(800) 262-6006
(509) 928-8000

Mac 101E

List price: \$195.00

SwitchBoard

List price: \$239.00

Templates & 101E Dust Cover

List price: \$14.95
Data Desk International
9524 S. W. Tualatin Sherwood Rd.
Tualatin, OR 97062
(800) 328-2337
(503) 692-9600

MouseMan

List price: \$129.00
Logitech, Inc.
6505 Kaiser Dr.
Fremont, CA 94555
(800) 231-7717
(510) 795-8500

PowerKey

List price: \$119.00

PowerKey Remote

List price: \$49.00
Sophisticated Circuits
19017 120th Ave. N.E., Ste. 106
Bothell, WA 98011
(206) 485-7979

Gravis Super Mouse

List price: \$129.95
Advanced Gravis
#111 7400 MacPherson Ave.
Burnaby, B.C. V5J 5B6
Canada
(604) 431-1807

UnMouse

List price: \$199.00
MicroTouch Systems
55 Jonspin Rd.
Wilmington, MA 01887
(508) 694-9900
(508) 694-9980 (fax)

HyperTextual

HyperCard

List price: \$199.00
Claris Corp.
5201 Patrick Henry Dr.
Santa Clara, CA 95052
(408) 727-8227

SuperCard

List price: \$299.00
Aldus Corp.
411 First Ave. S.
Seattle, WA 98104
(800) 333-2538
(206) 628-2320

MasterScript

List price: \$129.00
Heizer Software
1941 Oak Park Blvd., Ste. 30
P.O. Box 232019
Pleasant Hill, CA 94523
(510) 943-7667
(800) 888-7667
(510) 943-6882 (fax)

Macroing and Scripting

ControlTower

List price: \$149.95
Simple Software
220 Redwood Highway, Ste. 42
Mill Valley, CA 94941
(415) 381-2650
(415) 381-6417 (fax)

Frontier

List price: \$199.00
UserLand Software, Inc.
400 Seaport Ct.
Redwood City, CA 94063
(415) 369-6600
(415) 369-6618

MacroMaker

List price: \$59.95 (\$5.00 upgrade
for System 6 users)
Go Technology
P.O. Box 7667
Incline Village, NV 89450
(800) 468-5391

Tempo II Plus

List price: \$169.95
Affinity Microsystems, Ltd.
1050 Walnut St., Ste. 425
Boulder, CO 80302
(800) 367-6771
(303) 442-4840

Spreadsheets

Claris Resolve

List price: \$399.00
Claris Corp.
5201 Patrick Henry
Santa Clara, CA 95052
(408) 727-8227

Microsoft Excel

List price: \$495.00
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
(800) 426-9400
(206) 882-8080

Lotus 1-2-3

List price: \$495.00
Lotus Development Corp.
55 Cambridge Pwky.
Cambridge, MA 02142
(800) 688-8320
(617) 577-8500

Utilities

AfterDark

List price: \$49.95
Berkeley Systems, Inc.
2095 Rose St.
Berkeley, CA 94709
(510) 540-5535

DiskTools Plus (part of the File Director package)

List price: \$129.00

Pyro

List price: \$39.95
Fifth Generation Systems
11200 Industriplex Blvd.
Baton Rouge, LA 70809
(504) 291-9953
(800) 873-4384

DiskTop

List price: \$99.95
CE Software
1801 Industrial Circle,
P.O. Box 65580
West Des Moines, IA 50265
(515) 224-1412
(515) 224-4534

GOfer 2.0 (bundled with CE Software's DiskTop)

List price: \$79.95
Microlytics, Inc.
Two Tobey Village Office Park
Pittsford, NY 14534
(800) 828-6293
(716) 248-9150

HandOff II

List price: \$99.00
Connectix Corp.
2655 Campus Dr.
San Mateo, CA 94403
(800) 950-5880
(415) 571-5100

MasterFinder

List price: \$149.00

MultiClip

List price: \$149.00
Olduvai Corp.
7520 Red Road, Ste. A
South Miami, FL 33143

MultiMaster (part of Now Utilities 3.0)

SuperBoomerang (part of Now Utilities 3.0)

List price: \$129.00
Now Software
319 S.W. Washington St., 11th Fl.
Portland, OR 97204
(503) 274-2800
(503) 274-0670 (fax)

On Cue II

List price: \$99.00
ICOM Simulations, Inc.
648 S. Wheeling Rd.
Wheeling, IL 60090
(800) 877-4266
(305) 665-4665

ResEdit

List price: \$29.95
Apple Programmer's and
Developer's Assoc.
Apple Computer, Inc.
20525 Mariani Ave., M/S 33G
Cupertino, CA 95014
(800) 776-2333
(408) 996-1010

Shortcut

List price: \$79.95
Stuffit Deluxe
List price: \$99.95
Aladdin Systems
165 Westridge Dr.
Watsonville, CA 95076
(408) 761-6200

Word Processors and Text Editors

FullWrite Professional

List price: \$249.00
Borland International, Inc.
1800 Green Hills Rd.
Scotts Valley, CA 95066
(800) 331-0877
(408) 438-5300

MacWrite II

List price: \$129.00
Claris Corp.
5201 Patrick Henry
Santa Clara, CA 95052
(408) 727-8227

Microsoft Word

List price: \$495.00
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
(800) 426-9400
(206) 882-8080

MORE

List price: \$395.00
Symantec Corp.
10201 Torre Ave.
Cupertino, CA 95014
(800) 441-7234
(408) 253-9600

Nisus

List price: \$395.00
QUED/M
List price: \$119.00
Nisus Software
P.O. Box 1300
107 S. Cedros
Solana Beach, CA 92075
(800) 922-2993
(619) 481-1477

Vantage

List price: \$99.95
Baseline Publishing
1770 Moriah Woods Blvd., Ste. 14
Memphis, TN 38117
(800) 926-9677
(901) 682-9676

WriteNow

List price: \$249.00
T/Maker Co.
1390 Villa St.
Mountain View, CA 94041
(415) 962-0195

WordPerfect

List price: \$495.00
WordPerfect Corp.
1555 N. Technology Way
Orem, UT 84057
(800) 451-5151
(801) 225-5000

Shareware and Public Domain software

Boomerang

\$30.00 shareware

Available on CompuServe and America Online (this shareware version of the commercial SuperBoomerang program is no longer supported).

Escapade

Available on the enclosed disk.

FlashWrite][

\$15.00 shareware

Available on CompuServe and America Online.

FreeTerm

Free (as its name implies) and often bundled with modems
Available on CompuServe and America Online.

Telnet

Free from the National Center for Supercomputing Applications
Available from most FTP archives (sumex.stanford.edu and mac.archive.umich.edu).

McSink

\$45.00 shareware (a shareware version of Vantage)
Available on CompuServe and America Online.

Rebound

Available on the enclosed disk.

Stuffit Lite 3.0

\$25.00 shareware
Available on CompuServe and America Online.

Other Addresses and Numbers

Berkeley Macintosh User's Group (BMUG)

1442A Walnut St., #62
Berkeley, CA 94709-1496
(510) 549-BMUG

MacConnection

14 Mill St.
Marlow, NH 03456
(800) 800-2222
(603) 446-7791 (fax)

MacWarehouse

P.O. Box 3013
1690 Oak St.
Lakewood, NJ 08701-3013
(800) 255-6227
(908) 905-9279 (fax)

MacZone

18005 NE 68th St., Ste. A110
Redmond, WA 98052-6716
(800) 248-9948
(206) 881-3421 (fax)

And Lastly...

If I've made any errors you'd like for me to fix in the next edition of this book, or if you have a really cool macro you want to share, drop me a line. If you can, send your mail to the Internet addresses; it comes directly to my computer. Don't have Internet access? AppleLink, America Online, and CompuServe all have gateways that allow you to send email to Internet accounts.

You can reach me at the following E-mail addresses:

- Internet: pferry@msmail.kgs.ukans.edu or pferry@cyberpunk.kgs.ukans.edu
- AOL: pferry
- CompuServe: 73057,466

B

A P P E N D I X


Getting to Know QuicKeys

If you are new to QuicKeys, this appendix is provided as a quick start document to introduce you to the components of the commercial QuicKeys package. It is also a quick reference if you need to refresh your memory about a component while you are reading *The Automatic Mac*. However, this appendix is not intended to be a substitute for the actual QuicKeys manual. For those questions where only a complete answer will do, you should always refer to your QuicKeys manual.

If you don't currently own QuicKeys and are using the demonstration version of the program provided on the enclosed utility disk, you'll notice that I refer to two programs that aren't on that disk. QK Icons and Instant QuicKeys are only available with the full version of the program. Although I haven't mentioned them in the book or in this appendix, the full version of the program also includes the ButtonAction, Choosy, Display, MenuDecision, MenuWait, Panels, ProcessSwap, Screen Ease, SoftKeys, SpeakerChanger, Stuff, Type Ease, UnStuff, Which Printer, WindowDecision, and WindowWait.

After the overview of the QuicKeys package, I discuss some limitations of QuicKeys 2.1.2 (and earlier) and show you how to import sequences that are disconnected from their keysets.

The QuicKeys Editor

You create and edit macros in the QuicKeys editor (Fig. B-1). You open the QuicKeys window by selecting QuicKeys from the Apple menu or pressing the keyboard shortcut you've assigned (by default QuicKeys is installed with the shortcut -Option-Return).

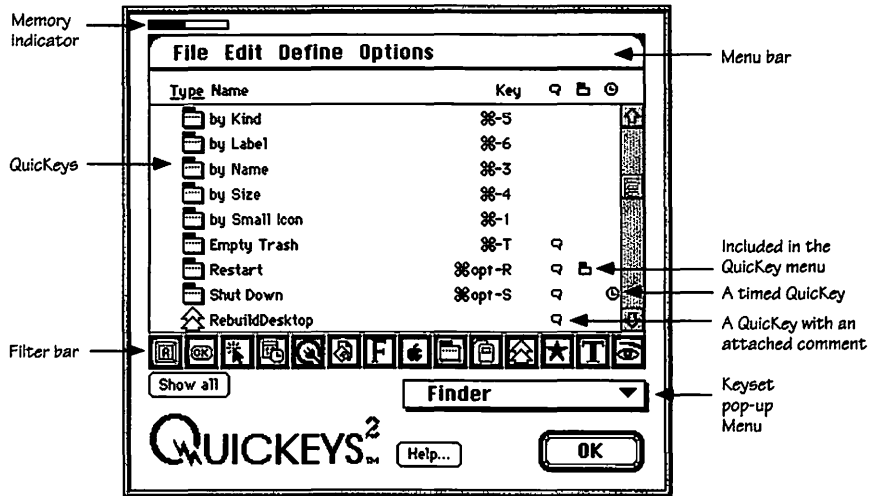


Figure B-1: The QuickKeys editor.

Let's look briefly at each part of the QuickKeys editor.

The Memory Indicator

At the top-left corner of the window is the Memory Indicator. Clicking on it opens the Statistics dialog box (Fig. B-2).

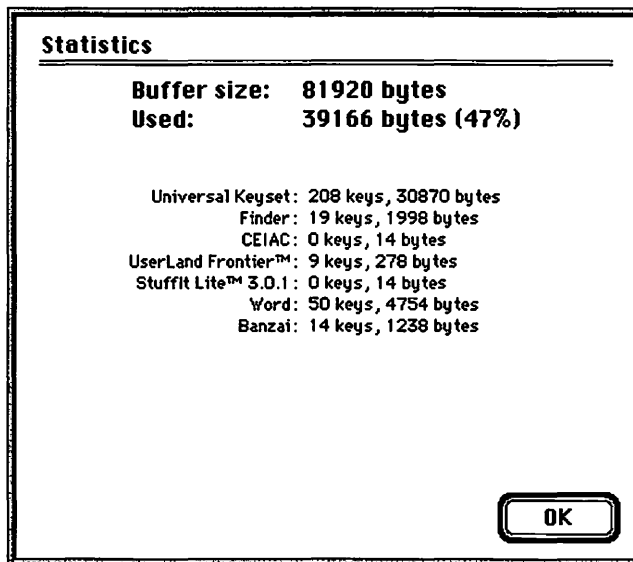


Figure B-2: The QuickKeys Statistics dialog box.

The Statistics dialog box shows the size of the memory buffer you have set for QuickKeys and the amount of the buffer that your keysets are currently using. As your keysets grow, more of the Memory Indicator bar fills (Fig. B-3), and QuickKeys may be unable to load some of your larger keysets. Whenever this occurs, QuickKeys warns you that it cannot load your keyset (Fig. B-4). As the Memory Indicator fills, you should increase the size of your buffer.

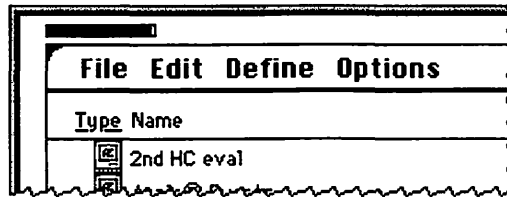


Figure B-3: As you add more QuickKeys and your keysets grow, the Memory Indicator bar fills.

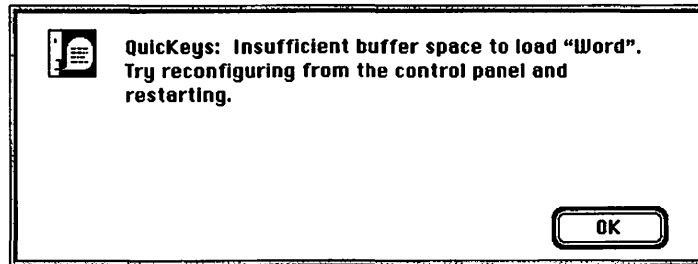


Figure B-4: As your keysets begin to fill your allocated buffer, QuickKeys has trouble loading your larger keysets and will warn you when it cannot load a keyset.

To change your buffer size, open the QuickKeys control panel and click the Configure button. This opens the QuickKeys Options dialog box (Fig. B-5).

You can change the size of your QuickKeys buffer by selecting one of the Buffer size radio buttons along the left side of the dialog box. If you aren't sure how large to make your buffer, you can use one of two methods to determine how much memory to reserve for QuickKeys.

If you rarely have more than two programs open and you really don't want to make the buffer any larger than necessary, you should add the size of your Universal keyset to the size of your largest program keyset. Use the total of those two keysets plus 10 percent for your buffer size.

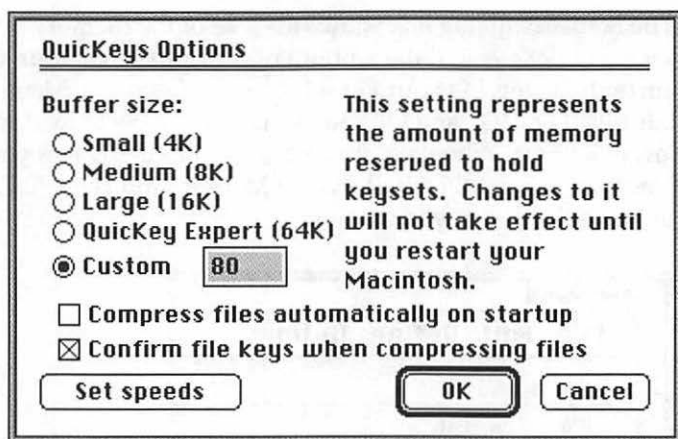


Figure B-5: The QuickKeys Options dialog box in the control panel.

On the other hand, if you have more than ten megabytes of RAM and you tend to work with several programs open at once, you should add the size of your Universal keyset to the sizes of the largest program keysets you will have open simultaneously. Use the total of those keysets plus 10 percent for your buffer size.

Of course, you can tweak the buffer size a little at a time until you find the settings that are right for your configuration. If you do not create new QuickKeys on a regular basis, setting the buffer size is less of an issue than if you are routinely creating new QuickKeys. If you create new QuickKeys often, you may want to set a larger buffer.

QuickKeys Menus

The QuickKeys menus are below the Memory Indicator bar. I will only point out features unique to QuickKeys in this section.

The File Menu

The File menu (Fig. B-6) is similar to most programs. The commands in the File menu allow you to manipulate your QuickKeys keysets. The Open command allows you to open keysets. When you select the Open command, the open dialog box automatically defaults to the Keysets folder inside of the QuickKeys Folder. However, you can open any keyset you have saved to disk using the Save A Copy or Save Selection commands. The Close command is only active when you open a keyset with the Open command. You cannot close a keyset that QuickKeys has opened.

File	
New Keyset...	
Open...	⌘O
Close	⌘W
Save	⌘S
Save a Copy...	
Save Selection...	
Page Setup...	
Print...	⌘P
Quit	⌘Q

Figure B-6: QuickKeys File menu.

The Save a Copy and Save Selection commands also work as you would expect them to. The Save a Copy command allows you to save the current keyset. This saves all of the QuickKeys in the keyset; that is, it is an exact copy. The Save Selection command allows you to save a copy of any individual QuickKeys you have selected.

NOTE: While QuickKey sequences appear in your program keysets, they are actually stored in a separate Sequences folder. If you open the Keysets folder and copy a keyset using the Finder or a utility like DiskTop, you will not get the sequence files associated with that keyset. The Save a Copy and Save Selection commands pack the sequences inside the keyset so you can pass your keysets to a friend.

The Define Menu

The Define menu (Fig. B-7) contains the commands for creating all of your QuickKeys. Since I've discussed most of these at various points in the book, I will refer here to each of those places as I discuss the different QuickKeys.

Alias QuickKeys allow you to substitute one keystroke for another in an application. For instance, you could use an alias to press **⌘-X** whenever you pressed **F3**. For a complete discussion of using alias QuickKeys, refer to Chapter 3.



Figure B-7: The QuickKeys Define menu.

Button QuickKeys allow you to assign keystrokes to buttons, radio buttons, and checkboxes in an application. You could create a button QuickKey, for example, to select the Color/Grayscale checkbox whenever you press a certain keystroke. I discuss using button QuickKeys in Chapter 8.

A click QuickKey lets you assign keystrokes for clicking and dragging objects. You could create a click QuickKey that drags the file under the cursor into the trash. I discuss using click QuickKeys to move windows in Chapter 3.

Date/time QuickKeys enter the current date and time. Date/time QuickKeys are commonly used for entering the date in a letter or memo or for time-stamping documents. I discuss using the date/time QuickKey in Chapter 3.

Extensions are mini-programs that add more power and flexibility to QuickKeys. QuickKeys ships with a number of Extensions for—among other things—mounting hard drives, repeating steps in sequences, and playing sounds. You create a QuickKey using an extension just as you would any other QuickKey. When you select an extension from the hierarchical Extensions menu, an editor opens where you can configure the settings for the QuickKey. All of the Extensions in Figure B-8 are discussed in this book. You can refer to the index for specific references.

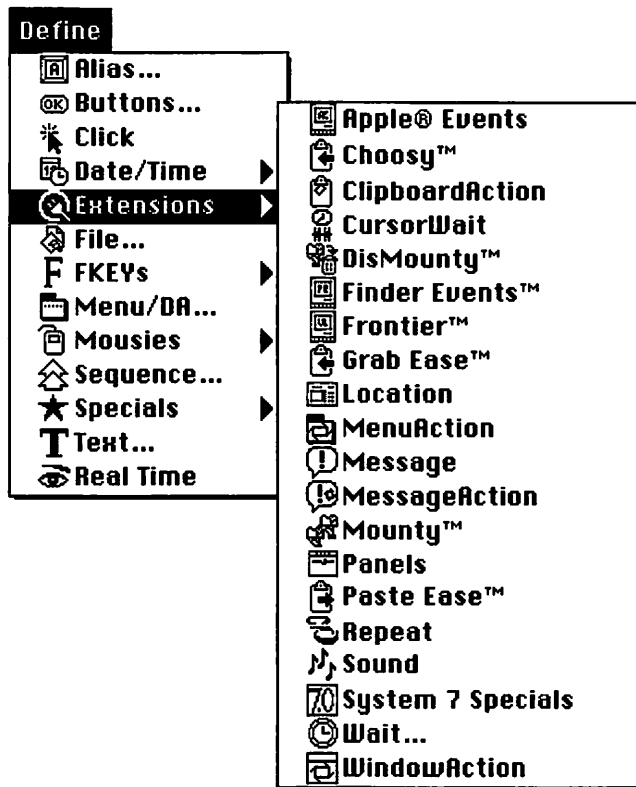


Figure B-8: The QuickKeys Extensions hierarchical menu.

File QuickKeys let you launch an application or open a file with a single keystroke—even when that application or file is buried several layers deep in folders. I mention file QuickKeys in Chapter 6.

I have not discussed FKey QuickKeys in any of the chapters. They let you assign a keystroke to any FKeys installed in your system or incorporate FKeys into sequences. When you select an FKey from the hierarchical FKeys menu, an editor opens (Fig. B-9), and you can change the FKey you've selected or assign a keystroke.

Menu/DA QuickKeys allow you to assign any keystroke to any menu item or desk accessory. For example, you can assign a keystroke to a menu item that does not already have one. You should notice that the filter bar has separate icons for menu and DA QuickKeys, though they are the same kind of QuickKey. The icons allow you to look at your DA and menu QuickKeys separately. I discuss menu/DA QuickKeys in Chapter 4.

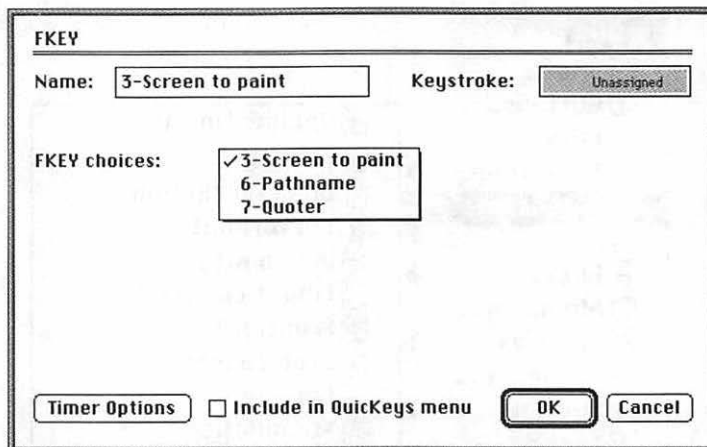


Figure B-9: The FKey editor.

Mousies are mouse actions for adjusting or viewing the contents of windows. For instance, you can use the mousies to assign Page Up and Page Down functions in programs that don't already have them. I refer to mousies in Chapter 3.

Sequences are the most powerful feature of QuicKeys. Sequences let you combine individual QuicKeys into a series that can perform any number of actions and be invoked with a single keystroke. Most of the macros in this book are sequences, but the first discussion of sequences is in Chapter 3.

Specials are a mixed bag of functions. There are specials for controlling QuicKeys, selecting windows, adding typographer's quotes to text, transferring between applications, shutting down, and restarting your Macintosh. For a discussion of specials, refer to Chapter 3.

Text QuicKeys let you assign up to 255 characters to a keystroke. The most common use of text QuicKeys is for entering text that you have to enter every day, like your home address. I discuss text QuicKeys in Chapter 4.

Real-time QuicKeys record your exact movements. During playback, a real-time QuicKey performs your movements at the speed you made them. Sequences, on the other hand, play back as fast as your Macintosh can go. You'll find a discussion of real-time QuicKeys in Chapter 6.

In addition to these QuicKeys, you will also find the pause QuicKey on the bottom of the Define menu of the Sequence editor. The pause

QuickKey has two settings: It either pauses a sequence for a specified period of time or pauses and waits for the user to resume playing the sequence. A timed pause waits a specified number of seconds before continuing to play the sequence. You can interrupt a timed pause with a click of the mouse. A user pause temporarily stops the playback of the sequence. The Apple icon in the menu bar is replaced by two flashing bars (like the pause symbol on a CD player or a tape player). The macro doesn't resume until the user selects Pause or Cancel from the QuickKeys menu. I discuss how to use the pause QuickKey in Chapter 4.

The Options Menu

The Sort by Type, Sort by Name, and Sort by Key commands allow you to reorder the listing of QuickKeys (Fig. B-10). They correspond to the Type, Name, and Key headings in the bar between the Menu bar and the scrolling list of QuickKeys.

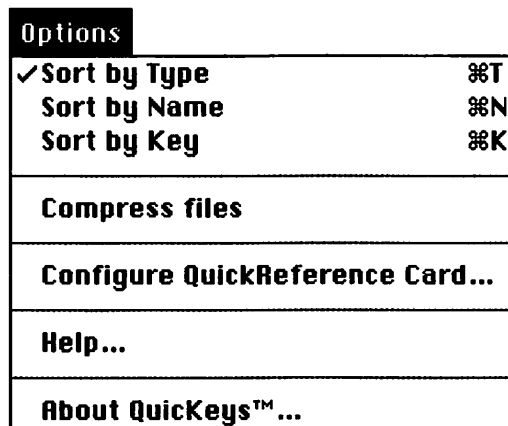


Figure B-10: QuickKeys Options menu.

The Compress Files command compresses your sequences and real-time QuickKeys. In other words, the command deletes any sequence or real-time files for which QuickKeys cannot find a corresponding QuickKey in a keyset. If you delete a sequence or real-time QuickKey, it is not deleted until the keyset is compressed. The command allows you to reclaim some disk space. You can select the Compress files automatically on startup checkbox in the QuickKeys Options dialog box (Fig. B-5) if you want QuickKeys to do this for you automatically. However, I recommend that you don't select the Compress files checkbox. If you delete a sequence and decide later that you want it back, QuickKeys may already have deleted it. (For more information about recovering sequences, see

the “Importing Sequences” section later in this appendix.) You should compress your files on a regular basis, but only do so when you know you won’t lose something you might need later.

The Configure the QuickReference card... command opens a dialog box (Fig. B-11) that lets you configure the appearance of the QuickReference card. You can change how the QuickKeys are sorted and which QuickKeys are displayed on the QuickReference Card after it opens; the settings in this dialog box determine what it looks like when it first opens. If you choose to list only the application or universal QuickKeys in the dialog box, you won’t be able to show the other set of QuickKeys after the QuickReference card opens. You also won’t be able to reduce the size of the QuickReference card after you open it if you have selected the Expand to fill screen option.

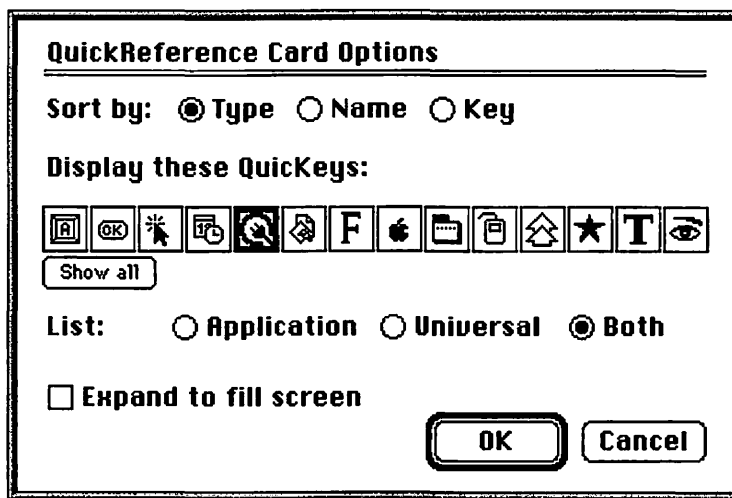


Figure B-11: The QuickKeys Options dialog box.

The QuickKeys Listing

Below the menu bar is a scrolling list of the QuickKeys in the open keyset. You can reorder the listing by clicking the Type, Name, and Key headings.

The menu, clock, and balloon icons show which QuickKeys are included in the QuickKeys menu, have Timer options set, or have comments attached. You can add a QuickKey to the QuickKeys menu by clicking beneath the menu and next to the QuickKey. A small menu icon appears next to the QuickKey (Fig. B-1). You can set Timer options by clicking

beneath the clock icon. A dialog box opens, and you can specify when QuicKeys automatically plays the macro (refer to Chapter 3 for more information). You can attach a 255-character comment to a QuicKey by clicking beneath the balloon icon. When you do, a dialog box opens where you can enter your comment. If you ever have trouble remembering what a sequence or QuicKey is supposed to do, you can attach a comment so you'll never forget. You can read any attached comments by clicking on the balloon icon next to the QuicKey.

You can change keysets by selecting a different keyset from the pop-up menu at the bottom of the QuicKeys window (Fig. B-12). The Keysets menu contains all of the currently open keysets.



Figure B-12: The Keysets pop-up menu.

You can change the QuicKeys displayed in the scrolling list by selecting different combinations of icons in the Filter bar. Each icon represents a different type of QuicKey and corresponds to the icons in the Type column of the listing. The Show All button beneath the Filter bar lets you display all of the QuicKeys in a keyset.

QK Icons

QK Icons is a small program that began shipping with QuicKeys 2.1.2. It allows you to turn a QuicKey into a mini-program. Once you make a QK Icon from one of your QuicKeys, you can play it by double-clicking the icon.

You create a QK Icon by running the QK Icons program. It reads your current keysets and displays them in a scrolling window (Fig. B-13). You select a QuicKey and click the Create button.



Figure B-13: The QK Icons window.

QK Icons displays a dialog box where you can give your QK Icon a name (Fig. B-14) and click the Save button.

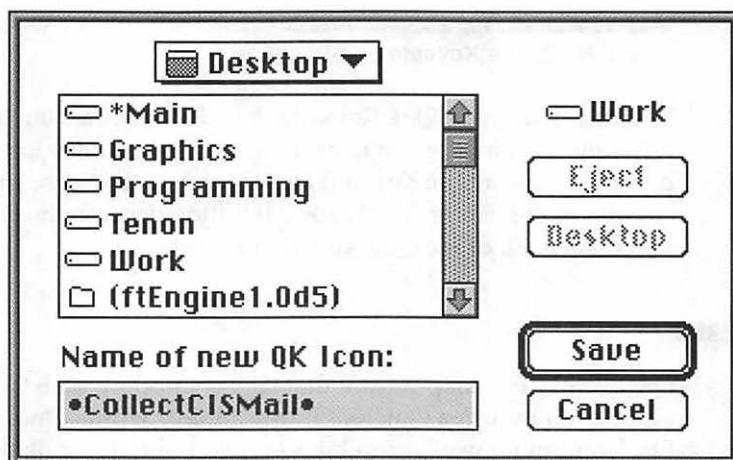


Figure B-14: Naming a QK Icon.

The result is a double-clickable icon on your desktop (Fig. B-15) that plays a macro as if you had pressed a keystroke.



Figure B-15: A sequence saved as a QK icon.

Instant QuicKeys

Another part of the QuicKeys package that began shipping with QuicKeys 2.1.2 is Instant QuicKeys. The latest version, Instant QuicKeys 2.0, began shipping with QuicKeys 2.1.3. Instant QuicKeys is a program that helps you create a number of QuicKeys very quickly. It walks you through creating a collection of very basic QuicKeys and then helps you group them using the SoftKeys extension (SoftKeys is a QuicKeys extension that lets you display your QuicKeys on a palette).

The invaluable portion of Instant QuicKeys is the Keystroke Assigner. It adds a graphical interface to keystroke assignment and, as a result, makes it easier to manage your QuicKeys.

The Keystroke Assigner displays a layout of your keyboard and a scrolling list of your QuicKeys. Keys that already have QuicKeys assigned to them are marked with a "Q" or the QuicKeys icon. As you move your pointer over the different keys on the layout, you notice that the key appears in the Keystroke box at the top of the dialog box. If a shortcut has already been assigned to the key, the shortcut's name appears in the QuicKey Name box. As you hold down any combination of modifier keys (Shift, Control, Command, or Option) and move the pointer over the keys in the layout, the information changes in the Keystroke and QuicKey Name boxes.

If you want to see all of the shortcuts associated with a particular key, you can click on the key in the layout. A pop-up list opens that contains all of the shortcuts assigned to the key and their keystrokes (Fig. B-16).

If you want to assign a keystroke to a QuicKey, you only have to drag the QuicKey from the scrolling list to that key. When you release the QuicKey on the key, a dialog box asks you which modifier keys you want to use with the keystroke. If you want to change a keystroke assignment, you drag the QuicKey from its current keystroke to the new keystroke. Did you make a mistake? Then you simply drag the QuicKey into the Trash; its keystroke is unassigned.

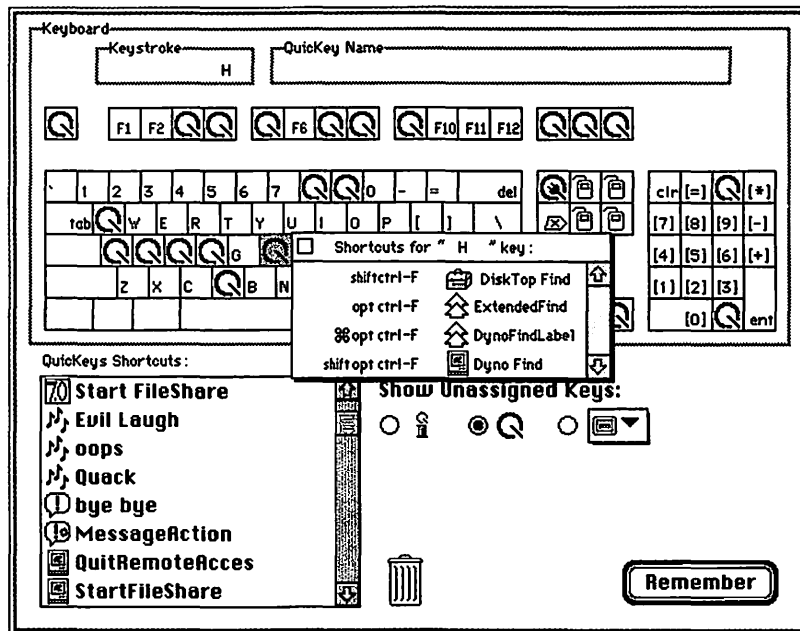


Figure B-16: The Keystroke Assigner displays a pop-up list of shortcuts assted with a particular key.

If you've ever had trouble conceptualizing how your QuickKeys are grouped on your keyboard, the Keystroke Assigner should eliminate that difficulty.

Some Limitations

There are some known limitations in QuickKeys 2.1.x and earlier that you should be aware of. These are not fatal problems, but they could be confusing if you don't know about them.

Ten Keyset Limit

QuickKeys has a self-imposed limit that allows only ten keysets to be open at once. When Macintoshes had a limited amount of memory, no one bumped into this limit. Now that the price of memory has come down and virtual memory is built into System 7, you may encounter this limit. CE Software doesn't plan to remove this limitation until the next full revision of QuickKeys.

Don't Give Keysets the Same Name

QuicKeys keeps track of keysets in the pop-up menu by name. If you open two keysets with the same name, you will only be able to see the individual QuicKeys in one keyset. You will probably encounter this phenomenon when you open a keyset you saved to disk to copy individual QuicKeys from it—for example, when you have made a backup copy of your keyset. If you can, rename the keyset from which you are copying the QuicKeys. If you don't want to rename the keyset, open the keyset, select and copy the keys, and then close QuicKeys. Now reopen QuicKeys and you'll be able to paste the individual QuicKeys you copied from the keyset. CE Software doesn't plan to correct this problem until the next full revision of QuicKeys.

Limit Keysets to 32K

QuicKeys 2.1.2 and earlier versions have trouble with keysets that are larger than 32K. If some of the sequences in the keyset suddenly lose their QuicKeys or you have more crashes than usual, you may be experiencing the 32K limit. You can find out how large a keyset is by clicking on the Memory Indicator in the upper-left corner of the QuicKeys editor (Fig. B-1) to open the Statistics dialog box (Fig. B-2). Chances are only your Universal keyset is this large, but any keyset with a large number of sequences could exceed 32K. If you have an earlier version of QuicKeys and your keyset is larger than 32K, you should remove some QuicKeys or sequences or get a later version of QuicKeys. QuicKeys 2.1.3 (which should be available by the time this book is in your hands) removes this 32K limit.

Importing Sequences

Have you ever deleted a sequence from your keyset that you thought had outlived its usefulness and then decided you wanted it back? Or have you ever noticed that a sequence had mysteriously lost all of its QuicKeys? If you've ever been in either situation, you probably thought you didn't have any choice but to rebuild the sequence. You don't have to. You can import the QuicKeys from that sequence into a new sequence.

Sequence files aren't deleted until you select the Compress files command. If you have selected the Compress files automatically on startup checkbox in the QuicKeys Options dialog box (Fig. B-5), restarting your Macintosh compresses the files. If you haven't compressed your files, you can use the Import button and the Other... command to open your lost or disconnected sequence.

To import deleted or disconnected sequences:

1. Open QuickKeys and select Sequence from the Define menu.
2. When the Sequence editor opens, click the Import button. The group of buttons on the right side of the editor is replaced by a scrolling list of QuickKeys and a pop-up menu (Fig. B-17).

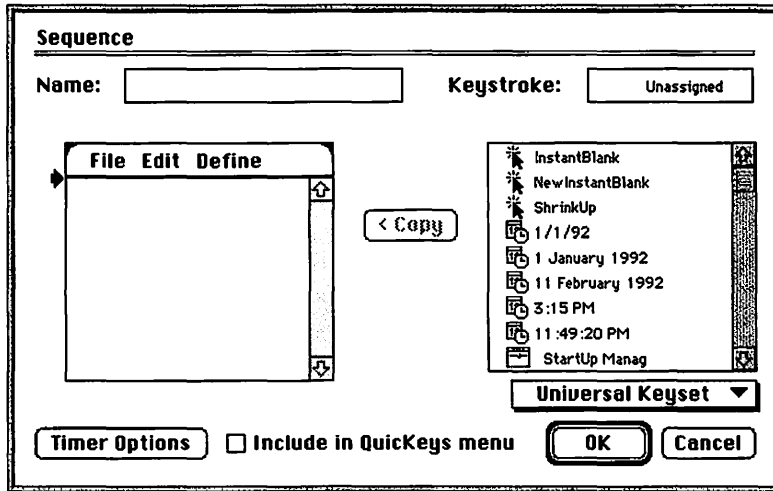


Figure B-17: When you click the Import button in the Sequence editor, a scrolling list of your QuickKeys replaces the Record More, Import, and Literal buttons.

3. Click the Pop-up menu. You'll see it contains your currently open keysets as well as Other... and None commands (Fig. B-18). Select the Other... command. A dialog opens to your Keysets folder.
4. Select QuickKeys Folder from the pop-up menu above the scrolling list.
5. Select the Sequences folder and click the Open button (Fig. B-19).
6. When the Sequences folder opens, you'll see a list of your sequence names with numbers after them (Fig. B-20). Each of these is a different version of a sequence. Whenever you edit a sequence, QuickKeys makes a copy of it and adds a number to the end of the name.

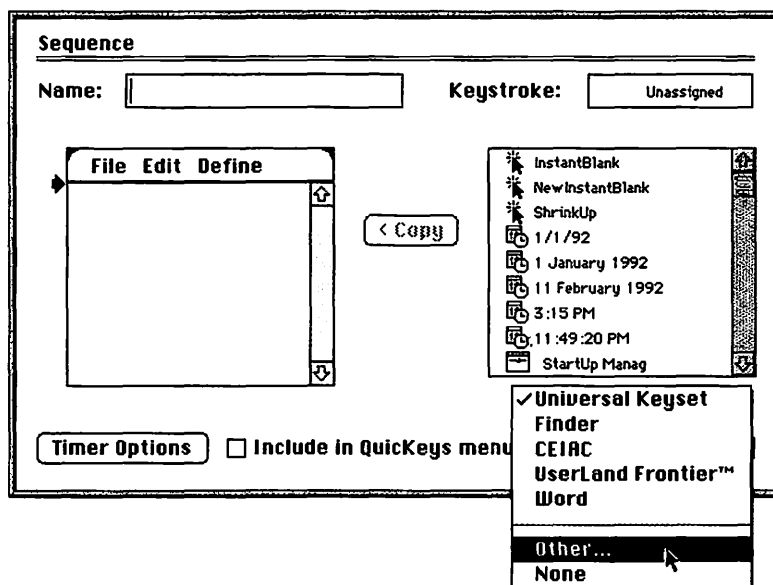


Figure B-18: Select the Other... command from the pop-up menu.

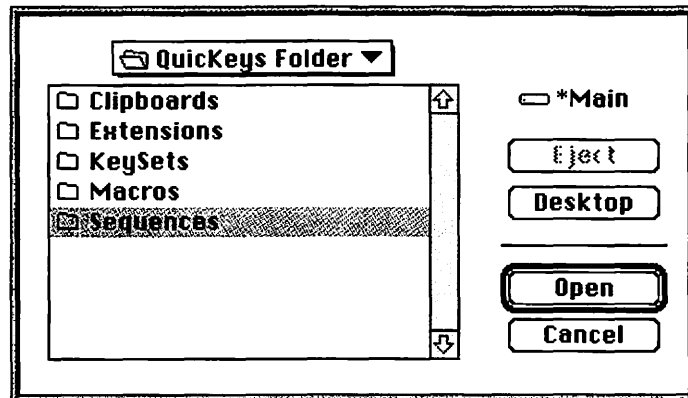


Figure B-19: The folders in your QuickKeys folder.

7. Scroll until you see the sequence you want to recover. Select it and then click the Open button. The scrolling list on the right side of the dialog box is replaced by the QuickKeys in your sequence (Fig. B-21). Notice also that the Sequence name appears in the pop-up menu.
8. Select the QuickKeys in the sequence and click the copy button (Fig. B-22).

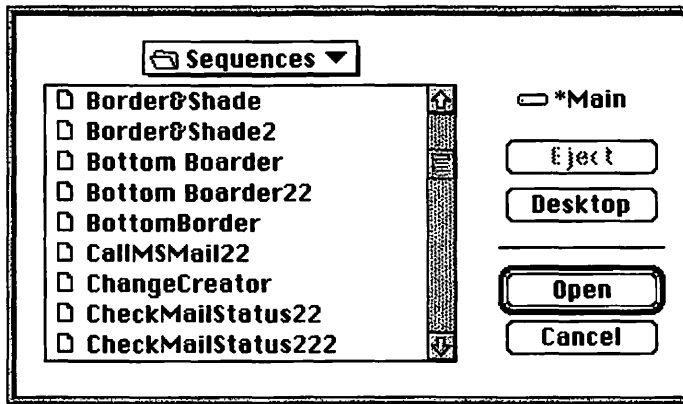


Figure B-20: You can see the different versions of sequences you have edited in your Sequences folder.

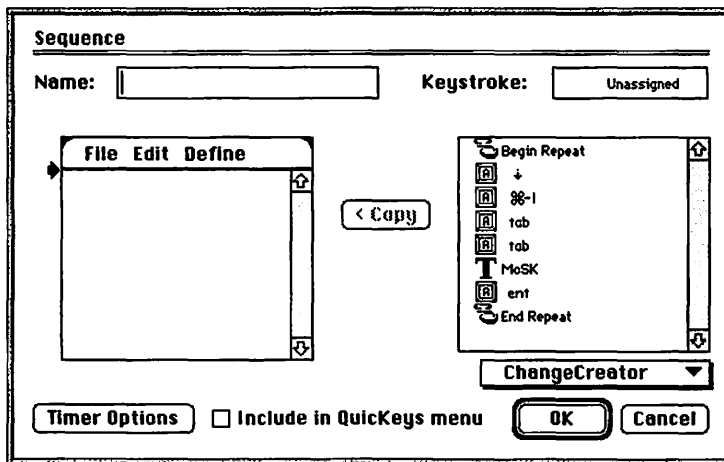


Figure B-21: An imported sequence as it appears in the Sequence editor.

9. Select None from the bottom of the pop-up menu on the right to close the sequence you are recovering.
10. Name your sequence and assign a keystroke. Then click the OK button to close the Sequence editor and click OK again to close QuickKeys and save your changes.

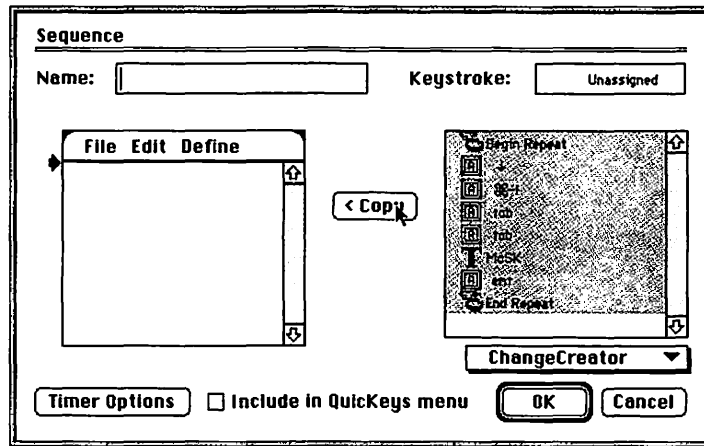


Figure B-22: Importing the QuicKeys from the sequence.

Now you can select the Compress files command and you won't have to worry that QuicKeys is deleting the sequence files you want to keep. If you are having trouble with disconnected sequences and you are using QuicKeys 2.1.2 or earlier, you should get version 2.1.3 from CE Software.

C

APPENDIX

Macros and the Disk

This appendix contains a listing of macros discussed in *The Automatic Mac* as well as a listing of the contents of the companion disk. I have listed the macros by chapter to make it a little more convenient for you to locate them, but you will find exact page numbers for each of the macros by referring to the index.

Macros Listed by Chapter

You will find that I have discussed over 100 macros in this book, but you should also be aware that the keysets on the disk contain even more macros. In some cases, I have explained how to create a macro for a specific application, like Microsoft Word, in the book. If it is appropriate, you will also find the same macro on disk for other word processing programs. In other instances, I have referred to a macro in the text that is very similar to one I have just explained how to create; you will also find most of those macros on the disk. A few of the programs listed here, such as Pyro and After Dark, are not represented by a keyset on the disk. In all, the disk contains over 600 macros in the various keysets.

Location	Macro Name	Program Name
Chapter 3	Menu QuicKey	PageMaker
	Alias QuicKeys	MORE
	HideMeFPD	any program
	DragHome	any program
	DragRight	any program
	Tile2Windows	any program
	InstantScreenSaver	Pyro & AfterDark
	ArchivingCuts	Word

Table C-1: Listing of macros discussed in this book.

Location	Macro Name	Program Name
	TransposeCharacter	Word, any word processor
	TransposeWord	Word, any word processor
	CompletePrint	any program
	CheckThenPrint	any program with spell checker
	StartMeUp	any program
	SaveAll	Word
	SaveACopy	PageMaker
	TimeLogger	TeachText & FlashWrite II
	ChangeCreator	DiskTop, ResEdit
	ExtendedFind	DiskTop
Chapter 4	2SpacesTo1	Word
	InsertMarker	Word
	FindMarker	Word
	FindMarkInFiles	Word
	GoToPage	Word
	CombineFiles	Word
	Combine&Format	Word
	GraphicList	Word
	InsertGraphic	Word
	FormatAWord	Word
	Bold1stLine	Word
	Head1	WriteNow, Word
	LetterTemplate	Word
	BibTemplate	Word
	II/NTEnvelope	Word
	DynoPrintOne	Word
Chapter 5	SplitHorizontal	Lotus 1-2-3
	ReturnAlias	Excel, Lotus 1-2-3, Resolve
	Shift-Enter	Excel, Lotus 1-2-3, Resolve
	Magnify200%	Excel, Lotus 1-2-3, Resolve
	Reduce25%	Resolve
	ExcelCut	Excel
	RightJustify	Excel
	IncreaseSize	Excel
	AutoSUM	Lotus 1-2-3, Resolve
	ConfiguredSUM	Lotus 1-2-3, Resolve
	AreaOfaCircle	Excel
	VolOfaCylinder	Excel
	FormatLabels	Excel
	Border&Shade	Excel

Table C-1: Continued.

Location	Macro Name	Program Name
Chapter 6	OutlineBorder	Resolve
	FormatWorksheet	Excel
	CreateChart	Excel
	CheckTotal	Excel
	Selecting Tools	
	with a click	Canvas
	with a RealTime	Canvas
	with AppleEvents	Canvas
	ToggleRulers	MacDraw Pro
	Reduce & Magnify	
	with Real Time	UltraPaint
	with AppleEvents	Canvas
	IntelligentTools	
	with Real Time	Canvas
	with AppleEvents	Canvas
	WrapText	PageMaker
	TightWrap	PageMaker
	AutomatedPlace	PageMaker
Chapter 7	ButtonInfo	HyperCard
	FindWhole	HyperCard
	ClickMarquee	HyperCard
	SelectMarquee	HyperCard
	Previous	HyperCard
	Next	HyperCard
	OpenHome	HyperCard
	OnMouseUp	HyperCard
	FunctionTemp	HyperCard, MasterScript
	Cut2Bkgnd	HyperCard
	HC2Scrap	HyperCard
	SetUserLevel	HyperCard
	ToggleUserLevel	HyperCard
	SetLevelTo5	HyperCard
	SetUpButton	HyperCard
	SetUpField	HyperCard
Chapter 8	CompressMail	VersaTerm or Microphone II
	VAXLogin	VersaTerm or Microphone II
	AutoConnect	VersaTerm or Microphone II
	Down18	Microphone II or Telnet
	Bottom	Microphone II or Telnet

Table C-1: Continued.

Location	Macro Name	Program Name
	CallMSMail	AppleTalk Remote Access/ MS Mail/QuickMail
	RemoteBackUp	AppleTalk Remote Access/ DiskTop
	HangUpOffice Get&SendMail	AppleTalk Remote Access CompuServe Information Manager
	SetUpReplyWin	CompuServe Information Manager or AOL
	GetMail1 & 2*	AOL
	NewMail*	AOL
	SignOff*	AOL
	LogonAOL*	AOL
	New MUT Files	AOL
Chapter 9	TogglePartLabel	Filemaker Pro
	TogglePercent	Filemaker Pro
	ToggleStatus	Filemaker Pro
	ZoomIn	Filemaker Pro
	ZoomOut	Filemaker Pro
	SetUpField	Filemaker Pro
	SetUpNumber	Filemaker Pro
	GoToFirst	Filemaker Pro
	GoToLast	Filemaker Pro
	JumpToRecord	Filemaker Pro
	AlignLeftEdges	Filemaker Pro
	AutoAddress	Filemaker Pro
	SetUpSearch	Filemaker Pro
	SetUp&Alert	Filemaker Pro
	AutoExport	Filemaker Pro
	AutoImport	Filemaker Pro
Chapter 10	MenuQuickKeys	Finder
	Trash&ShutDown	Finder
	EmptyTrash	Finder
	NightPrint	Finder/any program
	StartUpSharing	Finder
	SharingSetUp	Finder
	SetPrivileges	Finder
	Rebuilding the Desktop	Finder
	Quitting the Finder	Finder

*A suite of macros that work together.

Table C-1: Continued.

Location	Macro Name	Program Name
Chapter 11	ShowingFolders	Finder
	BackingUpFiles	Finder & DiskTop
	IncrementalBackUp	Finder & DiskTop
	BackUpModified	Finder & DiskTop
	StartSharing	Finder, Sharing Setup
	SharingSequence	Finder, Sharing Setup
	CreatePublisher	Excel 4.0
	DynoFind	Dynodex 3.0
	DynoCopyRec	Dynodex 3.0
	Launch&Lookup	Dynodex 3.0
	CreateSEA	StuffIt Lite or Deluxe 3.0
	AEUnStuff	StuffIt Lite or Deluxe 3.0

Table C-1: Continued.

Contents of the Disk

The disk contains a self-extracting archive and several folders. Below I briefly explain how to install the various components on the disk.

Demo QK2.sea

This archive contains the 30-day demonstration version of QuicKeys. If you do not own QuicKeys, you should copy this file to your hard disk and double-click it to extract the DemoQKInstall program. When you run DemoQKInstall, it installs a demo QuicKeys, CEIAC, CEToolbox, and the extensions mentioned in this book. If you are installing the demo QuicKeys, you will need to copy the macros from the Macros folder using the instructions in the "If you already own QuicKeys..." section of Chapter 1.

Macros Folder

The Keysets folder contains most of the macros described in this book. If you already own QuicKeys, you will need to copy the macros using the instructions in Chapter 1.

Extensions Folder

The Extensions folder contains the extensions Simeon Leifer wrote exclusively for *The Automatic Mac*. The four extensions are loaded automatically by the demo QuicKeys. If you already own QuicKeys, you

should use the Extensions Manager program, which came with your copy of QuickKeys, to install the extensions. For more information about installing extensions, refer to your QuickKeys manual.

Rebound 1.0

This folder contains Fred Reed's Rebound control panel. This freeware control panel adds a bounce-back feature in Open and Save dialog boxes. Fred Reed was kind enough to rewrite Rebound to work with System 7 and allow us to include it with this book. To install Rebound, drag it to your System folder. If you are running System 7, you should install Rebound in the Control Panels folder inside your System folder.

Escapade 1.3.2

This folder contains Christopher Wysocki's Escapade control panel. Among other things, this freeware control panel allows you to copy from and paste into the text fields of dialog boxes. It also allows you to select buttons in dialog boxes from your keyboard. Christopher Wysocki has graciously allowed us to include Escapade on the disk. To install Escapade, drag it to your System folder. If you are running System 7, you should install Escapade in the Control Panels folder inside your System folder.

Index

Symbols

2SpacesTo1 QuickKey, 87-91
2xMag QuickKey, 183

A

Advanced Gravis mouse, 26
AEUnStuff QuickKey, 388-389
AfterDark screen saver program, 51-52, 398
Aldus PageMaker, *see* PageMaker
alias QuickKeys, 38-40, 407
aligning database objects, 301-302
AlignLeftEdges QuickKey, 302
alignment commands, 141-142
Alignment dialog box (FileMaker Pro), 301
Alignment window (Microsoft Excel), 142
Amazing Paint, 395
America Online, 265, 394
 electronic mail
 collecting, 265-268
 saving, 268
FlashSession feature, 274
forums, opening, 274-276

Go To Keyword window, 275
launching, 269-271
MacIntosh Utilities forum window, 276
New Mail window, 272
 shutting down Macintosh, 282
Apple Event Registry, 363
Apple Events, 363-364
 and Control Tower, 390-391
 Frontier scripting environment, 390
 parameters, 364
 QuickStuff, 384-388
 QuickUnStuff, 388-389
 Required suite, 363
 stuffing files/folders, 383-390
Apple Events extension, 10
 EmptyTrash QuickKey, 323
 Event menu, 367
 Return Value menu, 370
 Send Event to menu, 366
 tools
 selecting, 172-177
 setting up, 186-189
Apple Talk Remote Access, 394
 backing up files, 250-257
 disconnecting, 257-259
 file sharing, 373-374

AppleLink, 394

AppleScript program, 7, 389-390

archiving cuts, 53-54

AreaOfACircle QuickKey, 146-150

AutoAddress QuickKey, 303-304

AutoConnect QuickKey, 238-239

AutoExport QuickKey, 310-313

AutoImport QuickKey, 313-314

AutoSUM QuickKey, 145-146

B

backing up files, 72-76, 347-357

- incremental backups, 353-355

- modified files, 357-360

- remotely, 250-257

- to AppleShare server, 355-357

BackUpMods QuickKey, 357-358

BackUpProject QuickKey, 347-353

batch files, 1

Berkeley Macintosh User's Group
(BMUG), 400

bibliography template, creating, 122-125

BibTemplate QuickKey, 122-125

BMUG (Berkeley Macintosh User's
Group), 400

Bold1stLine QuickKey, 116-117

boldface, applying, 116-117

Boomerang shareware program, 400

Border&Shade QuickKey, 154-155

business letter template, creating, 120-122

Button Info... dialog box (HyperCard),
224

button QuickKeys, 408

buttons, HyperCard

- configuring, 224-225

- moving to background, 216-218

C

CANV class code, 174

Canvas, 396

- external tools, 176

- Foreground Color palette, 186

- Parameter editor dialog box, 175

- Pen Pattern palette, 185

- Spray tool, 185

- tools, selecting, 169-170

 - saving settings, 184-186

 - with AppleEvents extension,
172-177

 - with click QuickKeys, 169-170

 - with real-time QuickKeys, 171-172

- view, changing, 181-183

- Zoom In/Zoom Out windoid, 183

cards (HyperCard)

- moving between, 207-210

- moving objects between layers,
216-218

CE InterApplications Communication
(CEIAC), 10-11

CE Toolbox, 10

CEIAC (CE InterApplications
Communication), 10-11

cells

- borders, adding, 154-156

- formatting, 151-156

- shading, 154-156

ChangeCreator QuickKey, 79-82

ChangeRuler QuickKey, 179

Character dialog box (Microsoft Word),
61

characters, transposing, 55-56

charts, creating from spreadsheets,
162-163

CheckThenPrint QuickKey, 62-65

CheckTotal QuickKey, 164-166

- Claris Resolve, 397
 - Paste Function window, 144
 - Publish and Subscribe, automating, 374-376
 - spreadsheet views, changing, 137
 - SUM formula, entering, 145-146
- Click Editor, 42
 - ClickUpLeft QuicKey, 50
 - DragHome QuicKey, 46
 - HideMe QuicKey, 43
 - InstantBlank QuicKey, 52
- Click Location editor, 42
 - Click QuicKey, 43-45
 - DragRight QuicKey, 47
 - TogglePartLabel QuicKey, 289
 - Window icon, 170
- click QuicKeys, 43-45, 408
- ClickMarquee QuicKey, 205-206
- ClickUpLeft QuicKey, 50
- ClipboardAction extension
 - AutoConnect QuicKey, 238
 - CheckTotal QuicKey, 164-166
 - ToggleUserLevel QuicKey, 222
- Close (File menu) command, 406
- Close mousies, 41
- Close QuicKey, 34
- Close Window mousie, 35
- CollectCISMail QuicKey, 282
- column labels, formatting, 157-160
- CombineFiles QuicKey, 99-106
 - Paste Ease editor, 105
 - pauses, adding, 101-102
 - Repeat extension, 101
- combining files, 99-106
 - lines, adding, 103-104
- commands
 - alignment, 141-142
 - Close (File menu), 406
 - Compress Files (Options menu), 411
 - Configure the QuickReference card... (Options menu), 412
 - File (Define menu), 16-17
 - Open (File menu), 11, 406
 - Save A Copy (File menu), 406-407
 - Save Selection (File menu), 406-407
 - Sort by Key (Options menu), 411
 - Sort by Name (Options menu), 411
 - Sort by Type (Options menu), 411
- communications programs
 - America Online, 265
 - collecting electronic mail, 265-268
 - FlashSession feature, 274
 - Go To Keyword window, 275
 - launching, 269-271
 - MacIntosh Utilities forum window, 276
 - New Mail window, 272
 - opening forums, 274-276
 - saving electronic mail, 268
 - shutting down Macintosh, 282
 - backing up files, 250-257
 - CompuServe Information Manager (CIM)
 - automating electronic mail, 259-263
 - File editor, 280
 - replying to electronic mail, 263-265
 - shutting down Macintosh, 279-282
 - disconnecting Remote Access, 257-259
 - electronic mail
 - cleaning mailbox, 230-234
 - printing, 285
 - reading from home, 243-249
 - log-in procedure, automating, 234-239
 - VT-102 terminal keypad, mimicing, 240-243
- CompletePrint QuicKey, 58-59
- Compress Files (Options menu)
 - command, 411
- CompressMail QuicKey, 230-234
- CompuServe Information Manager (CIM), 394
 - electronic mail
 - automating, 259-263
 - replying to, 263-265
 - File editor, 280
 - Send & Receive window, 260
 - shutting down Macintosh, 279-282

- Configure the QuickReference card...
 - (Options menu) command, 412
- ConfiguredSUM QuickKey, 146
- configuring
 - HyperCard, 224-228
 - printers, 58-62
- CONNECT, 394
- Control Area dialog box, 60
- Control Tower program, 7, 390-391, 397
- Copy To dialog box (DiskTop), 255
- CreateChart QuickKey, 162-163
- CreateSEA QuickKey, 384-387
- creating
 - macros
 - as modules, 20
 - grouping macros, 22-23
 - keystrokes, assigning, 22
 - program-specific, 33-36
 - strategies, 20-23
 - templates, 27
 - universal versus program-specific, 21-22
 - screen savers, 51-52
 - styles, 117-119
- CursorWait extension (New MUT Files QuickKey), 277
- Cut2Bkgnd QuickKey, 216-217
- cutting
 - archiving, 53-54
 - spreadsheet items, 140-141

D

- databases
 - addresses
 - entering, 303-304
 - looking up, 377-383
 - aligning objects, 301-302
 - envelopes, printing, 128-130
 - fields
 - number, setting attributes, 294-296
 - setting up, 291-296
 - text, setting attributes, 293-294
 - keyboard shortcuts, adding, 287-291
 - records
 - converting for word processor, 310-313
 - importing, 313-314
 - paging through, 296-300
 - searches, automating, 304-310
 - views, navigating, 300
- DataDesk International templates, 27
- Date Format dialog box (FileMaker Pro), 294
- date/time QuickKeys, 408
- DecreaseSize QuickKey, 144
- Define menu, 39, 407-411
 - File command, 16-17
- deleting files, 318-320
- Demo QK2.sea archive, 427
- Desktop, rebuilding, 340-342
- dialog boxes
 - Control Area, 60
 - File Open, 17-18
 - Grab Ease, 104
 - interrupt, 50
 - New Folder, 354
 - Options, 405-406, 412
 - Repeat, 71
 - Select Files, 332
 - Set Speeds, 68
 - Statistics, 404
 - Timer Options, 19, 75
- DiskTools Plus, 398
- DiskTools program, backing up files, 250-257
- DiskTop program, 79-82, 398
 - backing up files, 250-257, 347-360
 - Copy To dialog box, 255, 352
 - deleting files, 318-320
 - Find Criteria dialog box, 360
 - Get Info window, 81
- DisMounty extension, 356
- displaying
 - files, 345-347
 - folders, 345-347
 - memory buffer size, 405

documents

markers

- finding, 93-97
- inserting, 91-93

opening, 16-20

pages, jumping to, 97-98

spaces, finding, 87-91

spell checking, 62-65

dragging windows

home, 44-46

to the right, 46-47

DragHome QuickKey, 44-46

DragRight QuickKey, 46-47

Dynodex, 395

addresses, looking up, 377-383

envelopes, printing, 128-130

DynoFind QuickKey, 377-381

DynoPrintOne QuickKey, 128-129

Ee-mail, *see* electronic mail

editors, 34

Click, 42

ClickUpLeft QuickKey, 50

DragHome QuickKey, 46

HideMe QuickKey, 43

InstantBlank QuickKey, 52

Click Location, 42

Click QuickKey, 43-45

DragRight QuickKey, 47

TogglePartLabel QuickKey, 289

Window icon, 170

Menu

Close QuickKey, 34

Restart QuickKey, 317

ToggleRulers QuickKey, 178

Parameter

2xMag QuickKey, 183

PenPattern QuickKey, 189

Paste Ease (CombineFiles QuickKey), 105

Sequence (CompletePrint QuickKey), 59

WindowAction

GetMail1 QuickKey, 267

Microsoft Mail, 245

New MUT Files QuickKey, 277

Quit&Place QuickKey, 198

electronic mail

America Online

collecting, 265-268

saving, 268

cleaning mailbox, 230-234

CompuServe Information Manager (CIM)

automating, 259-263

replying to, 263-265

disconnecting Remote Access, 257-259

printing, 285

reading from home, 243-249

Empty&ShutDown QuickKey, 319-321

emptying trash, 321-325

EmptyTrash QuickKey, 321-325

Enter key, functioning as Return key, 134-135

envelopes, printing, 125-130

Escapade freeware program, 20, 109, 400, 428

events

Apple Event, 363

External Tool, 177

magnification, 182

Excel, *see* Microsoft Excel

Export dialog box (FileMaker Pro), 311

ExtendedFind QuickKey, 83-86

extensions, 3-4, 408

Apple Events, 10

EmptyTrash QuickKey, 323

Event menu, 367

Return Value menu, 370

Send Event to menu, 366

selecting tools, 172-177

setting up tools, 186-189

ClipboardAction

AutoConnect QuickKey, 238

CheckTotal QuickKey, 164-166

ToggleUserLevel QuickKey, 222

- CursorWait (New MUT Files QuickKey), 277
- DisMounty, 356
- Finder Event, 11
 - Open command, 331
 - Print command, 326
 - Select Files dialog box, 332
- Message
 - AreaOfaCircle QuickKey, 148
 - ExtendedFind QuickKey, 84
 - II/NTEnvelope QuickKey, 127
 - SaveCopy QuickKey, 73
 - SetUp&Alert QuickKey, 308
 - TimeLogger QuickKey, 77
- Mounty, 356
- Repeat (CombineFiles QuickKey), 101
- SoftKeys, 23
- WindowAction
 - Empty&ShutDown QuickKey, 320
 - RebuildDesktop QuickKey, 341
 - SaveCopy QuickKey, 73
 - Sharing QuickKey, 335
- Extensions folder, 427
- External Tool event, 177

F

- F-Key Flipbook, 27
- Field Info dialog box (HyperCard), 226
- fields
 - database
 - setting up, 291-296
 - text attributes, setting, 293-294
 - HyperCard
 - configuring, 226-228
 - moving to background, 216-218
- File (Define menu) command, 16-17
- File menu, 406-407
- File Open dialog box, 17-18
- file QuickKeys, 409

- FileMaker Pro, 395
 - aligning objects, 301-302
 - Alignment dialog box, 301
 - Book icon, 297
 - Date Format dialog box, 294
 - Export dialog box, 311
 - fields
 - number, setting attributes, 294-296
 - setting up, 291-296
 - text, setting attributes, 293-294
 - keyboard shortcuts, adding, 287-291
 - Layout pop-up menu, 300
 - Number Format dialog box, 295
 - Part Label icon, 288
 - records, paging through, 296-300
 - searches, automating, 304-310
 - Status Panel icon, 288
 - Symbols pop-up menu, 310
 - Text Format dialog box, 292-293
 - views, navigating, 300
 - Zoom icon, 288
- files
 - backing up, 72-76, 347-357
 - incremental backups, 353-355
 - modified files, 357-360
 - remotely, 250-257
 - to AppleShare server, 355-357
 - batch, 1
 - combining, 99-106
 - lines, adding, 103-104
 - deleting, 318-320
 - displaying, 345-347
 - finding, 83-86
 - printing, 325-329
 - saving, 69-72
 - sharing, 329-333
 - setting privileges automatically, 333-345
 - starting on remote computer, 365-374
 - stopping, 370
 - stuffing, 383-390
 - text, opening, 79-82

Finder

- backing up files, 347-357
 - incremental backups, 353-355
 - modified files, 357-360
 - to AppleShare server, 355-357
- deleting files, 318-320
- Desktop, rebuilding, 340-342
- displaying files, 345-347
- printing files, 325-329
- quitting, 342-345
- sharing files, 329-333
 - setting privileges automatically, 333-339
- shutting down Macintosh, 321
- windows
 - dragging, 44-46
 - hiding, 41-44

Finder Event extension, 11

- Open command, 331
- Open dialog box, 196
- Print command, 326
- Select Files dialog box, 332

finding

- files, 83-86
- markers, 93-97
- spaces, 87-91

FindMarker QuickKey, 93-97

FindWhole QuickKey, 204

Fkey Catalog, 396

FKey QuickKeys, 409

FlashSession (America Online), 274

FlashWrite II shareware program, 78-79, 400

folders

- displaying, 345-347
- stuffing, 383-390

Font dialog box (HyperCard), 227

Font window (Microsoft Excel), 152

fonts, formatting, 114-115

FormatAWord QuickKey, 114-115

FormatLabels QuickKey, 151-154

formatting

- cells, 151-156
- fonts, 114-115
- spreadsheets, 156-164
 - column labels, 157-160
 - row labels, 157-160
- text, 114-115
 - styles, creating, 117-119

FormatWorksheet QuickKey, 157-160

formulas, SUM, entering, 145-146

FreeTerm freeware program, 400

freeware programs

- Escapade, 20, 428
- FreeTerm, 400
- Rebound, 400, 428
- SCSIProbe, 94
- Telnet, 400

Frontier scripting environment, 6, 390, 397

FullWrite Professional, 399

functions, HyperCard, creating

- templates for, 213-216

FunctionTemp QuickKey, 216

G

GenTransposChar QuickKey, 56

Get Info window (DiskTop), 81

Get&SendMail QuickKey, 259-263

GetMail1 QuickKey, 265-268

GetMSMail QuickKey, 243-247

GetQuickMail QuickKey, 248-249

Go Technology, Hot Keys, 5

GOfer 2.0, 398

GoToFirst QuickKey, 296-300

GoToPage QuickKey, 97

Grab Ease dialog box, 104

graphics

- checklist, format, 110
- inserting, 109-113
- markers, adding, 106-109

- rulers
 - settings, changing, 179
 - toggling, 177-180
- text, wrapping, 190-192
- tools, selecting, 167-168
 - saving settings, 184-186
 - with Apple Events extension, 172-177
 - with click QuickKey, 168-170
 - with real-time QuickKey, 170-172
- views, changing, 180-183
- GraphList QuickKey, 106-109
- Gravis Super Mouse, 396

H

- HandOff II, 398
- HangUpOffice QuickKey, 257-259
- hardware, 396
- HC2Scrap QuickKey, 218
- HideMe QuickKey, 43
- hiding windows, 41-44
- Hot Keys macros, 5
- HyperCard, 397
 - Button Info... dialog box, 224
 - buttons
 - configuring, 224-225
 - creating, 204
 - moving to background, 216-218
 - cards
 - moving between, 207-210
 - moving objects between layers, 216-218
 - Field Info dialog box, 226
 - fields
 - configuring, 226-228
 - creating, 204
 - moving to background, 216-218
 - Font dialog box, 227
 - functions, creating templates for, 213-216
 - keyboard shortcuts, adding, 202-204
 - message handlers, creating templates for, 213-216
 - patterns, selecting, 204-207
 - stacks, opening multiple, 210-212
 - tools, selecting, 204-207
 - user levels, switching between, 218-228

I

- icons
 - Part Label (FileMaker Pro), 288
 - Status Panel (FileMaker Pro), 288
 - Zoom In, 290
 - Zoom Out, 290
 - Zoom Percentage, 291
- ID codes, 174
- II/NTEnvelope QuickKey, 125-130
 - Message extension, 127
- importing
 - database records, 313-314
 - sequences, 417-421
- IncreaseSize QuickKey, 143-144
- incremental backups, 353-355
- IncrementBackup QuickKey, 353-355
- InsertGraphic QuickKey, 109-113
- inserting
 - graphics, 109-113
 - markers, 91-93
- insertion point, moving, 37-40
- InsertMarker QuickKey, 91-93
- installing QuickKeys demo version, 8-13
- Instant QuickKeys program, 415-416
- InstantBlank QuickKey, 51-52
- intelligent tools (graphics programs, 184-189
- Internet, 230
- interrupt dialog box, 50
- interrupting macros, 49

J-K

keyboard shortcuts

adding

to database programs, 287-291

to spreadsheet programs, 132-133

assigning, 22

to VT-102 keypad, 239-243

creating

for Finder, 316-317

for HyperCard commands, 202-204

for View menu commands, 318

overlapping, 40

running out of, 23-26

keysets

copying into QuickKeys, 11-13

limitations, 416-417

naming, 417

opening, 406

Resolve, 156

saving, 406

Keysets folder, 427

Keysets menu, 413

Keystroke Assigner (Instant QuickKeys program), 415

L

labels

column, formatting, 157-160

formatting, 151-154

row, formatting, 157-160

LaserWriter II series printers, printing envelopes, 125-130

Launch&Lookup QuickKey, 382-383

Layout pop-up menu (FileMaker Pro), 300

letters

business, creating template, 120-122

printing, 125-130

LetterTemplate QuickKey, 120-122

log-in procedure, automating, 234-239

logging time, 76-79

Logitech mouse, 25

LogOnAOL QuickKey, 269-272

Lotus 1-2-3, 398

keyboard shortcuts, adding, 132-133

SUM formula, entering, 145-146

Window Split window, 132

M

Mac 101E, 396

Mac Pro Plus, 396

MacConnection, 400

MacDraw, rulers

settings, changing, 179

toggling, 177-180

MacDraw Pro, 396

Rulers dialog box, 179

View menu, 178

Macintosh, shutting down, 321

Macintosh Utilities forum, 276

MacPS program, 2

MacroMaker program, 4-5, 397

macros

creating

as modules, 20

grouping macros, 22-23

keystrokes, assigning, 22

program-specific, 33-36

strategies, 20-23

templates, 27

universal versus program-specific, 21-22

future of, 6-7

history, 1-7

interrupting, 49

menu, 32-37

opening text files, 79-82

planning, 15-16

running

time, setting, 19, 20

with mouse, 25-26

text editing, 37-40

- MacWarehouse, 400
- MacWrite II, 399
 - graphics, inserting, 113
- MacZone, 400
- Magnification menu (UltraPaint), 181
- Magnify200% QuicKey, 137
- markers
 - finding, 93-97
 - graphic, adding, 106-109
 - inserting, 91-93
- MasterFinder, 398
- MasterScript, HyperCard add-on, 216, 397
- McSink shareware program, 400
- memory, displaying buffer size, 405
- Memory Indicator, 404-406
- Menu editor
 - Close QuicKey, 34
 - Restart QuicKey, 317
 - ToggleRulers QuicKey, 178
- menu macros, 32-37
- Menu/DA QuicKeys, 409
- MenuAction QuicKey, 279-281
- menus
 - Define, 39, 407-411
 - File, 406-407
 - Keysets, 413
 - Options, 411-412
 - View, creating keyboard shortcuts, 318
- Message extension
 - AreaOfaCircle QuicKey, 148
 - ExtendedFind QuicKey, 84
 - II/NTEnvelope QuicKey, 127
 - SaveCopy QuicKey, 73
 - SetUp&Alert QuicKey, 308
 - TimeLogger QuicKey, 77
- message handlers (HyperCard), creating templates for, 213-216
- Microphone II, 394
 - VT-102 terminal, mimicing keypad, 240-243
- Microsoft Excel, 397
 - Alignment window, 142
 - charts, creating from spreadsheets, 162-163
 - Font window, 152
 - Number window, 159
 - Page Setup dialog box, 160
 - spreadsheets
 - alignment commands, 141-142
 - cutting items, 140-141
 - formatting, 156-164
 - formatting cells, 151-156
 - views, changing, 135, 139-140
 - text size, increasing/decreasing, 143-144
 - Toolbars windows, 154
 - Zoom window, 136
- Microsoft Mail, 394
 - log-in window, 246
 - mailbox window, 246
 - WindowAction editor, 245
- Microsoft Windows
 - characters, transposing, 55-56
 - dragging windows, 46-47
 - tiling windows, 48-49
- Microsoft Word, 399
 - archiving cuts, 53-54
 - bibliography template, creating, 122-125
 - boldface, applying, 116-117
 - Character dialog box, 61
 - documents, spell checking, 62-65
 - envelopes, printing, 125-130
 - graphics
 - inserting, 109-113
 - markers, adding, 106-109
 - markers
 - finding, 93-97
 - inserting, 91-93
 - pages, jumping to, 97-98
 - Paragraph dialog box, 123
 - printer setup, configuring, 58-62

- saving files, 69-72
- Search menu, 94
- spaces, finding, 87-91
- styles, 118-119
- text, formatting, 114-115
- MicroTouch Systems, UnMouse
 - touchpad, 23-25
- modules, 20
- MonitorStartup QuickKey, 329-333
- MORE, 399
- Mounity extension, 356
- mouse, 40-41
 - Advanced Gravis, 26
 - hiding windows, 41-44
 - Logitech, 25
 - playing macros, 25-26
- mouse tracking programs, 2-5
- MouseMan, 396
- mousies, 34, 41, 410
 - Close, 41
 - Close Window, 35
 - hierarchical menu, 36
 - Zoom, 41
- moving
 - between HyperCard cards, 207-210
 - buttons (HyperCard) to background, 216-218
 - insertion point, 37-40
- MultiClip, 398
- Multicomp, F-Key Flipbook, 27
- MultiMaster, 398

N

- naming keysets, 417
- Navigator, 394
- New Folder dialog box, 354
- New MUT Files QuickKey, 278
- NewMail QuickKey, 271-273
- NEWS, assigning VT-102 terminal
 - keystrokes, 239-243
- Next QuickKey, 207-210

- Nisus, 399
- NitePrintJob QuickKey, 325-329
- Number Format dialog box (FileMaker Pro), 295
- Number window (Microsoft Excel), 159

O

- On Cue II, 398
- OnMouseUpTemp QuickKey, 215
- Open (File menu) command, 11, 406
- Open dialog box (Finder Events editor), 196
- OpenHome QuickKey, 210-212
- opening
 - documents, 16-20
 - keysets, 406
 - programs, 65-69
 - stacks (HyperCard), 210-212
 - text files, 79-82
- Options dialog box, 405-406, 412
- Options menu, 411-412
- OutlineBorder QuickKey, 156

P

- Page Setup dialog box (Microsoft Excel), 160
- PageMaker, 395
 - backing up files, 72-76
 - Place command, activating, 192-198
 - Place dialog box, 193
 - Text wrap dialog box, 190
 - wrapping text around graphics, 190-192
- pages, jumping to, 97-98
- palettes, pop-up, 168-169
- Paragraph dialog box (Microsoft Word), 123
- paragraphs, boldface, applying to first line, 116-117

- Parameter editor
 - 2xMag QuickKey, 183
 - PenPattern QuickKey, 189
- Part Label (FileMaker Pro) icon, 288
- Paste Ease editor (CombineFiles QuickKey), 105
- patterns, selecting in HyperCard, 204-207
- PenPattern QuickKey, 189
- PhoneNET Liasion, 394
- Place command (PageMaker), activating, 192-198
- Place dialog box (PageMaker), 193
- pop-up palettes, 168-169
- PowerKey control panel device, 283-285, 396
- Prev QuickKey, 207-210
- printers
 - configuring, 58-62
 - LaserWriter II series, printing envelopes, 125-130
- printing
 - electronic mail, 285
 - envelopes, 125-130
 - files, 325-329
 - letters, 125-130
- PrintMessages QuickKey, 285-286
- Program-to-Program Communication Toolbox, 367
- programs
 - After Dark, 51-52, 398
 - Amazing Paint, 395
 - AppleScript, 7, 389-390
 - Canvas, 396
 - external tools, 176
 - Foreground Color palette, 186
 - Parameter editor dialog box, 175
 - Pen Pattern palette, 185
 - Spray tool, 185
 - tools, selecting, 169-177, 184-186
 - view, changing, 181-183
 - Zoom In/Zoom Out windoid, 183
- Claris Resolve, 397
 - Paste Function window, 144
 - Publish and Subscribe, automating, 374-376
 - spreadsheet views, changing, 137
 - SUM formula, entering, 145-146
- communications, 394-395
- Control Tower, 7, 390-391, 397
- database, 395
- DiskTop, 79-82
 - backing up files, 250-257, 347-360
 - Copy To dialog box, 255, 352
 - deleting files, 318-320
 - Find Criteria dialog box, 360
 - Get Info window, 81
- electronic mail, 394-395
- FileMaker Pro, 395
 - aligning objects, 301-302
 - Alignment dialog box, 301
 - Book icon, 297
 - Date Format dialog box, 294
 - Export dialog box, 311
 - fields, 291-296
 - keyboard shortcuts, adding, 287-291
 - Layout pop-up menu, 300
 - Number Format dialog box, 295
 - Part Label icon, 288
 - records, paging through, 296-300
 - searches, automating, 304-310
 - Status Panel icon, 288
 - Symbols pop-up menu, 310
 - Text Format dialog box, 292-293
 - views, navigating, 300
 - Zoom icon, 288
- freeware
 - Escapade, 428
 - FreeTerm, 400
 - Rebound, 400, 428
 - SCSIProbe, 94
 - Telnet, 400
- Frontier, 6, 390, 397
- graphics, 395-396
- HandOff II, 398
- hypertextual, 397

- Instant QuicKeys, 415-416
- Lotus 1-2-3, 398
 - keyboard shortcuts, adding, 132-133
 - SUM formula, entering, 145-146
 - Window Split window, 132
- MacDraw Pro, 396
 - Rulers dialog box, 179
 - View menu, 178
- MacPS, 2
- MacroMaker, 4-5
- MacWrite II, 399
 - graphics, inserting, 113
- Microsoft Excel, 397
 - alignment commands, 141-142
 - Alignment window, 142
 - charts, creating from spreadsheets, 162-163
 - cutting items, 140-141
 - Font window, 152
 - formatting cells, 151-156
 - formatting spreadsheets, 156-164
 - Number window, 159
 - Page Setup dialog box, 160
 - views, changing, 135, 139-140
 - text size, increasing/decreasing, 143-144
 - Toolbars windows, 154
 - Zoom window, 136
- Microsoft Mail, 394
 - log-in window, 246
 - mailbox window, 246
 - WindowAction editor, 245
- Microsoft Windows
 - characters, transposing, 55-56
 - dragging windows, 46-47
 - tiling windows, 48-49
- Microsoft Word, 399
 - adding markers, 106-109
 - archiving cuts, 53-54
 - bibliography template, creating, 122-125
 - boldface, applying, 116-117
 - Character dialog box, 61
 - documents, spell checking, 62-65
 - envelopes, printing, 125-130
 - finding markers, 93-97
 - inserting graphics, 109-113
 - inserting markers, 91-93
 - pages, jumping to, 97-98
 - Paragraph dialog box, 123
 - printer setup, configuring, 58-62
 - saving files, 69-72
 - Search menu, 94
 - spaces, finding, 87-91
 - styles, 118-119
 - text, formatting, 114-115
- mouse tracking, 2-5
- On Cue II, 398
- opening, 65-69
- page layout, 395-396
- PageMaker, 395
 - backing up files, 72-76
 - Place command, activating, 192-198
 - Place dialog box, 193
 - Text wrap dialog box, 190
 - wrapping text around graphics, 190-192
- QK Icons, 413-415
- QuicKeys, *see* QuicKeys
- ResEdit, 36, 399
- scripting, 397
- shareware
 - Boomerang, 400
 - Escapade, 109, 400
 - FlashWrite II, 78-79, 400
 - McSink, 400
 - StuffIt Lite, 383-390, 400
- spreadsheet, 397
- Tempo, 2-3
- text editors, 399-401
- UltraPaint
 - Magnification pop-up menu, 181
 - view, changing, 180-183
 - word processing, 399-401
- WordPerfect, 399
- WriteNow, 399
 - styles, creating, 117-119
 - transposing words, 57-58
- Publish and Subscribe, automating, 374-376

Q

QK Icons program, 413-415

QuickKeys, 3-4

2SpacesTo1, 87-91

2xMag, 183

AEUnStuff, 388-389

alias, 38-40, 407

AlignLeftEdges, 302

AreaOfaCircle, 146-150

AutoAddress, 303-304

AutoConnect, 238-239

AutoExport, 310-313

AutoImport, 313-314

AutoSUM, 145-146

BackUpMods, 357-358

BackUpProject, 347-353

BibTemplate, 122-125

Bold1stLine, 116-117

Border&Shade, 154-155

button, 408

CE Toolbox, 10

CEIAC, 10-11

ChangeCreator, 79-82

ChangeRuler, 179

CheckThenPrint, 62-65

CheckTotal QuickKey, 164-166

click, 43-45, 408

ClickMarquee, 205-206

ClickUpLeft, 50

Close, 34

CollectCISMail, 282

CombineFiles, 99-106

 Paste Ease editor, 105

 pauses, adding, 101-102

 Repeat extension, 101

CompletePrint, 58-59

CompressMail, 230-234

ConfiguredSUM, 146

CreateChart, 162-163

CreateSEA, 384-387

Cut2Bkgnd, 216-217

date/time, 408

DecreaseSize, 144

DragHome, 44-46

DragRight, 46

DynoFind, 377-381

DynoPrintOne, 128-129

Empty&ShutDown, 319-321

EmptyTrash, 321-325

ExtendedFind, 83-86

extensions, 3-4

file, 409

FindMarker, 93-97

FindWhole, 204

FKey, 409

FormatAWord, 114-115

FormatLabels, 151-154

FormatWorksheet, 157-160

FunctionTemp, 216

GenTransposChar, 56

Get&SendMail, 259-263

GetMail1, 265-268

GetMSMail, 243-247

GetQuickMail, 248-249

GoToFirst, 296-300

GoToPage, 97

GraphList, 106-109

HangUpOffice, 257-259

HC2Scrap, 218

HideMe, 43

II/NTEnvelope, 125-130

IncreaseSize, 143-144

IncrementBackup, 353-355

InsertGraphic, 109-113

InsertMarker, 91-93

installing demo version, 8-13

InstantBlank, 51-52

keysets

 copying, 11-13

 limitations, 416-417

 naming, 417

Launch&Lookup, 382-383

LetterTemplate, 120-122

limitations of, 416-417

listing, 412-413

- LogOnAOL, 269-272
 - Magnify200%, 137
 - Menu/DA, 409
 - MenuAction, 279-281
 - MonitorStartup, 329-333
 - New MUT Files, 277-278
 - NewMail, 271-273
 - Next, 207-210
 - NitePrintJob, 325-329
 - OnMouseUpTemp, 215
 - OpenHome, 210-212
 - OutlineBorder, 156
 - PenPattern, 189
 - Prev, 207-210
 - PrintMessages, 285-286
 - Quick Reference Card, 23
 - QuickUnStuff, 388-389
 - Quit&Place, 192-198
 - QuitFinder, 342-345
 - real-time, 410
 - RebuildDesktop, 340-342
 - RemoteBackUp, 250-257
 - RepeatingPlace, 198-199
 - ResolveDoScript, 374-376
 - Restart, 316-317
 - Return alias, 134-135
 - RightJustify, 141-142
 - SaveACopy&Quit, 76
 - SaveAll, 69-72
 - SaveCopy, 72-76
 - SelectMarquee, 207
 - Sequence Editor, 4
 - SetPrivileges, 336-338
 - SetUp&Alert, 308
 - SetUpAOLReply, 265
 - SetUpButton, 224-225
 - SetUpField, 226-227, 291-292
 - SetUpNumber, 294
 - SetUpReplyWin, 263-265
 - SetUpSearch, 304-309
 - SetUpSpray, 189
 - SetUserLevel, 218-223
 - Sharing, 333-336
 - Shift-Enter alias, 135
 - ShowQKFolder, 345-347
 - SignOff, 268-269, 282-283
 - SplitHorizontal, 132-133
 - StartMeUP, 65-69
 - StartSharing, 338-339, 366-370
 - system requirements, vi-vii
 - text, 37-41, 410
 - TimeLogger, 76-79
 - TiteWrap, 190-191
 - TogglePartLabel, 288-289
 - ToggleRulers, 177-180
 - ToggleUserLevel, 220-222
 - TransposeWord, 57
 - UseThousands, 296
 - VAXLogin, 234-239
 - VolOfaCylinder, 150
 - WrapText, 191-192
 - QuicKeys editor, 403
 - Memory Indicator, 404-406
 - menus, 406-412
 - QuickMail, 395
 - reading messages, 248-249
 - QuickStuff Apple Event, 384-388
 - QuickUnStuff Apple Event, 388-389
 - QuickUnStuff QuickKey, 388-389
 - Quit&Place QuickKey, 192-198
 - QuitFinder QuickKey, 342-345
 - quitting Finder, 342-345
- R**
- real-time QuicKeys, 410
 - compressing, 411
 - Rebound freeware program, 400, 428
 - RebuildDesktop QuickKey, 340-342
 - rebuilding Desktop, 340-342
 - records (database)
 - converting for word processor, 310-313
 - importing, 313-314
 - paging through, 296-300
 - searches, automating, 304-310

Remote Access, *see* Apple Talk Remote Access
RemoteBackUp QuicKey, 250-257
Repeat dialog box, 71
Repeat extension (CombineFiles QuicKey), 101
RepeatingPlace QuicKey, 198-199
Required suite (Apple Events), 363
ResEdit program, 36, 399
Resolve, *see* Claris Resolve
ResolveDoScript QuicKey, 374-376
Restart QuicKey, 316-317
Return alias QuicKey, 134-135
RightJustify QuicKey, 141-142
rows, formatting labels, 157-160
rulers
 settings, changing, 179
 togglng, 177-180
Rulers dialog box (MacDraw Pro), 179
running macros
 time, setting, 19-20
 with mouse, 25-26

S

Save A Copy (File menu) command, 406-407
Save Selection (File menu) command, 406-407
SaveACopy&Quit QuicKeys, 76
SaveAll QuicKey, 69-72
SaveCopy QuicKey, 72-76
 Message extension, 73
 WindowAction extension, 73
saving
 files, 69-72
 keysets, 406
Scrapbook, archiving cuts, 53-54
screen savers, creating, 51-52
SCSIProbe freeware program, 94
Search menu (Microsoft Word), 94
Select Files dialog box, 332
Select rear window special, 41

Select second window special, 41
selecting
 graphics tools, 167-168
 in HyperCard, 204-207
 with Apple Events extension, 172-177
 with click QuicKey, 168-170
 with real-time QuicKey, 170-172
 HyperCard patterns, selecting, 204-207
 text, 40
SelectMarquee QuicKey, 207
Sequence editor, 4
 CompletePrint QuicKey, 59
sequences, 8, 410
 compressing, 411
 importing, 417-421
Set Speeds dialog box, 68
SetPrivileges QuicKey, 336-338
SetUp&Alert QuicKey, 308
SetUpAOLReply QuicKey, 265
SetUpButton QuicKey, 224-225
SetUpField QuicKey, 226-227, 291-292
SetUpNumber QuicKey, 294
SetUpReplyWin QuicKey, 263-265
SetUpSearch QuicKey, 304-309
SetUpSpray QuicKey, 189
SetUserLevel QuicKey, 218-223
shading, adding to spreadsheet cells, 154-156
shareware programs
 Boomerang, 400
 Escapade, 109, 400
 FlashWrite II, 78-79, 400
 McSink, 400
 StuffIt Lite, 383-390, 400
sharing files, 329-333
 setting privileges automatically, 333-339
 starting on remote computer, 365-374
 stopping, 370
Sharing QuicKey, 333-336
Sharing Setup control panel, 330, 365
Shift-Enter alias QuicKey, 135
Shortcut, 399

- ShowQKFolder QuickKey, 345-347
 - shutting down Macintosh, 321
 - SignOff QuickKey, 268-269, 282-283
 - Simple Software, Control Tower, 7
 - SoftKeys extension, 23
 - Sophisticated Circuits, PowerKey
 - control panel device, 283-285
 - Sort by Key (Options menu)
 - command, 411
 - Sort by Name (Options menu)
 - command, 411
 - Sort by Type (Options menu)
 - command, 411
 - specials, 34, 41, 410
 - hierarchical menu, 48
 - Select rear window, 41
 - Select second window, 41
 - Toggle QuickKeys On/Off, 40
 - spell checking documents, 62-65
 - SplitHorizontal QuickKey, 132-133
 - Spray tool (Canvas), 185
 - spreadsheet programs, 397
 - spreadsheets
 - alignment commands, 141-142
 - cells
 - borders, adding, 154-156
 - formatting, 151-156
 - shading, adding, 154-156
 - charts, creating, 162-163
 - cutting items, 140-141
 - Enter key, functioning as Return key, 134-135
 - equations, solving, 146-150
 - formatting, 156-164
 - column labels, 157-160
 - row labels, 157-160
 - keyboard shortcuts, adding, 132-133
 - labels, formatting, 151, 154
 - SUM formula, entering, 145-146
 - text type size, increasing/decreasing, 143-144
 - views, changing, 135, 139-140
 - stacks (HyperCard)
 - cards, moving between, 207-210
 - opening multiple, 210-212
 - StartMeUp QuickKey, 65-69
 - StartSharing QuickKey, 366-370
 - StartSharing QuickKey, 338-339
 - startup, opening programs, 65-69
 - Statistics dialog box, 404
 - Status Panel (FileMaker Pro) icon, 288
 - StuffIt Lite shareware program, 383-390, 400
 - styles, creating, 117-119
 - SUM formula, entering, 145-146
 - SuperCard, *see* HyperCard
 - Symbols pop-up menu (FileMaker Pro), 310
 - System 7
 - AppleEvents extension
 - EmptyTrash QuickKey, 323
 - Event menu, 367
 - Return Value menu, 370
 - Send Event to menu, 366
 - tools, 172-177, 186-189
 - file sharing, 365-374
 - Program-to-Program Communication
 - Toolbox, 367
 - Publish and Subscribe, automating, 374-376
 - Sharing Setup control panel, 330, 365
 - trash, emptying, 321-325
- ## T
- telecommunications programs
 - America Online, 265
 - collecting electronic mail, 265-268
 - FlashSession feature, 274
 - Go To Keyword window, 275
 - launching, 269-271
 - MacIntosh Utilities forum
 - window, 276
 - New Mail window, 272
 - opening forums, 274-276
 - saving electronic mail, 268
 - shutting down Macintosh, 282
 - backing up files, 250-257

- CompuServe Information Manager (CIM)
 - automating electronic mail, 259-263
 - File editor, 280
 - replying to electronic mail, 263-265
 - shutting down Macintosh, 279-282
- disconnecting Remote Access, 257-259
- electronic mail
 - cleaning mailbox, 230-234
 - printing, 285
 - reading from home, 243-249
- log-in procedure, automating, 234-239
- VT-102 terminal keypad, mimicing, 240-243
- Teleport FullFax, 395
- Telnet freeware program, 400
- templates
 - bibliography, creating, 122-125
 - business letter, creating, 120-122
 - DataDesk International, 27
- Tempo II Plus, 397
- Tempo program, 2-3
- terminals, VT-102, mimicing keypad, 240-243
- text
 - formatting, 114-115
 - styles, creating, 117-119
 - selecting, 40
 - spreadsheet, type size, increasing/decreasing, 143-144
 - wrapping around graphics, 190-192
- text editing macros, 37-40
- text editor programs, 399-401
- text files, opening, 79-82
- Text Format dialog box (FileMaker Pro), 293
- text QuicKeys, 37-41, 410
- tiling windows, 48-50
- time, logging, 76-79
- TimeLogger QuicKey, 76-79
- Timer Options dialog box, 19, 75
- TiteWrap QuicKey, 190-191
- Toggle QuicKeys On/Off special, 40

- TogglePartLabel QuicKey, 288-289
- ToggleRulers QuicKey, 177-180
- ToggleUserLevel QuicKey, 220-222
- toggling
 - HyperCard user levels, 218-223
 - rulers, 177-180
- Toolbars window (Microsoft Excel), 154
- tools
 - intelligent, 184-189
 - selecting, 167-168
 - in HyperCard, 204-207
 - with Apple Events extension, 172-177
 - with click QuicKey, 168-170
 - with real-time QuicKey, 170-172
 - settings, saving, 184-186
- TransposeWord QuicKey, 57
- transposing
 - characters, 55-56
 - words, 57-58
- trash, emptying, 321-325

U

- UltraPaint
 - Magnification pop-up menu, 181
 - view, changing, 180-183
- UnMouse touchpad, 23-25, 396
- user levels (HyperCard), switching between, 218-223
- UseThousands QuicKey, 296

V

- Vantage, 399
- VAXLogin QuicKey, 234-239
- Versa Term-PRO, 395
 - electronic mailbox, cleaning, 230-234
 - log-in procedure, automating, 234-239
- View menu, keyboard shortcuts, creating, 318
- View menu (MacDraw Pro), 178

- views, changing
 - database programs, 300
 - graphics programs, 180-183
 - spreadsheet programs, 135, 139-140
- VolOfaCylinder QuicKey, 150
- VT-102 terminal, mimicing keypad, 240-243

W

- White Knight, 395
- windoids, 168-169
- Window Split window (Lotus 1-2-3), 132
- WindowAction extension
 - Empty&ShutDown QuicKey, 320
 - GetMail1 QuicKey, 267
 - Microsoft Mail, 245
 - New MUT Files QuicKey, 277
 - Quit&Place QuicKey, 198
 - RebuildDesktop QuicKey, 341
 - SaveCopy QuicKey, 73
 - Sharing QuicKey, 335
- windows
 - dragging
 - home, 44-46
 - to the right, 46-47
 - hiding, 41-44
 - Remote Access, 244
 - tiling, 48-50
 - VersaTerm-PRO, 232
- word processors, 399-401
 - combining files, 99-106
 - envelopes, printing, 125-130
 - graphics
 - inserting, 109-113
 - markers, adding, 106-109
 - markers, inserting, 91-97
 - pages, jumping between, 97-98
 - styles, creating, 117-119

- templates, creating
 - bibliography, 122-125
 - business letter, 120-122
- text
 - boldface, applying, 116-117
 - formatting, 114-115
 - spaces, finding, 87-91
- Word, *see* Microsoft Word
- WordPerfect, 399
- words, transposing, 57-58
- wrapping text, 190-192
- WrapText QuicKey, 191-192
- WriteNow, 399
 - styles, creating, 117-119
 - transposing words, 57-58

X-Z

- Zoom (FileMaker Pro) icon, 288
- Zoom In icons, 290
- Zoom mousies, 41
- Zoom Out icons, 290
- Zoom Percentage icon, 291
- Zoom window (Microsoft Excel), 136

SPECIAL OFFER!

**Enjoy the power of QuicKeys™ for just \$95
or order your next upgrade for 50% off!**

We'd like to make sure you get the most out of this book and enjoy the power of QuicKeys. Send in this coupon and receive the current version of QuicKeys for just \$95.00, a savings of \$54.00 off the suggested retail price. If you already own QuicKeys, use this coupon with your next QuicKeys upgrade notice to receive a 50% discount off the suggested upgrade price!

(DO NOT ENCLOSE CASH. Use a check, money order or credit card)

Please fill out the coupon completely. If you are requesting an upgrade, you must include your QuicKeys serial number and the upgrade notice you received from CE Software. Mail coupon to: **CE Software, Inc. • ATTN: Order Department • PO Box 65580
West Des Moines, Iowa • 50265 USA**

This offer expires 12/31/93 and is not good in combination with any other offers.



Please send me the current version of QuicKeys for just \$95.00.



Please send me the QuicKeys upgrade for 50% off suggested price.

Name: _____

Company: _____

Address: _____

City: _____ State: _____ Zip: _____

Country: _____ Serial Number: _____

(Required if upgrading)

Daytime Phone: (____) _____

☐ Enclosed is check or money order drawn on a U.S. bank.

Card Number: _____

Expiration Date: _____

Name on Card: _____

(Please print!)

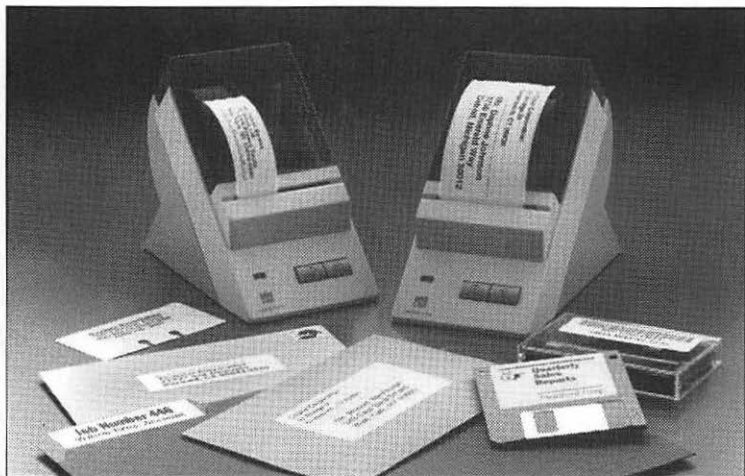
Cardholder Signature: _____

QuicKeys (\$95.00) or QuicKeys Half-Price Upgrade:	
Shipping & Handling (U.S.-Free, Canada \$5.00, Int'l \$15.00)	
CA, IA, GA, MA & VA only: please add appropriate sales tax.	
GRAND TOTAL	

Charge my credit card

- ☐ American Express
☐ MasterCard
☐ Visa

CoStar LabelWriter II Printers – At Special Discount Prices!



The easiest way to label everything around your office and home.

- ☐ Print address, disk, shipping, and video cassette labels – right from your screen.
- ☐ Mix text and graphics on labels. Select among hundreds of fonts.
- ☐ Print bar code 39, Interleaved 2 of 5, UPC, EAN and more.
- ☐ Print a label in just 3.5 seconds, a 350-name list in just 21 minutes.
- ☐ Smaller than a tape dispenser and whisper quiet, they fit anywhere.
- ☐ Labels available from CoStar® and Avery®.
- ☐ Available in DOS/Windows™ or Macintosh® versions.

	Regular Price	SPECIAL PRICE
LabelWriter® II	\$249	\$199
LabelWriter® II Plus	\$369	\$299

To take advantage of this special offer, just call us at 1-800-4-COSTAR or 1-203-661-9700 and ask for our customer service department. They will ask you to fax in a copy of this special offer at 1-203-661-1540.

We accept MasterCard, VISA and American Express. A small shipping and handling charge will be added along with the appropriate sales tax. Offer expires 12/31/93.



CoStar Corporation, 100 Field Point Road, Greenwich, CT 06830-6406

Important! Read Before Opening Sealed Diskette

END USER LICENSE AGREEMENT

The software in this package is provided to You on the condition that You agree with PRENTICE HALL COMPUTER PUBLISHING ("PHCP") to the terms and conditions set forth below. **Read this End User Agreement carefully. You will be bound by the terms of this agreement if you open the sealed diskette.** If You do not agree to the terms contained in this End User License Agreement, return the entire product, along with your receipt, to *Brady, Prentice Hall Computer Publishing, 15 Columbus Circle, 14th floor, New York, NY 10023, Attn: Refunds*, and your purchase price will be refunded.

PHCP grants, and You hereby accept, a personal, nonexclusive license to use this software program and associated documentation in this package, or any part of it ("Licensed Product"), subject to the following terms and conditions:

1. License

The license granted to You hereunder authorizes You to use the Licensed Product on any single computer system. A separate license, pursuant to a separate End User License Agreement, is required for any other computer system on which You intend to use the Licensed Product.

2. Term

This End User License Agreement is effective from the date of purchase by You of the Licensed Product and shall remain in force until terminated. You may terminate this End User License at any time by destroying the Licensed Product together with all copies in any form made by You or received by You. Your right to use or copy the Licensed Product will terminate if You fail to comply with any of the terms or conditions of this End User License Agreement. Upon such termination, You shall destroy the copies of the Licensed Product in your possession.

3. Restriction Against Transfer

This End User License Agreement, and the Licensed Product, may not be assigned, sublicensed, or otherwise transferred by You to another party unless the other party agrees to accept the terms and conditions of the End User License Agreement. If You transfer the Licensed Product, You must at the same time either transfer all copies, whether in printed or machine-readable form, to the same party or destroy any copies not transferred.

4. Restrictions Against Copying or Modifying the Licensed Product

The Licensed Product is copyrighted and except for certain limited uses as noted on the copyright page, may not be further copied without the prior written approval of PHCP. You may make one copy for backup purposes provided You reproduce and include the complete copyright notice on the backup copy. Any unauthorized copying is in violation of this Agreement and may also constitute a violation of the United States Copyright Law for which You could be liable in a civil or criminal lawsuit. **You may not use, transfer, copy, or otherwise reproduce the Licensed Product, or any part of it, except as expressly permitted in this End User License Agreement.**

5. Protection and Security

You shall take all reasonable steps to safeguard the Licensed Product and to ensure that no unauthorized person shall have access to it and that no unauthorized copy of any part of it in any form shall be made.

6. Limited Warranty

If You are the original consumer purchaser of a diskette and it is found to be defective in materials or workmanship (which shall not include problems relating to the nature or operation of the Licensed Product) under normal use, PHCP will replace it free of charge (or, at PHCP's option, refund your purchase price) within 30 days following the date of purchase. Following the 30-day period, and up to one year after purchase, PHCP will replace any such defective diskette upon payment of a \$5 charge (or, at PHCP's option, refund your purchase price), provided that the Limited Warranty Registration Card has been filed within 30 days following the date of purchase. Any request for replacement of a defective diskette must be accompanied by the original defective diskette and proof of date of purchase and purchase price. PHCP shall have no obligation to replace a diskette (or refund your purchase price based on claims of defects in the nature or operation of the Licensed Product).

The software program is provided "as is" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with You. Should the program prove defective, You (and not PHCP) assume the entire cost of necessary servicing, repair or correction.

Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to You. This warrant gives you specific legal rights, and You may also have other rights which vary from state to state.

PHCP does not warrant that the functions contained in the program will meet your requirements or that the operation of the program will be uninterrupted or error free. Neither PHCP nor anyone else who has been involved in the creation or production of this product shall be liable for any direct, indirect, incidental, special, or consequential damages, whether arising out of the use or inability to use the product, or any breach of a warranty, and PHCP shall have no responsibility except to replace the diskette pursuant to this limited warranty (or, at its option, provide a refund of the purchase price).

No sales personnel or other representative of any party involved in the distribution of the Licensed Product is authorized by PHCP to make any warranties with respect to the diskette or the Licensed Product beyond those contained in this Agreement. **Oral statements do not constitute warranties**, shall not be relied upon by You, and are not part of this Agreement. The entire agreement between PHCP and You is embodied in this Agreement.

7. General

If any provision of this End User License Agreement is determined to be invalid under any applicable statute of rule of law, it shall be deemed omitted and the remaining provisions shall continue in full force and effect. This End User License Agreement is to be governed by the construed in accordance with the laws of the State of New York.

Prentice Hall Computer Publishing and the authors make no warranties, express or implied in connection with the software, and expressly exclude all warranties of fitness for a particular purpose. Prentice Hall Computer Publishing and the authors shall have no liability for consequential, incidental, or exemplary damages.



**Keeping Pace With
Today's Microcom-
puter Technology**

Available at your
local bookstore, or
order by phone
(800) 428-5331.

Keep Pace With Today's Microcomputer Technology With

BRADY BOOKS

Act now and get on Brady's mailing list to receive all the latest information on our current and upcoming releases. Brady's best-selling computer books are always up-to-the-minute and geared to meet your needs. Topics include:

- Using major applications
- Beginning, intermediate, and advanced programming
- MS-DOS and Macintosh systems
- Business applications software

Brady continues to maintain its high level of excellence with some of the best, most accomplished writers in the industry:

- Peter Norton on operating systems and programming languages
- Jim Seymour on PC productivity
- Winn L. Rosch on hardware
- Plus, the complete line of official guides to Lotus Software and much, much more!

Get insights into tomorrow's technology with Brady Books.

**MAIL THIS CARD IN TODAY FOR YOUR FREE ENROLLMENT ON
BRADY'S MAILING LIST!**

Name _____

Address _____

City _____ State _____ Zip _____

Name of Book Purchased _____

Date of Purchase _____

Where was this book purchased? (*circle one*)

Retail Store

Computer Store

Mail Order

Place First Class
Postage Here
Post Office
Will Not Deliver
Without Postage

College Marketing Group
50 Cross Street
Winchester, MA 01890

ATT: Cheryl Read

• DISK REPLACEMENT ORDER FORM

In the event that the disk bound in to this book is defective, Prentice Hall Computer Publishing will send you a replacement disk free of charge.

Please fill out the information below and return this card to the address listed below with your original disk. Please print clearly.

BOOK TITLE _____ ISBN _____

NAME _____ PHONE _____

COMPANY _____ TITLE _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

Prentice Hall Computer Publishing, 11711 North College Avenue, Carmel IN 46032.
ATTN: Customer Service Department.

LIMITED WARRANTY REGISTRATION CARD

In order to preserve your rights as provided in the limited warranty, this card must be on file with PHCP within thirty days of purchase.

Please fill in the information requested:

BOOK TITLE _____ ISBN _____

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) _____

PUT
FIRST
CLASS
STAMP
HERE

Prentice Hall Computer Publishing
11711 N. College Avenue
Carmel, IN 46032

Attn: Order Department

BREAK AT SEAL

SOFTWARE FOR
The Automatic Mac
ISBN 0-13-052820-X

OPEN

DS/DD utility disk. Please make backup copy immediately. See Chapter 1 and Appendix C for installation and usage details.

///Brady

PATENT 4,549,888

121-70

PATENT 4,886,488

Make Your Mac Do Your Work For You!

Now you can automate repetitive tasks, modify software programs or improve the way programs work together with *The Automatic Mac*. Become a power user and learn clear, simple techniques for building, testing, and using macros for creative software solutions and greater productivity.

Based on QuickKeys, the best-selling macro-generating program from CE Software, this book shows you how to work efficiently with virtually any program—whether you're working with graphics, word processing, system maintenance, spreadsheets, HyperCard, telecommunications, or database applications. Plus, you learn to use QuickKeys to send and receive Apple Events and to extend your menu, windows, clipboard, and message capabilities.

Become a "power macroer" and you, too, can have an *automatic Mac*!

By pointing and clicking, you can:

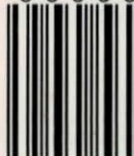
- Close, zoom, grow, or tile the windows in your program
- Press a single keystroke to get started
- Run the spell checker and automatically print the file
- Preset a page jump to quickly move information from page to page
- Insert a graphic place holder in your document
- Format a bibliography or a whole spreadsheet
- Switch quickly between tools
- Navigate quickly through HyperCard stacks
- Log on an electronic service, get your mail, and log off
- Enter the standard information in your database
- Search or sort the contents of a database; then select, set up, and print a report
- Mount AppleShare volumes
- Delete old files
- Open and print long documents and shut down your computer
- Generate a chart and paste it into your word processor document

Book includes a special discount offer from CE Software that gives you an economic boost, whether you are buying QuickKeys for the first time or upgrading to the next major release.

ISBN 0-13-052820-X



9 0000



9 780130 528209

\$39.95 U.S./\$49.95 CAN

Brady Books
Prentice Hall Computer Publishing