



TAKE CONTROL OF

# Automating Your Mac

A friendly guide to  
finding (or creating)  
time-saving shortcuts.

*by* **JOE KISSELL**

**\$14.99**

**6TH  
EDITION**

# Table of Contents

|                                                             |           |
|-------------------------------------------------------------|-----------|
| <b>Read Me First.....</b>                                   | <b>6</b>  |
| Settings vs. Preferences .....                              | 7         |
| What's New in the Sixth Edition .....                       | 7         |
| <b>Introduction .....</b>                                   | <b>9</b>  |
| <b>Automation Quick Start .....</b>                         | <b>11</b> |
| <b>The State of Mac Automation .....</b>                    | <b>13</b> |
| <b>Develop an Automator's Mindset .....</b>                 | <b>18</b> |
| Learn the Basic Principle of Automation .....               | 19        |
| Learn What You Can Automate .....                           | 20        |
| Look for Automation Opportunities .....                     | 22        |
| Pick the Right Tools .....                                  | 28        |
| Let's Have a Chat About AI .....                            | 29        |
| <b>Use Built-In Automation Features .....</b>               | <b>32</b> |
| Use Your Mac's Built-In Keyboard Shortcuts .....            | 33        |
| Make Your Own Keyboard Shortcut .....                       | 36        |
| Use and Customize Toolbars.....                             | 39        |
| Use macOS Text Substitutions and Transformations.....       | 41        |
| Control Your Mac with Your Voice .....                      | 44        |
| Update Apple and Mac App Store Software Automatically ..... | 52        |
| Update Non-Mac App Store Software Automatically .....       | 57        |
| Work with Rule-Based Searches .....                         | 58        |
| Create and Use Smart Containers .....                       | 62        |
| Deal with Evolving Mac Security Features .....              | 64        |
| <b>Automate Your Input Devices .....</b>                    | <b>73</b> |
| Use Trackpad and Magic Mouse Gestures .....                 | 74        |
| Use BetterTouchTool .....                                   | 76        |
| Customize Your Touch Bar .....                              | 77        |
| Save Clicks with Third-Party Input Devices.....             | 79        |

|                                                              |            |
|--------------------------------------------------------------|------------|
| Program an Input Device with USB Overdrive .....             | 81         |
| Control Anything with a Stream Deck .....                    | 81         |
| Learn About Other Special Input Devices .....                | 82         |
| <b>Automate Text Expansion .....</b>                         | <b>84</b>  |
| Use Text Replacement in macOS .....                          | 85         |
| Use a Third-Party Text Expansion Utility .....               | 87         |
| <b>Automate the Finder .....</b>                             | <b>92</b>  |
| Use Spotlight as a Launcher .....                            | 92         |
| Use a Third-Party Launcher .....                             | 95         |
| Organize Files with Hazel .....                              | 100        |
| <b>Supercharge Your Clipboard .....</b>                      | <b>104</b> |
| Use Spotlight's Clipboard History .....                      | 104        |
| Learn What a Clipboard Utility Can Do .....                  | 107        |
| Use a Macro or Launcher Utility .....                        | 109        |
| Use a macOS Clipboard Utility .....                          | 111        |
| <b>Automate Individual Apps .....</b>                        | <b>112</b> |
| Automate Microsoft Office .....                              | 112        |
| Automate Nisus Writer Pro .....                              | 119        |
| Discover Other Internally Scriptable Apps .....              | 130        |
| <b>Automate Email .....</b>                                  | <b>133</b> |
| Use Server-Based Rules .....                                 | 134        |
| Automate Apple Mail .....                                    | 135        |
| Automate MailMaven .....                                     | 144        |
| Automate Outlook Email with Rules .....                      | 144        |
| Automate Other Email Apps .....                              | 145        |
| <b>Automate the Web .....</b>                                | <b>146</b> |
| Log In Faster with iCloud Keychain and Safari Autofill ..... | 146        |
| Automate Web Logins with a Password Manager .....            | 154        |
| Automate Cloud Services .....                                | 156        |
| Discover Other Web Automation Options .....                  | 160        |
| <b>Automate Backup and Syncing .....</b>                     | <b>163</b> |
| Run Backups Automatically with Time Machine .....            | 164        |

|                                                     |            |
|-----------------------------------------------------|------------|
| Create Hands-Off Versioned Backups .....            | 170        |
| Automate Mac-to-Mac Syncing .....                   | 171        |
| <b>Discover macOS Automation Technologies .....</b> | <b>175</b> |
| Apple's Core Automation Technologies .....          | 175        |
| Using JavaScript for Automation .....               | 177        |
| Using Swift for Automation .....                    | 179        |
| <b>Use Services for Systemwide Shortcuts .....</b>  | <b>180</b> |
| Configure Services .....                            | 181        |
| Find and Use Services .....                         | 182        |
| <b>Get Started with Shortcuts .....</b>             | <b>184</b> |
| Explore Shortcuts .....                             | 185        |
| Run a Shortcut .....                                | 186        |
| Import Automator Workflows .....                    | 188        |
| Customize a Shortcut .....                          | 189        |
| Create a Shortcut .....                             | 191        |
| Create an Automation .....                          | 198        |
| Share a Shortcut .....                              | 200        |
| <b>Get Started with Automator .....</b>             | <b>202</b> |
| Create a Simple Automator Workflow .....            | 203        |
| Create an Automator Droplet .....                   | 206        |
| Create Your Own Service .....                       | 209        |
| Find and Run Sample Workflows .....                 | 214        |
| Learn More about Automator .....                    | 214        |
| <b>Get Started with AppleScript .....</b>           | <b>215</b> |
| Write a Simple AppleScript .....                    | 215        |
| Learn What AppleScript Can Do .....                 | 217        |
| Understand AppleScript Basics .....                 | 219        |
| Find and Run Example AppleScripts .....             | 224        |
| Edit an Existing AppleScript .....                  | 228        |
| Use GUI Scripting .....                             | 230        |
| Use AppleScript Folder Actions .....                | 233        |
| Learn More About AppleScript .....                  | 235        |

|                                                         |                   |
|---------------------------------------------------------|-------------------|
| <b>Script the Command Line with Shell Scripts .....</b> | <b>237</b>        |
| Create Your Own Shell Script .....                      | 238               |
| Try Another Script .....                                | 241               |
| Shell Scripts Outside the Shell .....                   | 241               |
| <b>Run Actions Automatically .....</b>                  | <b>243</b>        |
| Use Login Items .....                                   | 244               |
| Use Calendar Events .....                               | 247               |
| Use Launchd .....                                       | 249               |
| <b>Use Omni Automation .....</b>                        | <b>264</b>        |
| What Omni Automation Can Do .....                       | 264               |
| Try Omni Automation in OmniGraffle Pro .....            | 267               |
| <b>Use a Macro Utility .....</b>                        | <b>271</b>        |
| Control Your Mac with Keyboard Maestro .....            | 273               |
| Use Another Macro Utility .....                         | 292               |
| <b>About This Book .....</b>                            | <b>295</b>        |
| Ebook Extras .....                                      | 295               |
| About the Author and Publisher .....                    | 296               |
| Acknowledgments .....                                   | 296               |
| Credits .....                                           | 297               |
| <b>Copyright and Fine Print .....</b>                   | <b>298</b>        |
| <b>Also by Joe Kissell .....</b>                        | <b>299</b>        |
| <b><i>Hazel Coupon .....</i></b>                        | <b><i>301</i></b> |
| <b><i>Keyboard Maestro Coupon .....</i></b>             | <b><i>302</i></b> |
| <b><i>LaunchBar Coupon .....</i></b>                    | <b><i>303</i></b> |
| <b><i>TypeIt4Me Coupon .....</i></b>                    | <b><i>304</i></b> |
| <b><i>Typinator Coupon .....</i></b>                    | <b><i>305</i></b> |

# Read Me First

Welcome to *Take Control of Automating Your Mac, Sixth Edition*, version 6.0, published in January 2026 by alt concepts. This book was written by Joe Kissell and edited by Glenn Fleishman.

This book explores many ways to work faster, increase your efficiency, and have more fun using your Mac by automating common activities. Many of these techniques require no extra software, and nearly all of them are suitable for ordinary users without special technical skills.

If you want to share this ebook with a friend, we ask that you do so as you would with a physical book: “lend” it for a quick look, but ask your friend to buy a copy for careful reading or reference. Discounted [classroom and Mac user group copies](#) are available.

Copyright © 2026, Joe Kissell. All rights reserved.

---

## Updates and More

---

You can access extras related to this ebook on the Web (use the link in [Ebook Extras](#), near the end; it’s available only to purchasers). On the ebook’s Take Control Extras page, you can:

- Download any available new version of the ebook for free, or buy any subsequent edition at a discount.
- Access the book in both PDF and EPUB formats. (Learn about reading on mobile devices on our [Device Advice](#) page.)
- Read the ebook’s blog. You may find new tips or information, links to author interviews, and update plans for the ebook.

If you bought this ebook from the Take Control website, it has been added to your account, where you can download it in other formats and access any future updates. However, if you bought this ebook elsewhere, you can add it to your account manually; see [Ebook Extras](#).

---

## Settings vs. Preferences

---

In macOS Ventura, Apple replaced System Preferences with System Settings, and in most apps, the Preferences menu item and window became a Settings menu item and window. In this book, I sometimes use a shorthand like “go to Mail > Settings/Preferences” or “open System Settings/System Preferences” to reflect both possibilities; when the details are significantly different, I spell them out separately as “Ventura or later” and “Monterey or earlier.” When necessary, I may amplify the text with references to further changes in Tahoe or Sequoia.

---

## What’s New in the Sixth Edition

---

This new edition brings the book up to date with macOS 26 Tahoe, which involved extensive changes throughout. The other major changes are:

- Updated [The State of Mac Automation](#) to reflect the situation as of January 2026
- Added the topic [Let’s Have a Chat About AI](#) to address the use of generative AI in automation, as well as what’s coming soon in Siri
- Edited the [Use Siri](#) topic to correct some small errors and mention the forthcoming Siri update
- Added Snippetty to the list of options in [Use a Third-Party Text Expansion Utility](#)
- Updated [Use Spotlight as a Launcher](#) and added a sidebar, [Spotlight’s New Tricks](#), to cover some of the Spotlight changes in Tahoe
- Added a new topic, [Use Spotlight’s Clipboard History](#), to cover the clipboard history feature Apple added to Spotlight in Tahoe
- Reluctantly added the sidebar [Is Nisus Writer on Its Last Legs?](#) to explain what I know about the state and likely future of Nisus Writer Pro

- Updated the instructions for creating rules on iCloud.com in [Use Server-Based Rules](#)
- Added [Automate MailMaven](#) to describe the automation capabilities in my new favorite email app
- Updated [Log In Faster with iCloud Keychain and Safari Autofill](#) to be current with macOS Tahoe
- Brought the instructions for [IFTTT](#) up to date
- Updated and expanded the Get Started with Shortcuts chapter to introduce new features added in Tahoe, particularly [About Intelligent Actions](#) and [Create an Automation](#)



# Introduction

If you'll forgive the cliché, life is too short. I have long-term goals and dreams that need my attention. And every single day, I have work to do, a family to take care of, books to read, and TV shows to watch. I don't have time to waste on tedious tasks that my expensive, modern Mac is perfectly capable of handling for me.

This book is about taking back your time by *automating* your Mac, which sounds like a fancy and high-tech undertaking. But in fact, all I mean by “automating” is finding shortcuts: ways to do the same thing, only more quickly and easily, with fewer manual steps. That might be as simple as learning a keyboard shortcut for a common command or scheduling a task to run at a certain time. You can accomplish more advanced automation tasks, too, by using macro utilities or writing AppleScripts, but you might be surprised to find that some of the most powerful and effective automation techniques require the least effort and skill to set up. In many cases, you won't even need extra software.

You do *not* have to be a programmer or computer geek to do most of the stuff I explain in this book. I've written this for ordinary readers—smart but non-technical. On the other hand, if you *are* a computer geek, I do mention a few techniques that require above-average technical chops. For any kind of reader, I hope you'll find ideas here you can use to make your work more efficient.

**Tip:** If you *are* interested in learning about programming but don't know where to start, try [Hour of Code](#).

Let me share a story. A certain bookkeeping task used to take me about five minutes a day. After running through it a few hundred times, I decided that I couldn't stand it any longer. So I spent a full day working up a very snazzy macro that combined Keyboard Maestro with AppleScript—two tools I discuss later in this book—to perform the whole task without any intervention at all.

Now, you may be thinking, “That’s madness! Why would anyone spend a whole day automating something that takes just five minutes?” Well, I did it *to save time*. My calculation is that those eight hours are equivalent to five minutes a day for about three months—so after three months of using the macro, I get five extra minutes every single day to spend with my kids. That adds up: it’s more than 30 hours over the course of a year. All of a sudden that sounds pretty smart!

That’s an extreme example, as I would rarely spend so much time automating a single task. But it nicely illustrates my objective: I invest a bit of up-front time to shave off a few seconds here, a few minutes there, from tasks that I perform repeatedly. My work becomes less frustrating, freeing me up to concentrate on more interesting and creative tasks. (And I’m now more proficient, reducing the time it takes me to automate future tasks.) Whether I get more work done in the same amount of time or the same amount of work done in less time, the result is the same: higher productivity and greater happiness.

Many of the topics I discuss are *deep*, and I can only scratch the surface in a book like this. For example, Sal Soghoian and Bill Cheese-man once wrote an 895-page [book](#) on AppleScript, while I spend just a chapter on it. Other Take Control authors penned entire titles about Shortcuts, LaunchBar, and TextExpander, covered in this book only briefly. And I could write hundreds of pages about Keyboard Maestro, Nisus Writer Pro macros, and other topics. Automation is a virtually endless subject, but I’m sure you don’t want to read thousands of pages about it!

Instead, I want to offer you three things with this book:

- Show you lots of automation tools and techniques for your Mac.
- Offer concrete examples you can use as is or adapt to your needs.
- Inspire you with lists of further possibilities and learning resources.

The techniques in this book work with macOS 10.14 Catalina through macOS 26 Tahoe unless otherwise noted. Most of them work in older Mac operating systems, too—although the older your operating system, the less likely you’ll find that everything works as described.

# Automation Quick Start

There are dozens, if not hundreds, of ways to automate your Mac. This book explores many of the most interesting options. I suggest that you first read the [Introduction](#) and [Develop an Automator's Mindset](#). Then dip in wherever you like and jump around to techniques that interest you. If you're interested in ways of automating your Mac that require no extra software, pay special attention to the second chapter, [Use Built-In Automation Features](#).

## *Automation 101:*

- **Where things stand:** Get a snapshot of the changing Mac automation landscape; read [The State of Mac Automation](#).
- **Strategy:** Learn what you can automate and figure out where you can save time and effort in [Develop an Automator's Mindset](#).
- **Built-in features:** Discover the many ways you can make your Mac more efficient using nothing more than what Apple supplies in macOS; see [Use Built-In Automation Features](#).

## *Discover what you can automate:*

- **Input devices:** Get the most out of your mouse, trackball, trackpad, or other input device; see [Automate Your Input Devices](#).
- **Text expansion:** Insert commonly used words, phrases, variables, and even elaborate fill-in-the-blanks reports just by typing a few characters; see [Automate Text Expansion](#).
- **Finder:** Launch apps, open files and folders, play music, look up definitions, perform calculations, and much more with just a few keystrokes; see [Automate the Finder](#).
- **Clipboard:** Cut, copy, and paste like a pro using multiple clipboards, clipboard histories, and tools to manipulate what's on your clipboard; see [Supercharge Your Clipboard](#).

- **Apps:** Use capabilities built into major apps like Word, Excel, Nisus Writer Pro, and FileMaker Pro to automate repetitive or complex actions; see [Automate Individual Apps](#).
- **Email:** Eliminate spam, file messages instantly, send out automatic replies, and more as you [Automate Email](#).
- **Cloud services:** Connect cloud services to each other, create agents that watch the web for information that interests you, and even control your Mac remotely; see [Automate the Web](#).
- **Backup and sync:** Keep your data safely backed up, and optionally keep it in sync across more than one Mac; see [Automate Backup and Syncing](#).

### ***Learn about Apple's automation tools:***

- **Overview:** Read [Discover macOS Automation Technologies](#) for the basics of services, Automator, AppleScript, and shell scripts. Then delve into full chapters on each:
  - **Services:** [Use Services for Systemwide Shortcuts](#)
  - **Shortcuts:** [Get Started with Shortcuts](#)
  - **Automator:** [Get Started with Automator](#)
  - **AppleScript:** [Get Started with AppleScript](#)
  - **Shell scripts:** [Script the Command Line with Shell Scripts](#)
- **Hands off:** Use Login Items, Calendar events, or `launchd` to [Run Actions Automatically](#)

### ***Use advanced third-party automation tools:***

- **Omni Automation:** Get a taste of the cross-platform, JavaScript-based automation capabilities of apps by the Omni Group; see [Use Omni Automation](#).
- **Macro utilities:** Create sequences of steps you can replay with a click or a keystroke to do nearly anything you could do yourself with a mouse and keyboard; see [Use a Macro Utility](#).

# The State of Mac Automation

When I updated this book to its third edition in 2019, I wrote about huge changes that had taken place in the world of Mac automation in the previous few years. Remarkably, even *more* changed since then, so it's time once again to report and reflect on the good, the bad, and the unknowns of the automation present and future for Macs.

In 2014, when this book's first edition appeared, I thought Apple was likely heading in a direction towards more and better automation capabilities. Various Apple teams seemed to be paying greater attention to automation. Since then, signals have become decidedly mixed.

For starters, Apple laid off their key person behind automation for 20 years, Sal Soghoian, in 2016. Apple didn't replace Sal with anyone else, and, to all appearances, his departure from the company coincided with a growing indifference toward Mac automation at Apple. In a depressing shift, new and updated Apple apps began losing, not gaining, automation capabilities via AppleScript and Automator.

In addition, Apple began removing the Unix scripting languages that had been included with the operating system for eons. First PHP was taken out in macOS 12.0 Monterey, and then in version 12.3, Apple also excised Python. (As I write this in early 2026, Perl and Ruby remain, but Apple could still decide to remove them later.) Although it's still possible to download and install those scripting languages separately, the fact that they're no longer included by default is an extra hurdle to building automations that depend on them.

So that's the bad news. And if I had written this update a few years earlier, I might have concluded that Mac automation as a whole was on a hopelessly downward slope. But then a crazy thing happened: Apple took their popular Shortcuts automation app for iPhone/iPad and

adapted it to work in macOS too, starting in Monterey (see [Get Started with Shortcuts](#)).

Shortcuts lets you create automation workflows in a manner reminiscent of Automator. Indeed, Shortcuts can import (some) Automator workflows, and I wouldn't be surprised if at some point Apple ditches Automator altogether in favor of Shortcuts. Furthermore, the shortcuts you create on your Mac sync to, and run on, your iPhone and iPad—and vice versa—making general-purpose cross-platform scripting on Apple's products a reality for the first time. And, in macOS 26 Tahoe, Apple added some terrific new features to Shortcuts, including Automations (triggers that run Shortcuts in response to some event or condition); see [Create an Automation](#).

All that is excellent, and Shortcuts is in some respects more powerful than Automator. However, it's limited by the relatively small number of app-specific automation actions currently available. To be fair, some apps have huge lists of available actions, in some cases far surpassing their Automator support, if any existed. This includes BetterTouchTool, Calendar, Camo Studio, Contacts, Keyboard Maestro, and Pixelmator Pro.

Generally, though, neither Apple nor third-party developers seem to have put a lot of effort into transitioning Shortcuts into the sort of automation tool that professionals and businesses can count on for mission-critical tasks.

Also in Monterey, Apple began allowing developers to create extensions for Mail using a new architecture called MailKit; then, in Sonoma, they dropped support for the previous add-on architecture, called plugins (see [Automate Apple Mail with Extensions or Plugins](#)). In some respects, that set of moves across two operating systems has sort of been two steps forward, one step back, but it's refreshing to see Apple put real effort into offering developers officially supported ways to add capabilities to Mail, limited though they may be.

Tahoe added some surprisingly useful automation features to Spotlight, including Actions (which perform tasks in other apps or run

# Develop an Automator's Mindset

I think of myself as a relatively lazy person. I mean, yes, I've written over 75 books in the last 30+ years, not to mention hundreds of articles for Macworld, TidBITS, and other sites. And sure, I run a publishing company, teach tai chi classes, maintain indoor and outdoor gardens, and spend a lot of time with my wife and kids. But apart from those minor exceptions and maybe a dozen others, I'm pretty much a layabout.

However, there's more than one way to think about laziness. When it comes to your Mac, a *good* form of laziness is not wanting to spend unnecessary time doing something that's tedious or repetitive. A *bad* form of laziness is not bothering to figure out how to save yourself that effort later on.

I frequently interrupt my work to spend a half hour figuring out how best to automate a task that might take me a minute to do manually. But if I save a minute several times every day, that half hour pays for itself in no time. And from then on, I'm more efficient and happier. Even if a certain automation technique saves only a second or two, those seconds add up in a big way over time.

**Note:** To get a visual sense of how much time you might save in the long run, see the xkcd comic [Is It Worth the Time?](#).

But saving time isn't the only reason to automate activities on your Mac. You'll also make your work more consistent and accurate. Automation can prevent small errors that would otherwise trip you up, and save you the annoyance of looking up forgotten details.

As you read about automation methods, I want you to accept the fact that you'll have to spend (not "waste") time learning, experimenting, and setting things up, and that this work is going to be a temporary



drag on your productivity. Don't worry about it. The end result will make you much more productive. Bear in mind, too, that the effort-to-reward ratio is least favorable at the beginning, as you're getting to know the tools and techniques. But as you gain experience, you'll find that smaller amounts of effort produce greater rewards.

Your biggest challenge may be figuring out what you can automate and how. That's what I want to address in the remainder of this chapter.

---

## Learn the Basic Principle of Automation

---

I alluded to this in the [Introduction](#), but I want to make sure it's clear:

*For the purposes of this book, automation means finding shortcuts—easier ways to do the same things you're already doing regularly.*

I use such a broad and inclusive definition because I want you to get in the habit of looking for easier ways to do things on your Mac without the psychological barrier of thinking that automation is some intensely technical process only a computer geek could grasp.

You probably wouldn't have bought this book if you weren't already interested in simple labor-saving shortcuts: if there's something you do frequently that requires three clicks and you can come up with a way to do it with one click instead, you've saved yourself some effort. Or maybe you have a regular task that normally requires a dozen key-strokes and you can do it with a single menu command instead.

Once you feel confident finding shortcuts like these, you can work your way up to the more conventional sense of automation—setting up your Mac to perform complex sequences of useful tasks without any intervention. These could be tasks you would do anyway, or they could be tasks that would be too complicated, time-consuming, or onerous to bother doing by hand—tasks such as sorting your incoming email, monitoring sales figures, or moving files between disks or machines.

Keep in mind, however, that since it takes some time to set up any automation task, automation saves time only when the activity in question happens over and over again. Suppose an app has a deeply



# Use Built-In Automation Features

Although it may not be apparent at first glance, macOS contains *dozens* of built-in automation features, just waiting for you to make use of them. In fact, later in this book, I discuss numerous ways to take advantage of built-in features, such as:

- [Use Trackpad and Magic Mouse Gestures](#)
- [Use Text Replacement in macOS](#)
- [Use Spotlight as a Launcher](#)
- [Use Services for Systemwide Shortcuts](#)
- [Get Started with Shortcuts](#)
- [Get Started with Automator](#)
- [Get Started with AppleScript](#)
- [Script the Command Line with Shell Scripts](#)
- [Manage Incoming Apple Mail with Rules](#)
- [Search Faster with Smart Mailboxes](#)
- [Log In Faster with iCloud Keychain and Safari Autofill](#)
- [Run Backups Automatically with Time Machine](#)
- [Run Actions Automatically](#)

But in this chapter, I want to introduce you to a core set of built-in automation capabilities that don't fit logically within another topic—or that don't go as far as the more capable third-party tools that I discuss later in this book. Most of these involve things you can do in the Finder or in System Settings/System Preferences, and they're among the easiest ways to start automating your Mac.

---

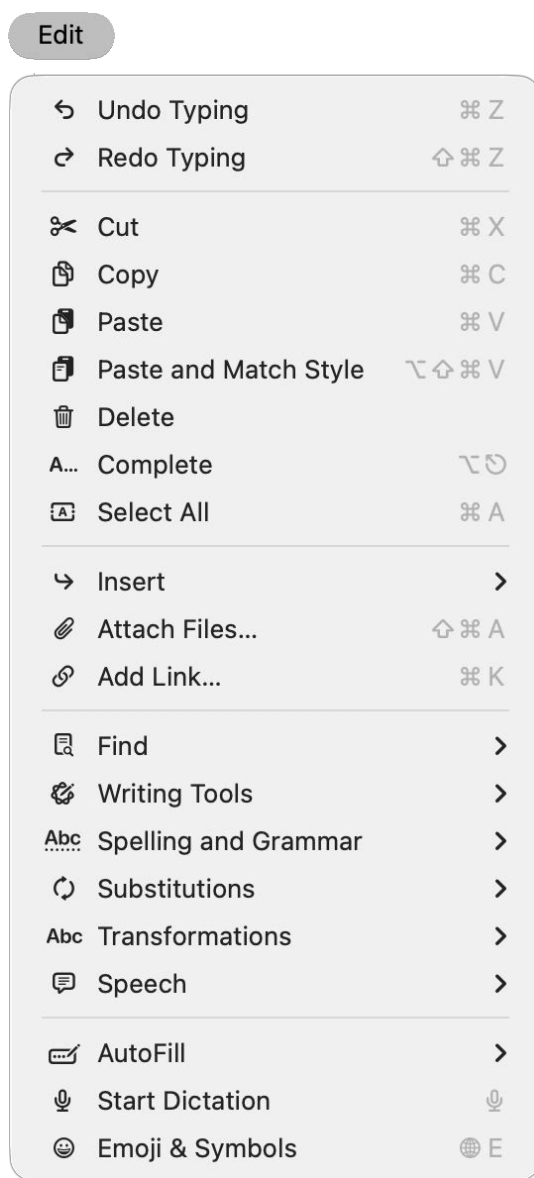
# Use Your Mac's Built-In Keyboard Shortcuts

---

Every app that comes with macOS, including the Finder, has keyboard shortcuts for common commands.

## Menu Shortcuts

The best-known type of keyboard shortcut performs a menu command. You can see the shortcuts right on the menus (**Figure 1**).



**Figure 1:** Examples of menu commands, many with predefined keyboard shortcuts.

# Automate Your Input Devices

We’ve talked about your keyboard already, and we return to it in several future chapters. But I want to take a moment to talk about other input devices, such as the one you use to move your pointer, as well as game controllers and other special-use input devices.

Remember when every Mac came with a one-button mouse? Now buttonless multi-touch trackpads and Magic Mice (with no visible buttons) are *de rigueur*, but it’s still easy to find third-party mice, trackballs, and other input devices with numerous configurable buttons, wheels, and other controls. Even Apple’s minimalist pointing devices can be configured to do special things with gestures and combinations of modifier keys and clicks.

Every extra button or control on an input device can be put to some good use. Although you need not use anything other than a simple keyboard and a pointing device with a single button, you may—depending on your needs, tasks, and disposition—find it easier and quicker to do certain tasks via a dedicated button or knob than with an obscure menu command or keyboard shortcut.

Would you indulge me in a brief story?

I used to manage software development for Kensington, a computer accessories company. One of our products was a four-button trackball called Expert Mouse (or, in some variants, Turbo Mouse). I shared a large office we called the Mouse Lab with three other people—Cris, Debra, and Don. One afternoon when we all should have been busy with more productive tasks, we made up a game that, while goofy, illustrates the kind of thing you can do with a bit of clever automation and a few extra buttons on your input device.

We each started by making rules in Outlook (our email app) to play unique sounds whenever we received an email message from one

another. For example, when I received a message from Don, my computer went Zing! but when Debra sent me a message, it went Pop! Everyone had a custom sound for each other person in the room.

Next, we configured MouseWorks (the software, since superseded by [KensingtonWorks](#), used to control our trackballs) so that each of the three extra buttons—besides the one used for a regular click—sent one of the others a blank email message.

Is your head spinning yet? Well, here's the result of our labors. I click button #2 on my trackball and Don's computer makes a Crack! sound. Don clicks button #3 on his trackball and Debra's computer makes a Ping! sound. Cris clicks button #4 on his trackball and my computer makes an Oof! sound. And so on. So we spent half the day zapping each other with our trackball buttons. You had to be there, I guess, but it was hilarious, like a virtual pillow fight.

That's not a *useful* example of automating input devices, I admit. But perhaps it will inspire you to think up customizations that will make you more productive.

---

## Use Trackpad and Magic Mouse Gestures

---

If you have a Mac laptop with a built-in trackpad, or a standalone Magic Trackpad, you have at your disposal a device that supports not just moving the pointer and clicking, but also scrolling, switching apps, displaying contextual menus, zooming, and numerous other actions by way of gestures such as swiping, pinching, and tapping (with one or more fingers). Apple's Magic Mouse also has a multitouch-capable top surface with support for many (but not quite all) of the same actions.

You must configure your trackpad or mouse with the gestures you want to use—that's the easy part. The harder part is training your fingers to perform these gestures until they become second nature.

# Automate Text Expansion

Even if you're a great typist, you can save time and increase your accuracy by using software that watches what you type and dynamically replaces abbreviations you've previously specified with longer chunks of text. (And if you're *not* a great typist, such software can increase your effective typing speed!)

I mentioned earlier that whenever I type `dttg`, my Mac automatically expands that into DEVONthink To Go. That's just one of dozens of abbreviations I use in my own work. I also use `TCo` to produce Take Control of, `syss` to produce System Settings, and so on. The longer and more complex the text in question, the more useful automatic text expansion becomes.

Text expansion isn't just for names and short phrases. You can use it for addresses, phone numbers, URLs, boilerplate text for common email replies, HTML code snippets, and so on. Depending on which software you use for text expansion, your snippets might also include styles (such as bold and italic), graphics, the current date or time, variables, AppleScripts or shell scripts, the contents of the clipboard, and more.

The great thing about text expansion is that you don't have to do anything special to use it—you simply type. You don't need modifier keys like Command or Control, and you don't need to hunt for menu commands. And it can be used nearly anywhere.

One catch, however, is that you must be careful when choosing abbreviations—since text expands as you type, you might end up making a lame mistake if you're not paying attention. For example, I thought I'd use `km` as an abbreviation for Keyboard Maestro, but then I tried typing a distance in kilometers and got a surprising result! So be sure to use abbreviations that will never occur on their own, or even as part of another word. One technique many people use to solve this problem is to double the first or last letter, as in `kmm` for Keyboard Maestro. Another is to add a slash (/) to the beginning of each abbreviation.

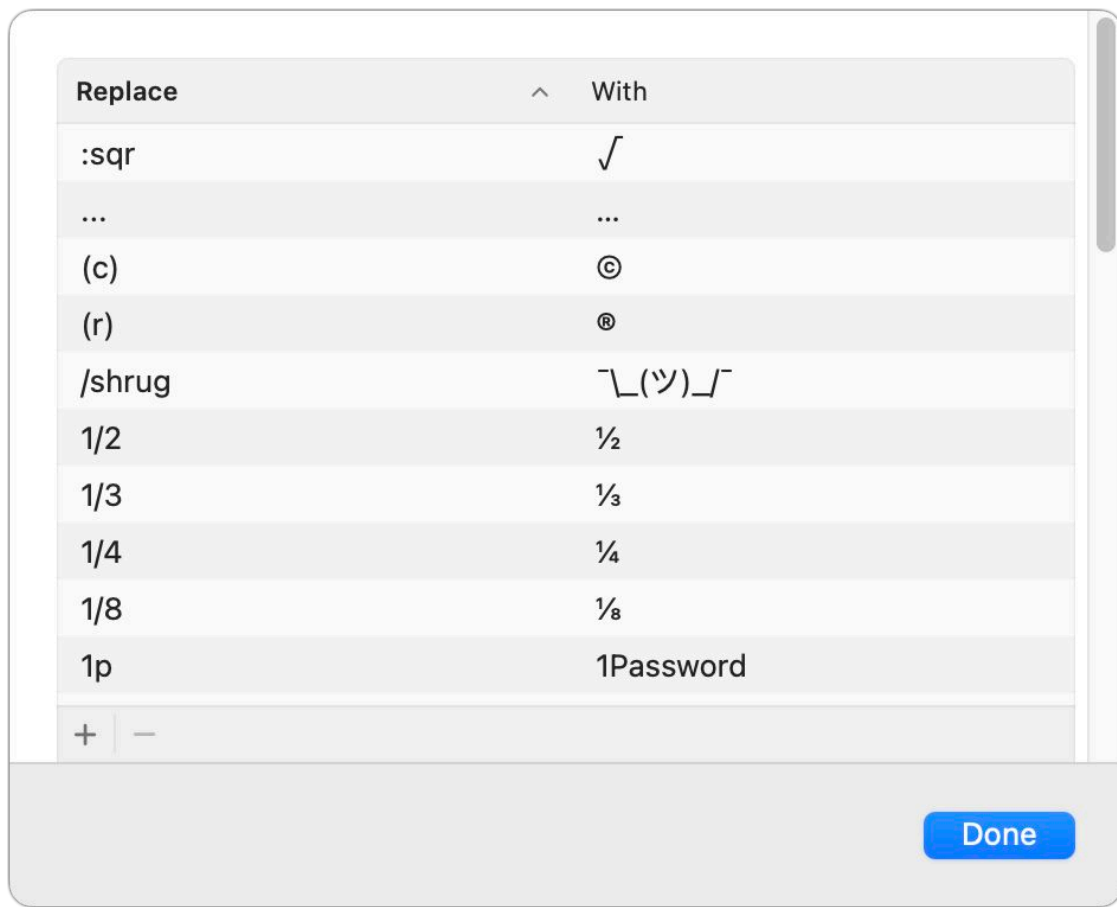
**Note:** Although I say “text expansion,” the very same feature can be used for simple replacements, even if the replacement isn’t *longer*. For example, if you frequently mistype “the” as “teh,” you could use text expansion to replace the latter with the former—or you can correct “MacWorld” to “Macworld” and “PhotoShop” to “Photoshop.”

## Use Text Replacement in macOS

macOS includes its own rather rudimentary text-expansion capability, known as text replacement.

To configure text replacement:

1. Go to System Settings > Keyboard > Text Replacements (Ventura or later; **Figure 29**) or System Preferences > Keyboard > Text (Monterey or earlier).



**Figure 29:** Set up macOS text replacements here. This image shows the window after numerous replacements have been added.

# Automate the Finder

The Finder is a special app that runs all the time and lets you navigate all the files, folders, and apps on your Mac. You use the Finder to organize, tag, and locate files; connect to other devices on your network; mount and unmount servers and external drives; and perform numerous other activities that involve files, folders, and volumes.

The Finder is also the main place people go to open apps that don't appear in the Dock. When you want to launch an app, you might open a Finder window, click Applications in the sidebar, scroll to the app you want, and double-click it. Similarly, if you want to open a document in its default app, you might dig through any number of nested folders in the Finder and then double-click that document.

Since you use the Finder so much, it offers prime opportunities for automation. For example, it's possible to launch apps and open documents without clicking and scrolling through any windows. In a second or less, while keeping both hands on the keyboard, you can open an app or document without even knowing where it's located.

In this chapter, I discuss how to get started automating the Finder using Spotlight, and then I move on to more powerful third-party utilities that let you do even more.

**Tip:** As you're working in the Finder, don't forget about the Quick Look feature. Just press Space with a file selected to see a full-size preview instantly without having to open an app—many common formats are supported. Quick Look also works in most launcher apps.

---

## Use Spotlight as a Launcher

---

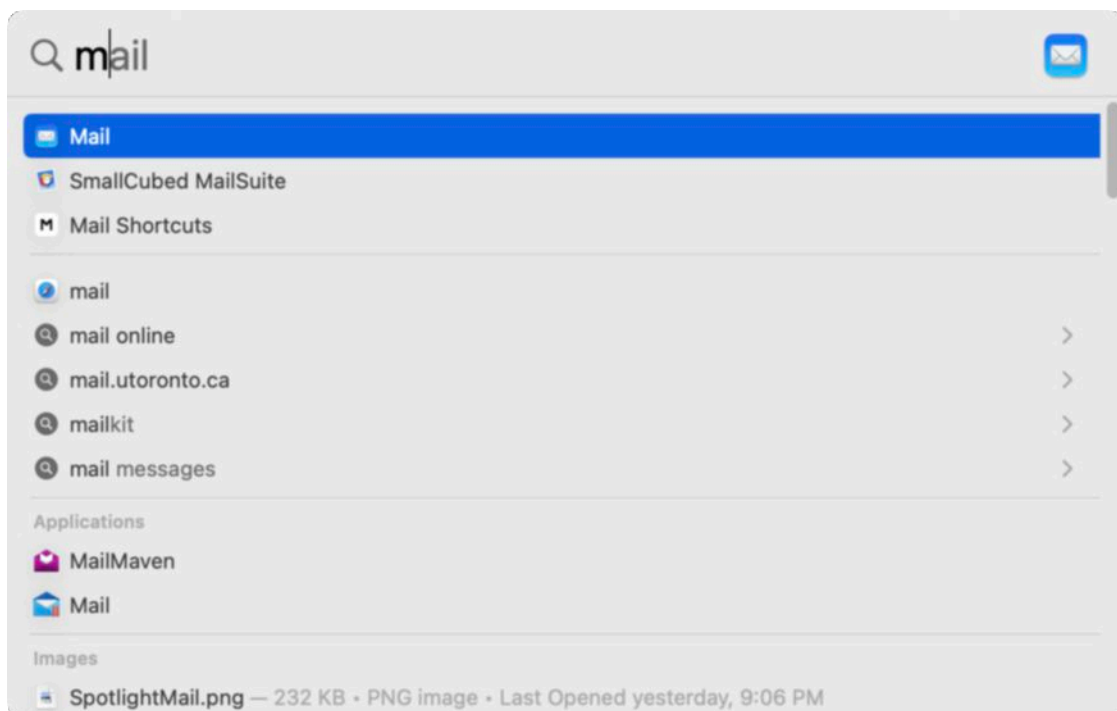
Spotlight, Apple's built-in file indexing and search feature, has always been an excellent way to *launch* apps (and open files) too. You can activate Spotlight with a click on the Spotlight  menu or a keyboard

shortcut, type a few letters of the app you're searching for, and press Return to launch it as soon as it's highlighted.

Spotlight can also be used to search the web, perform calculations, and control other aspects of your Mac's operation that might otherwise require a third-party launcher such as Alfred and LaunchBar (see [Use a Third-Party Launcher](#)) or a macro utility like Keyboard Maestro (see [Use a Macro Utility](#)). It can do even more starting in Tahoe; see [Spotlight's New Tricks](#), ahead.

To use Spotlight as a launcher:

1. Press  $\text{⌘}$ -Space or click the magnifying glass  icon in your menu bar to display the Spotlight window.
2. Begin typing an app's name (**Figure 30**). If the app's name contains multiple words, you can type the first letter of each (such as [du](#) for Disk Utility); you can also type InterCapped letters, as in [bc](#) for BusyCal. As Spotlight finds matching items, it lists apps at the top.



**Figure 30:** The Spotlight window as it appears for me after typing [m](#) in Sequoia and earlier. ([Spotlight's New Tricks](#) covers changes introduced in Tahoe.)



# Supercharge Your Clipboard

Cut, copy, paste. You’ve probably done those things thousands of times without even thinking about your *clipboard*, the temporary storage space macOS uses to hold whatever you’ve cut or copied. Your Mac’s built-in clipboard is boring, but numerous utilities can supplement or replace it with powerful new capabilities that will save you time and effort.

**Note:** iCloud’s Universal Clipboard automatically syncs your clipboard across your devices, as long as they’re signed in to the same iCloud account, have Wi-Fi and Bluetooth enabled, and are near each other. That’s neat, though I’ve found it less than perfectly reliable. See [How to use Universal Clipboard on Your Mac](#) at iMore for details.

With one of these utilities installed, you’ll never again have to worry about your Mac crashing right after you’ve cut or copied something but before you paste it. You’ll also be able to see and use things you copied to your clipboard hours or days ago, change the clipboard contents between the time you copy it and the time you paste it, and more.

Apple has a long history of taking great ideas from third-party developers and integrating them into macOS, sometimes in a smoother way, other times not—and sometimes killing sales of the products they replace. They did this in macOS 26 Tahoe by adding clipboard history to Spotlight. This upgrade is both less powerful and harder to use than the third-party utilities I highlight later. However, if you’re a Tahoe user, you should learn about its capabilities and limitations.

---

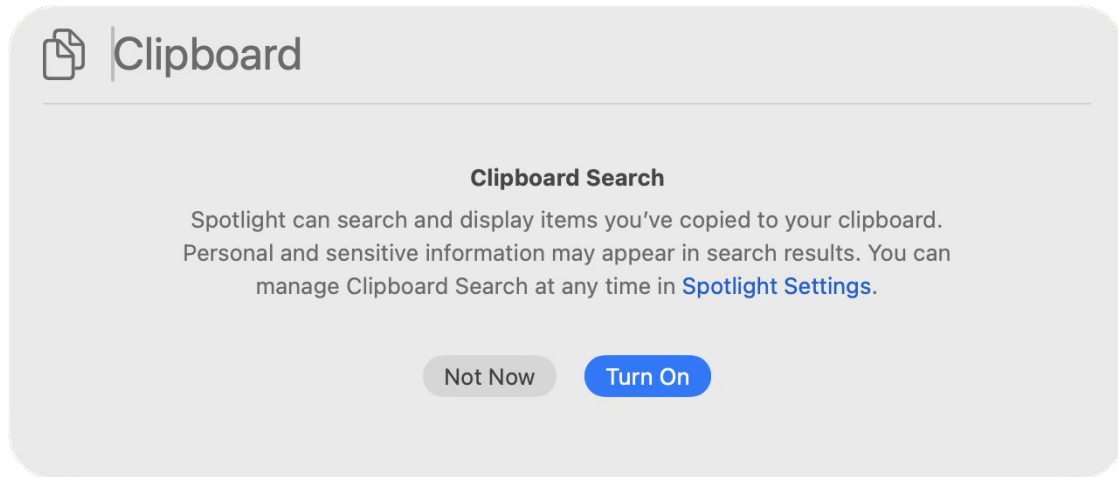
## Use Spotlight’s Clipboard History

---

Starting in Tahoe, macOS can remember everything placed on your Mac’s clipboard, thanks to a new feature in Spotlight. This includes not

only items you copied *on* your Mac, but ones copied *to* your Mac via Universal Clipboard from nearby Apple devices.

To access your clipboard history, open Spotlight and press  $\text{⌘}-4$  (or click the Clipboard  icon). The first time you do this, you're prompted to click Turn On to enable clipboard history (**Figure 33**). Only after the feature is enabled does Spotlight begin remembering what you put on your clipboard.



**Figure 33:** You must turn on Clipboard Search before using it.

**Note:** This topic was adapted from my book [\*Take Control of Tahoe\*](#).

You can choose to keep your clipboard history for 30 minutes, eight hours, or seven days by going to System Settings > Spotlight and choosing the desired time range from the “Clipboard history is available in Spotlight” pop-up menu near the bottom of the pane. In the same spot, you can also clear your history by clicking Clear Clipboard History or turn the feature off entirely using the Results from Clipboard switch.

Once you've turned it on, try copying a few things to your clipboard: text, images, URLs, files in the Finder, audio—you name it. After copying, once again open Spotlight and press  $\text{⌘}-4$ . The window now shows items recently copied to your clipboard (**Figure 34**).

# Automate Individual Apps

Later in this book, I talk about systemwide automation technologies built into macOS like Shortcuts, AppleScript, and Automator, which can automate the actions of individual apps. But there's often a better—or, at least, more thorough—way of doing that within an app itself. That's the topic of this chapter: using apps' built-in automation capabilities.

Due to the breadth and depth of in-app automation features, I can only provide an overview, basic instructions, and a few examples. You'll be able to accomplish some basic tasks and discover how to learn more.

I begin with Microsoft Office, partly because of its popularity and partly because of its extensive built-in programming language. I then move to Nisus Writer Pro, the very app I'm using to type these words, to illustrate a few different forms of automation that should be useful to anyone who works with words. Then I briefly discuss Google Apps Script, a macro language for Google Apps, and list the automation capabilities of several other popular apps.

**Tip:** If you want to automate Apple's iWork apps (Pages, Numbers, Keynote), you'll need to use either AppleScript or Automator. You can learn more about AppleScript for iWork at [iWork & Automation: Productivity Enhanced](#), part of the Mac OS X Automation site.

---

## Automate Microsoft Office

---

Microsoft Office—which in macOS comprises Word, Excel, PowerPoint, Outlook, and OneNote as its main components—is one of the world's best-known software packages. Microsoft long ago added a programming language to the suite called Visual Basic for Applications (VBA), which enables users to write macros that automate Office apps, optionally embed those macros in documents, and run them (with some limitations) on either macOS or Windows. Microsoft removed

VBA from Office 2008 for Mac, but brought it back in Office 2011. It's still there in Office 2024 (also available as part of an Office 365 subscription), and presumably, it will be in future versions too. (Microsoft Office also has excellent AppleScript support, which is an alternative way to accomplish many of the same tasks.)

**Note:** In Office for Mac, only Word, Excel, and PowerPoint support VBA.

What can you do with an Office macro? The sky's the limit, but here are a few simple examples, any of which could be done with a single click or keystroke:

- Perform a frequently used find-and-replace operation
- Format spreadsheet cells according to their contents
- Number all the instances of a certain phrase in a document
- Reformat a table
- Remove all the hyperlinks in a workbook
- Change all the tab stops in the current paragraph style
- Merge cells from two columns into a third column
- Resize all the graphics in a document

If you use Office extensively—and especially if you share documents with Windows users—it might be worth the effort to learn a bit of VBA since (unlike AppleScript) its macros work on both Windows PCs and Macs. But let me be frank: it's not great for beginners. VBA was designed for programmers, not for ordinary users. It won't do you any good beyond Office apps, and unlike AppleScript, VBA would never be called “English-like.” If you don't know much about programming already, there's a significant learning curve.

*However*, there's a sneaky way to get your foot in the door—to write a VBA macro without knowing any VBA at all. Office lets you *record* macros—that is, turn on recording, do some stuff while Office watches,

# Automate Email

Probably 90% of the time I spend using my Mac involves one of four apps: a text editor, a word processor (I am an author, after all), a web browser, and an email client. I send and receive large quantities of email, and I use email far more frequently than phone calls or text messages—maybe even more than speaking. It's my main means of communication.

Because my incoming and outgoing email volume is so high, I can't bear to spend any more time or effort than is absolutely necessary on filing or searching for messages. So I've thought long and hard about how to automate as much of that process as possible—while still ensuring that important messages never slip through the cracks.

One of my key strategies is to use *rules* (sometimes referred to as filters) to process messages as they come in. Each rule looks for certain *conditions* (criteria such as a sender, subject, or words in the message body), and then takes one or more *actions* whenever a match is found. For example, the rule might move the message into a certain mailbox, send an automatic reply, or delete the message.

If you're trying to keep your inbox under control, rules are one of the most powerful tools available. Because I presort my email with rules, tons of messages that don't require immediate attention never reach my inbox at all; instead, they're safely shunted to other mailboxes where I can review them at my convenience. Creating a good set of rules requires a bit of thought and effort, but once you've done that, those rules operate invisibly in the background.

Rules are a very powerful organizational aid, but making them is just one aspect of automating email. You may also want to simplify the manual filing of messages that aren't picked up by rules, add plugins that automate various other email actions, or use smart mailboxes as search shortcuts. I cover all those activities in this chapter.

---

## Use Server-Based Rules

---

Rules can operate either in your email client (such as Apple Mail) or directly on your incoming mail server. The huge advantage to server-based rules is that they can presort messages before you see them, even if your Mac email client isn't running. That greatly reduces the amount of mail you need to deal with on your iPhone or iPad.

I recommend starting with server-based rules if possible and then using rules in your email client for the actions you can't accomplish on the server, like running user-defined AppleScripts on matching messages or moving messages to mailboxes in other accounts.

Check with your email provider to see whether it offers server-side rules or filters, and if so, what the procedure is to configure them. Here's how to get started with iCloud and Gmail:

- **iCloud:** Log in to your account at [iCloud.com](https://icloud.com). Click Mail, and then click the gear  icon at the top of the sidebar and choose Settings from the pop-up menu. Select Rules. Then click Add a Rule to configure your first rule. (After the first rule, click the plus  icon.)
- **Gmail:** Log in to your Gmail account (using [this link](#) or whichever URL you normally use for a Google Apps account with a custom domain). From the Settings  pop-up menu at the top of the page, choose Settings, and then click Filters and Blocked Addresses. Click “Create a new filter” to begin setting up a custom filter.

Although the details vary from one provider to the next, rules always contain one or more conditions (things to search for) and then, when there's a match, perform one or more actions. For example, look for any message from a certain address (say, a company's PR department) and file it in a Newsletters mailbox.

If your email provider doesn't offer server-based rules, or if its conditions or actions don't meet your needs, you can move on to rules in your email client. If Mail is your preferred client, that's just one of the ways you can automate your email.

# Automate the Web

You might not think of web browsing as an activity that requires automation. You follow links, you read articles, watch cat videos, maybe make the occasional purchase, but that's all inherently manual, right? After all, I don't want my Mac to read Facebook posts for me or play games behind my back.

But in fact, the web offers numerous opportunities for shortcuts and simplification. For example, every time you're asked to supply a username and password, a credit card number, or a mailing address, your Mac can do that for you—no typing (or memorizing) required.

Here's another example: you keep checking a certain webpage—or maybe a specific portion of a page—for changes. Maybe you're waiting for an announcement, a sale, or a product update, or maybe you're looking for news stories about your neighborhood. Repeatedly checking a page for changes (whether once a day or several times a minute) is exactly the sort of labor-saving task computers are good at.

And then, looking more generally at cloud services that have a web presence, there are tons of opportunities for connecting things. Perhaps you want to post photos to Facebook after they appear in a shared Dropbox folder. Or save links from your favorite tweets to Instapaper. Or see an alert in the evening if tomorrow's weather forecast calls for rain. All sorts of things that can occur in one cloud service can trigger events in other cloud services—an area ripe for automation.

---

## Log In Faster with iCloud Keychain and Safari Autofill

---

Let's begin with an easy way to automate filling out all those pesky web forms, without the need for any extra software.

**Note:** This topic is adapted from my book [\*Take Control of iCloud\*](#).



The Mac version of Safari (like nearly all web browsers) can automatically fill in your contact information (name, address, phone number, and so on), as well as usernames and passwords, on web forms. Safari uses your Mac's systemwide keychain mechanism to securely store the portions of this data that aren't already in your Contacts app.

iCloud Keychain, included in macOS and iOS/iPadOS, extends this capability. It lets you *sync* a keychain across your Apple devices securely via the cloud. The biggest benefit is that Safari for iPhone and iPad can autofill usernames and passwords that you stored in a keychain on your Mac (and vice versa). But iCloud Keychain also includes:

- A strong password generator built into Safari (on all platforms)
- The capability to store, sync, and enter credit card information in web forms
- Support for multiple sets of credentials per site
- A way to view and remove passwords within Safari (macOS 14 Sonoma or earlier) or Apple's Passwords app (macOS 15 Sequoia, iOS 18, or iPadOS 18 or later), which offers additional features

In addition, if you turn on iCloud Keychain, it automatically syncs the settings for the accounts listed in the Internet Accounts pane of System Settings/System Preferences on your Mac (including email accounts) amongst your other Macs. This account syncing does *not* extend to iPhones and iPads.

**Note:** Starting in Sequoia, iOS 18, and iPadOS 18, Apple started to downplay the term "iCloud Keychain" in favor of "iCloud Passwords" or "Passwords & Keychain," depending on the context.

Starting in macOS 13 Ventura and iOS 16/iPadOS 16, Apple added support to Safari (and iCloud Keychain) for an authentication method called *passkeys*. A passkey uses a clever two-part encryption system instead of passwords and second factors to prove your identity. Once you set up a passkey for a site on one of your Apple devices (which is much simpler than creating a new password), it syncs to all the others too. Then, all you need to do to log in on that site is touch the finger-



# Automate Backup and Syncing

Anyone who has followed my writing for Macworld, TidBITS, or Take Control Books over the last decade or two is undoubtedly aware of my passion for good backups. I've written several books (including [\*Take Control of Backing Up Your Mac\*](#)) and lots of articles on the topic, and I preach about the importance of backups at every opportunity.

In this book, I'm not going to try to convince you to back up your Mac; I'll take for granted that you already know that's a good idea. Instead, I want to focus on *automating* backups. Believe it or not, there are still people who back up important files by dragging them to another disk once a day. Still others use backup software to do the job, but they back up only when they remember to run that software.

My feeling is that if you don't have hands-off backups, you're doing it wrong. Backups should happen all by themselves—whether once a week or multiple times an hour—without any intervention. Not only does it require extra effort to launch a backup app and click a button, it's an interruption—one you might put off if you're too busy, or forget about at a crucial moment right before losing data!

In this chapter, I discuss two backup scenarios: using Time Machine and using a third-party tool that creates versioned backups. You may not use both of these methods, but whichever one(s) you use, they should be automated.

I also talk briefly about automating syncing between Macs. Although that doesn't count as backup in my book, many of the same assumptions apply—and you may even be able to use the same software for both backups and syncing.

---

# Run Backups Automatically with Time Machine

---

Time Machine is the backup feature that Apple built into macOS. It's not necessarily the best backup tool, but it's reasonably good. Most importantly, it's extremely easy to set up, making it the path of least resistance for many users.

Time Machine ordinarily runs once an hour, backing up whatever has changed or been added since the previous hourly run. This happens in the background, with barely any visible clue. So, if you've set up Time Machine already, and you've kept the default options, there's nothing more to see here—move along to the next topic.

If you haven't already set up Time Machine and would like to—or if you configured it but turned off automatic backups—keep reading.

## Configure Time Machine in Ventura or Later

To activate Time Machine in Ventura or later:

1. Go to System Settings > General > Time Machine.
2. Click Add Backup Disk.

A dialog appears (**Figure 44**), listing all local and network volumes eligible to be a destination disk and the amount of free space on each local disk.

# Discover macOS Automation Technologies

As we’ve seen so far in this book, macOS offers lots of ways to automate individual activities. But some automation tasks require apps to talk to each other (or even to other computers), employ sophisticated logic or user interaction, or perform specialized functions that are unique to your situation. When simple tools aren’t up to the job, it’s time to bring in the heavy hitters.

In the next five chapters, I talk about a subset of technologies that go considerably deeper than things like keyboard shortcuts or text expansion. These technologies—Automator, AppleScript, services, shell scripts, and Shortcuts—aren’t so much tools as *platforms* built into macOS that you can use to create your own tools. As such, they’re more complex, but also far more powerful. In this brief chapter, I introduce you to these platforms and offer a bit of high-level advice about how to approach them, especially if you’re a beginner.

I also offer a brief introduction to [Using JavaScript for Automation](#). Although I don’t cover JavaScript extensively in this book, it forms the basis of the automation features in Omni products, which I discuss later (see [Use Omni Automation](#)). And I say a few words about using Apple’s newest programming language, Swift, as the basis for automating your Mac without writing new apps from scratch; see [Using Swift for Automation](#).

---

## Apple’s Core Automation Technologies

---

There’s certainly some overlap among the technologies we cover next. Indeed, I frequently have to flip a coin when choosing which of several approaches I should use to solve a given problem.

So I've chosen to arrange these topics in order of what I consider least to most intimidating. Get to know the ones earlier in the list first, and as your knowledge and skills grow (or your needs outgrow the less-intimidating tools), move on to the next:

- **Services** are plugins that add features for working with text, graphics, and more to almost any app. macOS comes with a number of built-in services, and many popular apps add their own. You can also install standalone third-party services, or create your own using Automator or other tools. See [Use Services for Systemwide Shortcuts](#).
- **Shortcuts** is an app that started out on iPhone and iPad, and appeared on Macs for the first time in Monterey. Like Automator (described next), it offers a simple, user-friendly way to wrap up sequences of tasks into a single automation. It's the hot new thing, and in some ways, it's even more powerful than Automator, but it's not quite ready to replace it yet. See [Get Started with Shortcuts](#).
- **Automator** lets you construct multi-step operations called workflows using graphical building blocks rather than code. (It does, however, let you incorporate code written in AppleScript, JavaScript, and other scripting languages, if needed to solve particular problems.) Automator makes it easy to experiment, and with a bit of creativity, you can create quite powerful workflows that solve everyday problems. See [Get Started with Automator](#).
- **AppleScript** is a language you can use to write programs that do all sorts of useful tasks on your Mac. It's meant to resemble English, but that's perhaps an exaggeration; in any case, AppleScript is certainly more difficult to use than Automator. Even so, AppleScript is far more approachable than heavy-duty programming languages like Swift and Objective-C, while still being quite capable. See [Get Started with AppleScript](#).
- **Shell scripts** run in the Terminal command-line utility, performing tasks using macOS's Unix underpinnings. Because shell scripts have direct access to all the low-level Unix programs that make up the core of macOS, they can solve problems that no other approach

# Use Services for Systemwide Shortcuts

In macOS, a *service* is a special, context-sensitive program that can operate almost anywhere. You can find services on the Services submenu of the application menu (that is, the menu with the current app's name). Alternatively, right-click (or Control-click) something and choose a service from the very bottom of the contextual menu. (Services appear on a Services *submenu* of the contextual menu if more than four of them are active and applicable to whatever you selected.)

**Note:** Quick Actions are a subset of services that includes any service you create as described in this chapter. Starting in Monterey, Quick Actions no longer appear in the Services submenu of the Finder's right-click/Control-click contextual menus, but rather in the Quick Actions submenu. In addition, they still appear in *Application Name > Services*.

If you look at the Services or Quick Actions submenus in different apps, and with different things selected, you'll notice that your choices change. For example, you'll see one set of commands when you select a file in the Finder, a different set when you select text in Mail, and yet another set when you have a graphic open in Preview. That's because each service is designed to operate only on certain kinds of data, or in certain contexts. The idea is to show you only the commands that are relevant to what you're currently doing.

Services, like Automator workflows and AppleScripts (discussed later in this book), can perform complex tasks for you with only one click. So they're fantastic automation tools that every Mac user should be aware of. As we'll see in a moment, you can use a combination of built-in services, third-party services, and services you create yourself to automate a wide variety of activities.

Before you do anything else, you should configure the services on your Mac to your taste. Not all the services installed on your Mac are necessarily active (only active services appear on the Services or Quick Actions submenus); you can enable or disable services as you wish. In addition, you can assign a keyboard shortcut to any service to avoid hunting for it in a hard-to-reach submenu.

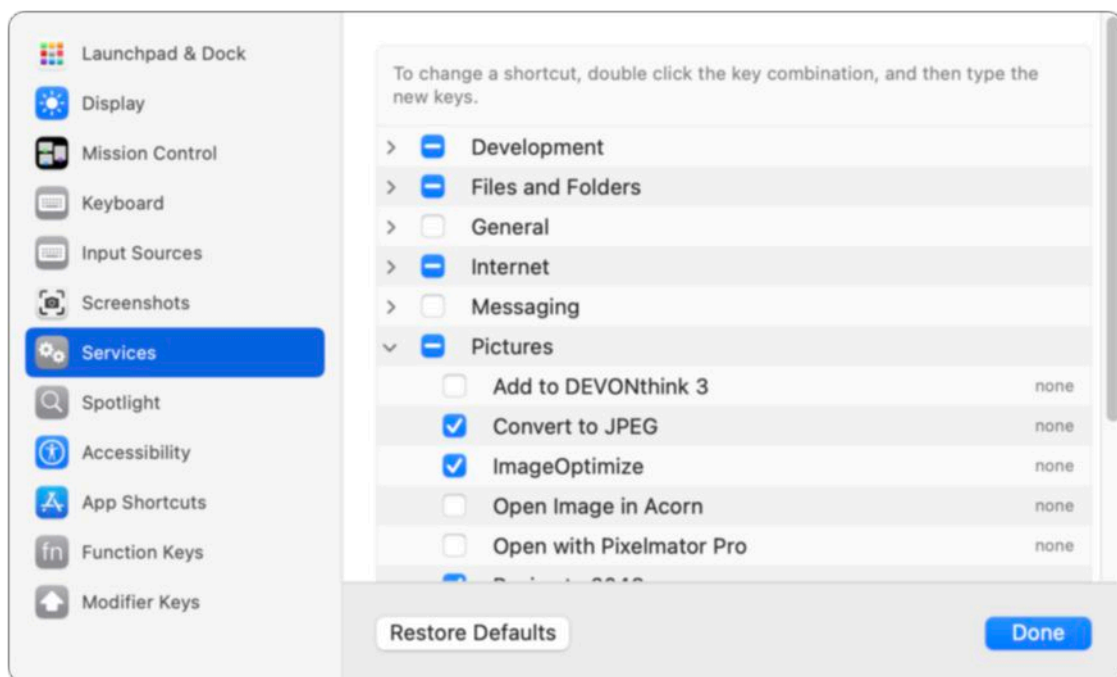
---

## Configure Services

---

To set up Services on your Mac:

1. In Ventura or later, go to System Settings > Keyboard > Keyboard Shortcuts > Services (see **Figure 48**); in Monterey or earlier, go to System Preferences > Keyboard > Shortcuts > Services.



**Figure 48:** Enable, disable, or add keyboard shortcuts to services here.

2. Select the checkbox next to any service you want to enable; deselect those you want to disable. (Remember, selected services appear in the Services or Quick Actions submenu *only* when the context—the app and data type—are appropriate.)

# Get Started with Shortcuts

This entire book is about shortcuts in the generic sense: things that save you time and effort. In this chapter, however, I want to introduce you to the *app* called Shortcuts, and the automations you can create with it—which are, of course, also called shortcuts!

Although Mac users have long had an embarrassment of riches when it comes to automation options, these opportunities on iPhone and iPad were negligible until the release of a third-party app called Workflow. After watching this app’s popularity rise over a few years, Apple bought it, renamed it to Shortcuts, and expanded its capabilities.

Once Shortcuts reached a sufficient level of maturity and usefulness on iPhone and iPad, Apple brought it to Macs, too, starting in Monterey, and added a number of features and refinements in subsequent releases. For the time being, Shortcuts supplements the existing options rather than replacing any of them, though (as I explain shortly) Apple does seem to be treating it as a successor to Automator.

Like Automator, Shortcuts works by stringing together a sequence of actions—some provided by Shortcuts itself, and others provided by various apps. Although the selection of actions and the layout of the screen are different, the basic process is similar. However, Shortcuts offers additional logic, such as an If action and built-in calculation tools, that can make it easier to achieve complex goals without relying on embedded AppleScripts or shell scripts.

Shortcuts for Mac automatically syncs and displays any shortcuts already on your iPhone or iPad. Many of those will work without modification on your Mac.

**Note:** Shortcuts made on iPhone or iPad are compatible only with M-series Macs or, on Intel-based Macs, apps built with [Mac Catalyst](#).



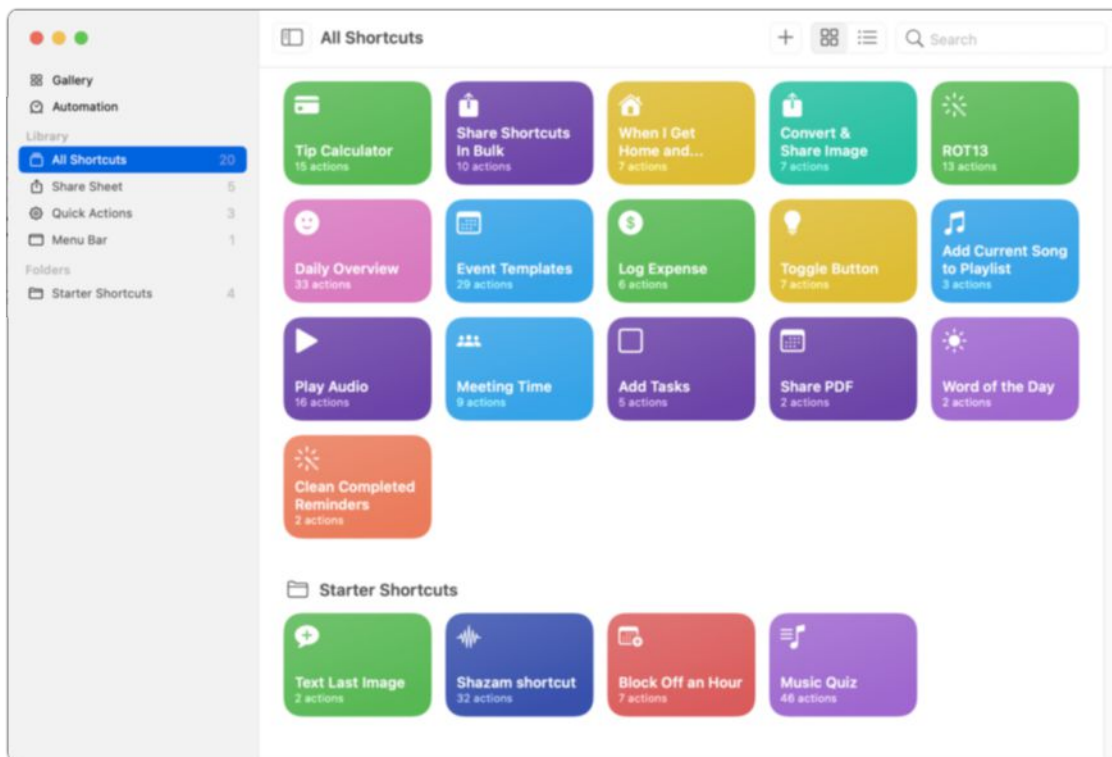
---

## Explore Shortcuts

---

Let's take a moment to look around the Shortcuts app and explore some of the prebuilt shortcuts you can use without any extra effort. This point is pretty important: you can get a lot of value from Shortcuts without ever creating any of your own custom shortcuts. You can also customize these existing shortcuts to meet your needs.

When you open Shortcuts (found in [/Applications](#)), you should see something like **Figure 49**. You will, of course, have different shortcuts than I do, and what you see will depend on which category is selected in the sidebar. **Figure 49** shows All Shortcuts selected, which includes any shortcuts you had on your iPhone or iPad, as well as a Starter Shortcuts category with some examples from Apple. (You can also see those by selecting Starter Shortcuts in the sidebar.) We'll get back to the Starter Shortcuts in a moment.



**Figure 49:** The All Shortcuts view in the Shortcuts app as it happened to appear on one of my Macs.

To see a wide collection of shortcuts supplied by Apple, select Gallery (**Figure 50**). Each of these can be installed with one click.



# Get Started with Automator

Automator is an easy-to-use technology, included as part of macOS, for bundling actions into sequences known as *workflows*. If programming gives you the willies, Automator is nothing to fear. You don't have to learn a new language or write in code—just drag things into a list, fill in some blanks, and check some boxes.

**Note:** Although it's often lumped together with AppleScript (which I discuss in the next chapter), Automator is a completely independent technology that just happens to be capable of many of the same things.

For example, a workflow can:

- Convert text files and graphics into an ebook in EPUB format
- Create a graphic from a word or phrase
- Import all the images from a webpage into Photos
- Create and mount a new disk image
- Convert a movie file to a size and format suitable for your iPhone or iPad
- Create an audio file with a synthesized voice reading the contents of a text file
- Add a 1-pixel border around any graphic
- Upload a file to each of several server destinations—with a different, pattern-based name in each place (that's how we upload Take Control books to the various places they need to go when they're ready for sale!)

In much the same way as in Shortcuts, workflows are made from building blocks called *actions*. Automator includes many actions, as do a number of the applications bundled with macOS (such as Calendar, Contacts, Mail, Music, and Preview). Just as a third-party app may or may not include AppleScript support, some apps come with their own Automator actions and some don't. A few third-party apps with good Automator support are BBEdit, LaunchBar, Microsoft Office, Pixelmator, and Transmit. (In some cases, notably BBEdit, Automator support for an app requires a [separate download](#) and installation.) In addition, actions can optionally contain instructions written in numerous programming and scripting languages, including AppleScript; JavaScript; Perl, Python, and Ruby (if installed); and shell scripts.

I'll show you how to make a few simple Automator workflows. Then I'll tell you about some of your other options, where to find existing workflows that you can use as is or modify to meet your needs, and how to learn more about Automator.

**Tip:** If you're unsure whether Automator or AppleScript is the best tool for a certain automation task, my advice is to try Automator first, because it's so much easier to use. If you get stuck, you can fall back on AppleScript (possibly even including the necessary AppleScript code as part of your Automator workflow).

---

## Create a Simple Automator Workflow

---

For an easy (yet somewhat fancy) introduction, we'll create a workflow that asks you to type some text, and then speaks it back to you:

1. Open Automator (in [/Applications](#)).
2. Click New Document.
3. In the dialog that appears, select Workflow (the default) as the document type and click Choose. Your window should now look something like **Figure 61**.

# Get Started with AppleScript

Whereas Automator lets you construct a workflow visually by dragging and dropping actions into a list, AppleScript is a [scripting language](#)—a type of simplified programming language that runs only in a specific environment (in this case, macOS). That means AppleScripts can run only on a Mac, and because your Mac must interpret the commands in the script as it runs, an AppleScript won't have the high performance of a conventional Mac app. Even so, AppleScripts can look and act like ordinary Mac apps. You may already be using some apps that were written in AppleScript without even realizing it!

AppleScript has been around since way back in 1993, and it's become popular among people who like to tinker but wouldn't consider themselves programmers, because it's built into macOS and is a lot easier to work with than a big, complicated language like Swift or Objective-C. AppleScript is often referred to as “English-like,” which is a generous description at best, but if you don't know much about programming, you can probably make more sense of AppleScript code than, say, Java.

---

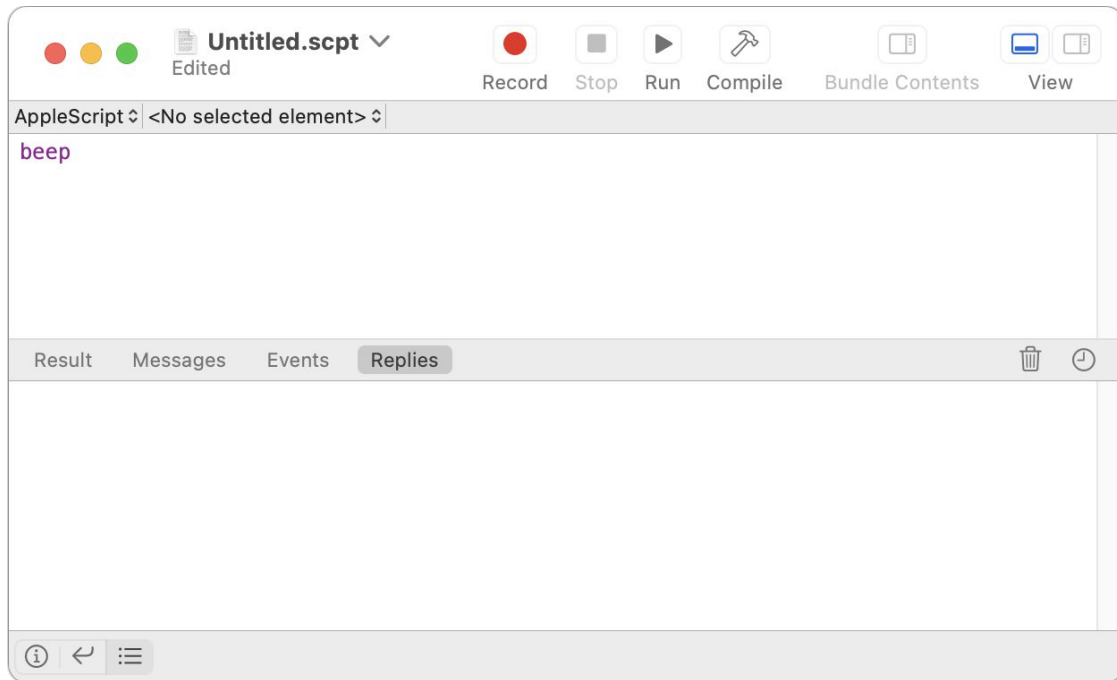
## Write a Simple AppleScript

---

In a moment, I'll give you some examples of what you can do with AppleScript (see [Learn What AppleScript Can Do](#)). But first—before I lose the attention of people who think programming is Just Too Scary—I'm going to show you how to write a complete AppleScript program with *exactly one English word*. Here we go:

1. Open Script Editor in [/Applications/Utilities](#).
2. In the window that appears, click New Document. A blank window opens.

3. Type the word **beep** (**Figure 68**). Your script is now complete!



**Figure 68:** Here's your first complete AppleScript!

4. Click the Run button.

Two things should happen:

- First, you'll hear your system alert sound. That was your program running—congratulations!
- Second, you'll notice that the word **beep** changed its appearance from a purple, monospaced font to a bold, blue, proportional font. That's because when you try to run an AppleScript, the script editor first *compiles* it, a process that checks to make sure it's properly written. If it is, it formats the entire script in an easier-to-read fashion (which will be more apparent with a longer script).

**Tip:** You can manually compile a script (without running it) to check its syntax, reformat it, and add automatic indentation by choosing Script > Compile (⌘-K) or clicking the Compile button on the toolbar.

Now that you've written and run an AppleScript, you can follow the same steps to run scripts other people have written—simply type (or copy and paste) the scripts into AppleScript Editor and click the Run

# Script the Command Line with Shell Scripts

As you probably know, macOS is based on Unix, and as such, there's a whole layer of functionality most users never see. But you can access a long list of hidden files and useful tools in the text-based world of the *command line*.

A command-line interface is a way of giving instructions to a computer and getting results back. You type a *command* (a word or other sequence of characters) and press Return or Enter. The computer then processes that command and displays the result (often in a list or other chunk of text). In most cases, all your input and output remains on the screen, scrolling up as more appears. But only one line—usually the last line of text in the window, and usually designated by a blinking cursor—is the actual command line, the one where commands appear when you type them.

You normally access the command line on a Mac using the Terminal utility (found in [/Applications/Utilities](#)). When you open Terminal, it runs a special program called a *shell*, which interprets the commands you type and delivers the text-based output. macOS comes with several different shells, but that detail is unimportant for our purposes.

What is important is that shells are *programmable*. You can put a series of shell commands in a text file and, without any special fuss, run it as a program. Shell scripts can automate nearly any activity you can perform on the command line. Although some scripts are fabulously complex, running many thousands of lines, we're concerned here with simpler tasks you can automate.

If you happen to be the sort of person who genuinely likes working in a command-line interface, you'll probably want to have lots of scripts that simplify the process for you. But even if you're entirely happy remaining in your Mac's graphical interface, shell scripts can be your

friends. They can modify files and folders, perform system functions, and do other sorts of tricks that are difficult or impossible to perform in other ways (such as using AppleScript or Automator). In fact, some of my favorite AppleScripts, Automator actions, and Keyboard Maestro macros rely heavily on embedded shell scripts.

Here are a few things shell scripts can do for you:

- Modify hidden preferences for macOS and its apps
- Securely delete specific files or folders without first moving them to the Trash
- Force background processes to quit without requiring the use of the Activity Monitor utility
- Load or unload daemons and agents, which control scheduled and background tasks (see Apple’s developer guide [Creating Launch Daemons and Agents](#))

Ordinarily, you create and run shell scripts in the Terminal utility (as I show you next). However, as I mentioned, many of the other utilities I cover in this book can run shell scripts too.

**Tip:** If you’d like to know all about the command line—how to get around, run programs, edit files, and (of course) work with shell scripts, pick up my book [Take Control of the Mac Command Line with Terminal](#).

---

## Create Your Own Shell Script

---

I want to give you a tiny taste of creating your own shell scripts. As with the other topics in this chapter, I’m not going to teach you anything about *programming* as such, just the mechanics of creating and using a simple shell script. I want you to have enough familiarity with the process that you can successfully reproduce and run shell scripts you may run across in magazines, in books, or on websites.

You can create and run a shell script in six easy steps.

# Run Actions Automatically

Recently I was setting up some background apps on one of my Macs to run every time I restart the computer and to relaunch automatically if they crashed. I was struck with the realization that, in earlier editions of this book, I had never explained how to do things like that—for all the examples of automation I gave, nearly all of them assumed that automated tasks would be triggered manually.

Let's correct that! In this chapter I tell you about various ways you can get things to happen *all by themselves* on your Mac: after the initial setup there are zero clicks, keystrokes, or button presses required.

To be sure, other parts of this book already discuss ways of getting certain things to happen on their own. Hazel (see [Organize Files with Hazel](#)) and AppleScript folder actions (see [Use AppleScript Folder Actions](#)) can move or otherwise act on files when they appear in a designated folder. Mail rules (see [Manage Incoming Apple Mail with Rules](#)) run automatically when a new message arrives. Both BetterTouchTool (see [Use BetterTouchTool](#)) and Keyboard Maestro (see [Control Your Mac with Keyboard Maestro](#)), can run macros on a schedule or in response to various system events, as can Shortcuts, starting in Tahoe (see [Create an Automation](#)).

While all those mechanisms are useful, what I cover in this chapter are tools built into macOS that enable you—potentially without installing any extra software—to trigger arbitrary actions of nearly any kind just when they need to happen; and, more specifically, trigger them in situations that may be inaccessible to those other methods.

For example:

- Run a certain program when your Mac restarts, regardless of which user logs in.
- Open your favorite browser when you log in, but not when another family member does.

- Speak a message whenever someone adds a file to a shared folder.
- Copy files for a project you're working on to a network server whenever you mount that volume in the Finder.
- Open the Pages document containing materials for the class you're teaching whenever your office hours begin.
- If your Mac is functioning as a server, make sure any relevant apps (such as a web server or database server) run as soon as the Mac (re)starts and relaunch automatically if they happen to crash.
- Open Excel, QuickBooks online (in a web browser), and Calculator at 9 A.M. on the one or two days a month you have to do accounting tasks for your small business.

In some cases, there are multiple ways to trigger actions like these, and you should pick whichever one you like best. Before choosing a method, I suggest acquainting yourself with all the options.

---

## Use Login Items

---

Let's begin with the simplest approach, which is adding something to a list of items macOS opens automatically when you log in. Chances are, quite a few apps you've installed already have either done this themselves (with your approval) or asked you to do so. You can find Login Items in System Settings or System Preferences:

- **macOS 15 Sequoia or later:** Go to System Settings > General > Login Items & Extensions.
- **macOS 13 Ventura or macOS 14 Sonoma:** Go to System Settings > General > Login Items.
- **macOS 12 Monterey or earlier:** Go to System Preferences > Users & Groups > Login Items.



# Use Omni Automation

Earlier in this book, I covered a number of apps with built-in automation tools, such as Microsoft Office and Nisus Writer Pro (see [Automate Individual Apps](#)). I also told you about systemwide platforms such as AppleScript (see [Get Started with AppleScript](#)) and JavaScript for Automation (see [Using JavaScript for Automation](#)), which can be used both for automating individual apps and for cross-application scripting.

Pro versions of apps made by the [Omni Group](#) can already be automated in any of these ways. Those include [OmniFocus Pro](#) (a task management app), [OmniGraffle Pro](#) (a graphing and charting app), [OmniOutliner Pro](#) (an outlining app), and [OmniPlan Pro](#) (a project management app). Thus it may seem strange that I'm addressing them in a separate chapter.

But I felt it was important to give them extra attention because of a fantastic innovation the company began rolling out a couple of years ago—something they call Omni Automation.

**Note:** A site by Sal Soghoian, [Voice Control and Omni Automation](#), shows you how to use Voice Control in Monterey 12.3 and later with Omni Group apps.

---

## What Omni Automation Can Do

---

In simple terms, Omni Automation is an implementation of JavaScript that (like JXA) permits automation both within and between apps. But it goes further by working *cross-platform* between the Mac and iPhone/iPad versions of a given Omni app. Write a script for OmniOutliner Pro on your Mac, and you can run the same one in OmniOutliner on your iPad (or vice versa). This is significant, because until now, the automation tools available for iPhone/iPad (such as Shortcuts) have been constrained to just the iOS/iPadOS platform. Now, for

the first time, there's a way to get *sort of a little bit close-ish* on an iPhone or iPad to what you've been able to do for years with AppleScript on a Mac—and you can write for both platforms at the same time.

To be sure, I need to make a couple of qualifications here. First, as terrific as this capability is, it's still not nearly as extensive or powerful as AppleScript or JXA on a Mac—partly because of the way iPhone/iPad apps are inherently isolated from each other (for important security reasons), and partly because, as I write this, only three apps offer this feature.

Second, the most exciting part of Omni Automation is what it does for iPhones and iPads—but this book isn't about iPhone and iPad automation, except in a tangential sense.

Even so—and even if you look only at the Mac side—Omni Automation offers some terrific capabilities:

- Third-party developers (including you!) can use Omni Automation to create plugins that add new features to Omni apps.
- Plugins can include scripts called *actions* that are triggered by a menu command, toolbar button, or another script.
- Scripts can also be encoded as URLs, which means that clicking or tapping a link on a webpage, in an HTML help document, or in a PDF document can run a script in an Omni app. URL-encoded actions can also be attached to individual objects in a document.
- Supported Omni apps include a built-in scripting console which lets you write your scripts right in the app.

In case you don't already use OmniFocus Pro, OmniGraffle Pro, OmniOutliner Pro, and/or OmniPlan Pro—or if you do, but you're having trouble visualizing how these capabilities could be put to practical use—let me offer some examples of what you could do with Omni Automation:

- Import list or tabular data from a webpage into a single- or multi-column OmniOutliner Pro document.

# Use a Macro Utility

Earlier in this book, I discussed Shortcuts, Automator, and AppleScript, three tools that can control numerous other apps and tie multiple actions together into easy-to-run shortcuts. All those technologies are powerful, free, and included with macOS.

But AppleScript’s learning curve precludes casual use, while it’s limited by the capabilities various apps choose to expose. Automator and Shortcuts are far easier for a beginner to use, but they, too, have fairly constrained palettes of capabilities—and not all the tasks you might wish to automate fit their “workflow” mold. Meanwhile, apps like Excel and Nisus Writer Pro have fantastic automation capabilities built in, but they’re largely confined to activities within those apps.

So we come to a category of automation tools that—at the risk of overstating my case—transcends these limitations. If you just want to get the job done—not necessarily in the most programmatically elegant way but in a fast, reliable, and flexible way—you want a macro utility. It’s the sort of tool I reach for most often for general-purpose automation tasks.

Like other kinds of tools covered in this book, the idea of a macro utility is straightforward. You pick an action, or a series of actions, from a list; these form the macro’s task. Then you pick one or more events to trigger that action—a keyboard shortcut, a button click, a change in network settings, or whatnot. That’s it: you have a macro.

What’s interesting about the utilities discussed in this chapter is that the lists of potential actions they offer as building blocks for macros are long and diverse. Some of these actions, similar to AppleScript verbs and Automator actions, directly control a particular app (Music, Safari, the Finder) or send instructions to macOS (shut down, change display brightness, switch users). Others manipulate behind-the-scenes resources (clipboards, variables, strings) or manage the flow of steps (if/then/else conditionals, loops, subroutines). Still others “play”

the visible interface, simulating button presses, menu commands, keystrokes, and mouse movements.

Put all this together and you have a toolkit that—with a bit of cleverness and patient testing—can automate almost any repetitive Mac task that doesn't require creativity or human intuition. Here are just a handful of examples, all of which can be done with a single click or keystroke:

- Remap keys on your keyboard to perform different functions
- Show the screen of a shared Mac
- Force a “stuck” Trash to empty
- Add keyboard shortcuts to menu commands in apps that don't support your Mac's built-in shortcuts
- Create an ad hoc Wi-Fi network
- Open an entire set of apps and documents
- Resize and reposition all your windows so they don't overlap
- Modify text or formatting according to predefined patterns
- Email the URL of the web page you're currently viewing to someone else
- Rotate, flip, resize, or crop all the images in a folder

Having sung the praises of macro utilities generally, I must level with you. For all practical purposes, we're talking about one utility: Keyboard Maestro. Sure, I'll mention a few other apps—see [Use Another Macro Utility](#)—and I noted earlier that BetterTouchTool has macro-like capabilities (see [Use BetterTouchTool](#)). Those other apps definitely have their place, but if you want a great macro utility for your Mac, Keyboard Maestro is (in my professional opinion) the best option by far.

# About This Book

Thank you for purchasing this Take Control book. We hope you find it both useful and enjoyable to read. We welcome your [comments](#).

---

## Ebook Extras

---

You can [access extras related to this ebook](#) on the web. Once you're on the ebook's Take Control Extras page, you can:

- Download any available new version of the ebook for free, or buy a subsequent edition at a discount.
- Access the book in both PDF and EPUB formats. (Learn about reading on mobile devices on our [Device Advice](#) page.)
- Read postings to the ebook's blog. These may include new information and tips, as well as links to author interviews. At the top of the blog, you can also see any update plans for the ebook.

If you bought this ebook from the Take Control website, it has been automatically added to your account, where you can download it in other formats and access any future updates.

### More Take Control Books

This is but one of many Take Control titles! We have books that cover a wide range of technology topics, with extra emphasis on Macs and other Apple products.

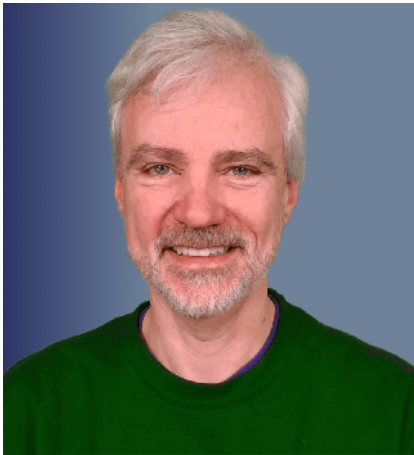
You can buy Take Control books from the [Take Control online catalog](#) as well as from venues such as Amazon and the Apple Books Store. But it's a better user experience and our authors earn more when you buy directly from us. Just saying...

Our ebooks are available in two formats, PDF and EPUB, which are viewable on any computer, smartphone, tablet, or e-reader. All are DRM-free.

---

## About the Author and Publisher

---



Joe Kissell is the author of more than 75 books about technology. In 2017, he also became the publisher of Take Control Books, when alt concepts—the company he runs along with his wife, [Morgen Jahnke](#)—acquired the Take Control imprint from TidBITS Publishing Inc.’s owners, Adam and Tonya Engst. Before he began writing full-time in 2003, Joe managed software development, a *totally normal* profession for someone with degrees in philosophy and linguistics.

In his hypothetical spare time, Joe likes to travel, walk, cook, read, and practice tai chi. He lives in Saskatoon, Saskatchewan, Canada, with Morgen and their sons. To contact Joe about this book, [send him email](#) and *please* include [Take Control of Automating Your Mac](#) in the subject. You can also sign up for [joeMail](#), his low-volume mailing list, follow him on Mastodon ([@joekissell](#)) or Bluesky ([@joekissell.com](#)), or visit his blog at [JoeKissell.com](#).

---

## Acknowledgments

---

Bill Cheeseman and Sal Soghoian offered invaluable input and suggestions for an earlier edition of this book. I also appreciated the helpful feedback of all the technical reviewers of the first edition—especially Peter N Lewis and Greg Scown. And I’m grateful to all the developers who generously provided the coupons at the end of this book. Thank you!

---

## Credits

---

- Publisher: Joe Kissell
- Editor: Glenn Fleishman
- Cover design: Sam Schick of [Neversink](#)
- Logo design: Geoff Allen of [FUN is OK](#)

# Copyright and Fine Print

*Take Control of Automating Your Mac, Sixth Edition*

ISBN: 978-1-990783-84-5

Copyright © 2026, Joe Kissell. All rights reserved.

[alt concepts](#), 419 8B-3110 8th St. East, Saskatoon, SK S7H 0W2 Canada

**Why Take Control?** We designed Take Control electronic books to help readers regain a measure of control in an oftentimes out-of-control universe. With Take Control, we also work to streamline the publication process so that information about quickly changing technical topics can be published while it's still relevant and accurate.

**Our books are DRM-free:** This ebook doesn't use digital rights management in any way because DRM makes life harder for everyone. So we ask a favor of our readers. If you want to share your copy of this ebook with a friend, please do so as you would a physical book, meaning that if your friend uses it regularly, they should buy a copy. Your support makes it possible for future Take Control ebooks to hit the internet long before you'd find the same information in a printed book. Plus, if you buy the ebook, you're entitled to any free updates that become available.

**Remember the trees!** You have our permission to make a single print copy of this ebook for personal use, if you must. Please reference this page if a print service refuses to print the ebook for copyright reasons.

**Caveat lector:** Although the author and alt concepts have made a reasonable effort to ensure the accuracy of the information herein, they assume no responsibility for errors or omissions. The information in this book is distributed "As Is," without warranty of any kind. Neither alt concepts nor the author shall be liable to any person or entity for any special, indirect, incidental, or consequential damages, including without limitation lost revenues or lost profits, that may result (or that are alleged to result) from the use of these materials. In other words, use this information at your own risk.

**It's just a name:** Many of the designations in this ebook used to distinguish products and services are claimed as trademarks or service marks. Any trademarks, service marks, product names, or named features that appear in this title are assumed to be the property of their respective owners. All product names and services are used in an editorial fashion only, with no intention of infringement. No such use, or the use of any trade name, is meant to convey endorsement or other affiliation with this title.

**We aren't Apple:** This title is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Inc. Because of the nature of this title, it uses terms that are registered trademarks or service marks of Apple Inc. If you're into that sort of thing, you can view a [complete list](#) of Apple Inc.'s registered trademarks and service marks.



# Also by Joe Kissell

Click any book title below or [visit our web catalog](#) to add more ebooks to your Take Control collection!

[\*Mac Basics\*](#): Get to know your Mac inside and out with this detailed guide.

[\*Take Control of 1Password\*](#): Slowed down by entering passwords repeatedly? Learn how to let 1Password do the heavy lifting.

[\*Take Control of Apple Mail\*](#): Learn the ins and outs of Apple's email app for Mac, iPhone, and iPad.

[\*Take Control of Backing Up Your Mac\*](#): Protect your Mac's valuable data from any sort of mishap.

[\*Take Control of DEVONthink 4\*](#): Master this powerful information management tool.

[\*Take Control of iCloud\*](#): Understand the many features, get set up properly, and enjoy iCloud!

[\*Take Control of MailMaven\*](#): Discover how to use this powerful Mac email app from SmallCubed.

[\*Take Control of Solving Mac Problems\*](#): Learn how to maintain, troubleshoot, and speed up your Mac.

[\*Take Control of the Mac Command Line with Terminal\*](#): Master your Mac's command-line interface and learn basic Unix skills.

[\*Take Control of Tahoe\*](#): Discover what's new in macOS 26, and get all the information you need to upgrade safely.

[\*Take Control of Your Digital Legacy\*](#): Make sure your important digital information is preserved for future generations.

[\*Take Control of Your Online Privacy\*](#): Learn what's private online (not much)—and what to do about it.

*Take Control of Your Paperless Office*: With your Mac, a scanner, and this ebook, you'll clear the chaos of an office overflowing with paper.

*Take Control of Your Passwords*: Overcome password overload without losing your cool.