Automator, AppleScript & Services

Apple's specialty has always been taking complex, expensive technologies and somehow making them simple, attractive, and magical. It's done it with video editing, digital photos, DVD authoring, WiFi, Web design, music production, podcasting—you name it.

Popularizing one particular task, however, has continued to elude Apple: *programming*.

Through the years, Apple has introduced various new technologies for helping novices automate their Macs:

- **AppleScript** (born in 1993) was the first Apple tool for automating your Mac. You type out English-like commands in a text file called a *script*, one command per line, and then click Run to have your Mac carry out the result (Figure 7-1, bottom).

  AppleScript is a power user's dream come true, but it's still a programming language. If you want to automate even a simple custom job like converting music files to MP3s, you have to get your hands dirty hunting for the AppleScript command that does exactly what you want—and that can be a real pain.

- **Automator** (born in Mac OS 10.4, Tiger) is a newer program that lets you create your own programs by assembling a series of visual building blocks called *actions*. Drag actions into the right order, click a big Run button, and your Mac faithfully runs through the list of steps you've given it (Figure 7-1, top).

  You have a list of preprogrammed actions at your fingertips, so you never have to do any coding or learn any programming language. Creating the little software robots (called *workflows)* is exceptionally easy.

On the other hand, your selection of building blocks is limited to what other programmers have already written, so Automator workflows are limited in what they can do. You can't automate a complex newspaper layout using Automator alone,

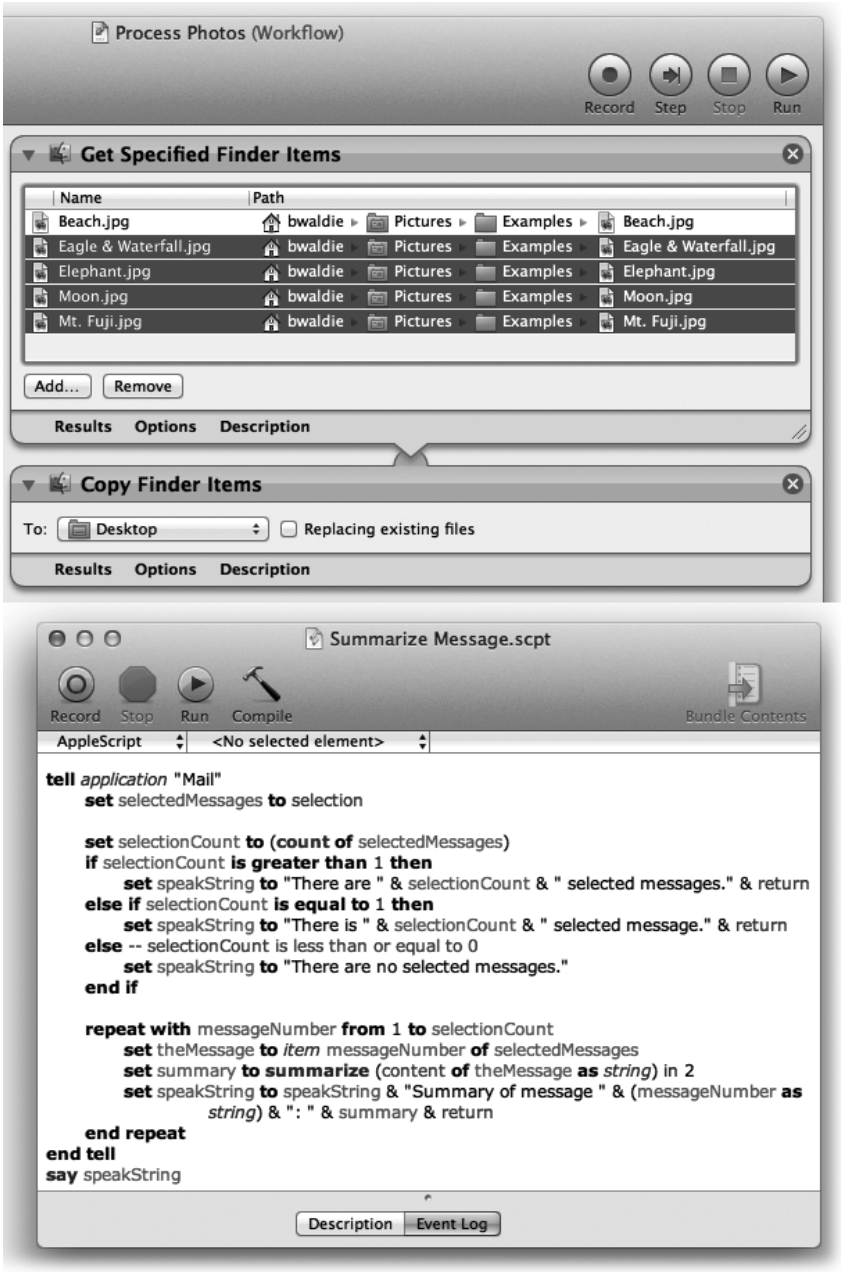

Figure 7-1: *The two faces of Mac automation.*

Top: Automator uses a visual, step-by-step metaphor for automation; the down-pointing arrows indicate how each building block "feeds" its result into the next action. You can usually tell what an Automator workflow does at a glance.

Bottom: Script Editor (the program you use to write and edit AppleScript code) is geekier. There's a good amount of programmerese here, and you have to know basic AppleScript syntax before you can begin to understand this script—let alone write your own.

for example, because nobody has written the building-block actions necessary to control all the stages of newspaper production.

For novices, the most exciting feature might be Watch Me Do mode. It lets you just *do* what you want Automator to learn, as Automator watches and memorizes each keystroke, mouse click, and menu selection. Later, Automator can replay those steps faithfully, like a true-blue macro program.

- **Services** are smart, handy commands, available in most Mac programs and in shortcut menus in the Finder. They're a place to *list* the Automator programs you've created as Services, making them available everywhere—along with a couple of dozen ready-made ones from Apple.

This chapter covers these three build-your-own-software technologies: Services, Automator, and AppleScript. True, the latter two require some technical ability. But even if you consider yourself a technophobe, at least read the section about Services. Some real gems await you, and you don't have to do anything but click them.

Services

For several generations of OS X, a menu called Services has sat in every single program's Application menu. Services are a little baffling, and not many people use them—but here's a crash course.

Services commands are contextual, meaning that they show up only when relevant; when a photo is selected, the text-related commands don't appear. Services can filter out unnecessary material automatically. Run a command on a phone number, for example, and Services are smart enough to ignore any other inadvertently selected text.

You can download tons of Services. To install one, just double-click it; the Mac offers to install it. Once installed, Services appear in a number of places:

- The Services submenu, in the Application menu of every single program.
- The shortcut menu that appears when you right-click or two-finger click a Finder icon.
- The shortcut menu that appears when you right-click or two-finger click highlighted text in a Services-compatible program.
- The ⚙ menu at the top of a Finder window.

Figure 7-2 shows what the Services menu looks like when you've highlighted some text in TextEdit.

You can turn off Services you never use (in System Preferences), thereby hiding them, and you can even assign your own keystrokes to the Services you do use. Finally, you can very easily create your *own* Services menu items, using Automator.

Even so, Services aren't first-class Mac citizens. They show up in the shortcut menus of most of the everyday Apple programs (Safari, Mail, TextEdit, Messages, Stickies,

Terminal, and so on). But weirdly enough, they're not available in any of the shortcut menus of commercial Apple programs, like iLife (iPhoto, iMovie, GarageBand), iWork (Pages, Keynote, Numbers), or the professional apps like Aperture and Final Cut. You can still get to some of them from the Application→Services menu in these programs, though, so all is not lost.

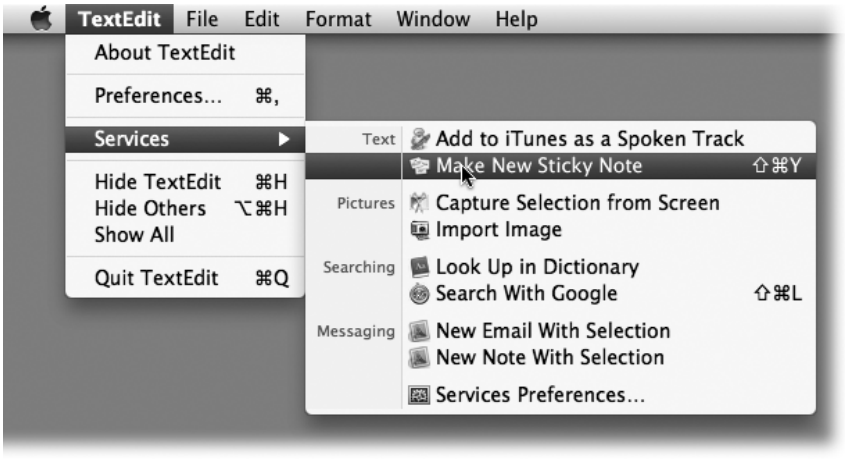

Figure 7-2: *In the Services menu, each command has a tiny icon to help you distinguish it, and the commands are grouped by function. Any keyboard shortcuts you've established appear right in the menu. Services are also available from the shortcut menus and from the Finder ✿ menu.*

Six Services that Come in Handy

If you visit System Preferences→Keyboard→Shortcuts, you see that OS X comes with a big set of starter Services. A lot of them are turned off at the outset. That's because a lot of them are fairly obscure or technical, like "Open man Page in Terminal" and "New Terminal at Folder" Services. Unless you're a Unix pro, you won't find these especially useful.

Here, however, are some examples of the good ones.

Drop in screenshots as you type

You're trying to explain to your mom, or your teacher, or your boss, how to connect to another Mac on the network. What you *really* need to create is an illustrated step-by-step guide.

So you open TextEdit and type up the first step. When it comes time to pop in an illustration, you right-click (or two-finger click) the spot on the page; from the shortcut menu, you choose Capture Selection from Screen.

A message appears: "Drag over the portion of the screen you want to capture." You do so, and *snap!*—a picture of the screen area you selected appears right in your document. You continue on that way, typing and illustrating as you go.

One-click desktop picture

Someone just sent you the most *wicked* cool digital photo. It's sitting on your desktop. Right-click or two-finger click it. From the shortcut menu, choose Set Desktop Picture. *Bam!*—that picture is now your desktop wallpaper, saving you a clunky trip to System Preferences.

Note: In the shortcut menus, Services commands appear at the bottom. If there are more than four of them, they get grouped into a *submenu* called Services.

Text to email, in one step

You're reading on the Web, and you find an article that *totally* settles the argument you've been having with your buddy. Select the text. From the Safari→Services menu, choose New Email With Selection.

Bing! Your Mac opens the Mail program and pastes the highlighted text into the body of a new outgoing email message. You're saved the trouble of copying, launching Mail, creating a new message, and pasting.

Text to a spoken iTunes track

You like to read *The New York Times*' main stories in the morning, but there's never time before work. Or you're supposed to read some document before school. It hits you: Too bad you couldn't *listen* to these documents, read to you on your iPod or iPhone, as you exercise or commute.

Now you can. In Safari, TextEdit, Mail, or wherever, select the text you want converted to audio. Right-click or two-finger click it. From the shortcut menu, choose Add to iTunes as a Spoken Track. In the dialog box that appears, enter a name for the track and choose one of the Mac's built-in digital voices, like Alex, to read it.

Zap! In a flash, you'll find a new file in your iTunes library, ready to sync to your iPod or iPhone. It's an *audio file* of the chosen voice, reading the article aloud.

Make a new sticky note

You find a phone number, address, Web address, sage proverb, whatever, in an email message, a TextEdit document, or on the Web. Highlight it and then press Shift-⌘-Y. *Pow!*—it's now pasted into a new yellow sticky note in Stickies.

(Shift-⌘-Y, it turns out, is the keyboard shortcut for the Make New Sticky Note command in Services. Any Services command can have a keystroke, and you can change them at will—including this one. See page 211.)

Shorten a long-winded writer

Someone has sent you a speech, proposal, or breakup letter. You haven't got time for the pain. "Just gimme the executive summary," you mutter to yourself.

Highlight all the text, right-click it (or two-finger click it), and choose Summarize. (If you don't see Summarize—or another one of the Services mentioned here—go

to System Preferences→Keyboard→Keyboard Shortcuts, scroll down, and turn on the checkbox to the left of the missing Service.)

Bing! Your Mac analyzes the sentences you've highlighted and, after a moment, launches Summary Service. This little program, which you probably never even knew you had, displays a greatly shortened version of the original text. Figure 7-3 offers details.

Note: Lots of add-on Mac programs stock the Services menu with commands of their own. OmniFocus, for example, adds an option to send selected text to your to-do inbox. OmniOutliner, QuickSilver, TextWrangler, Twitter, and many other programs add to the Services menu, too.

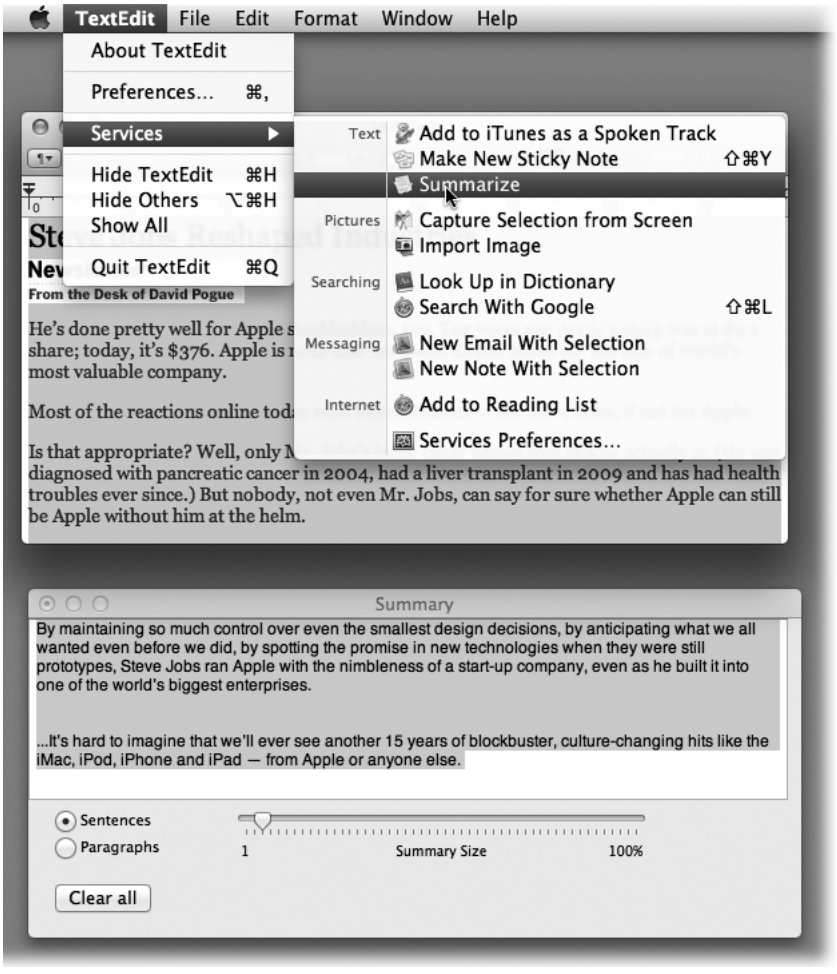

Figure 7-3: *Use the Summarize command to create a one- or two-paragraph summary (bottom) of a longer passage (top). Once the summary appears in the Summary Service program, you can make it more or less concise by dragging the Summary Size slider. You can also ask it to display the most statistically relevant paragraphs instead of sentences, just by clicking the appropriate radio button at the lower left. (Bear in mind that Summary Service doesn't do any rewriting; even OS X can't come up with something coherent if the original wasn't. Instead, Summary Service shows you the most statistically significant sentences.)*

More Great Examples from Downloadable Services

The preceding pages describe the best of the *built-in* Services. But there are hundreds more available on the Web.

Consider visiting *www.macosxautomation.com*, for example. That's the Web site maintained by Apple's Services/Automator/AppleScript team. Here, when you click Services, you can find a Download page filled with useful, ready-made Services. A few examples:

- **Disk Item • New Disk Image with Selection.** Highlight a folder or a group of icons. Choose this command from the ✿ menu, and *presto!*—a new disk image file (.dmg) containing the selected Finder items.
- **Mail • New Message with Video Snapshot.** You open the Mail program and select this command from the Services menu. Your iSight camera comes to life, takes a photo of your mug, and inserts it into a new outgoing message. For added flair, apply a stationery template to the new message before sending. ("Happy birthday, honey—here's a snazzy picture of me!")
- **Image • Change File Type.** Highlight some photo files in a Finder window. From the ✿ menu, choose this command. It auto-converts them to the new graphics file format of your choice.
- **PDF • Encrypt File.** Highlight a PDF document in the Finder, and then choose this command from the ✿ menu or the file's shortcut menu. You get a duplicate of the PDF file, encrypted with a password you supply, with the word "(encrypted)" appended to its name.

Clearly, there's all kinds of fun (and utility) to be found in Services.

Turning Them On and Off—and Adding Keyboard Shortcuts

The Services command center is in the System Preferences→Keyboard→Shortcuts pane. (You can also get there by choosing Services→Services Preferences from any program's Application menu.)

Here turn the checkboxes on or off to choose which Services you want available. (See page 210 for more on homemade keyboard shortcuts.)

Handier still, you can double-click to the right of a Service's name and then press a keyboard combination. That will become its new keyboard shortcut, available system-wide. (The usual caveats apply: Any keystroke you make up *here* overrides the same keystrokes that your programs may use for themselves, so be careful.)

Automator

Automator is the OS X program that lets you create your *own* little programs—and your own Services, if you like. Like most programs on your Mac, it sits waiting in your Launchpad (in a folder called Other). Click its icon to open it. (Automator's robot icon is supposedly named Otto. Get it? Otto Matic? Stop, you're killing us!)

Note: Like TextEdit, Pages, and other showcase Apple programs, Automator offers Autosave and Versions (page 218). Those are especially useful in a programming app like this; often, you'll get things working just fine, and then make some change that causes everything to break. No problem; now you can just revert back to the working version.

Seven Startup Templates

As you'll soon discover, building an Automator workflow is a satisfying intellectual exercise and a delicious talent to acquire. But if the point of all the effort is to create a timesaving, step-saving software robot, you'll need some way to *trigger* it—to run it, just the way you run an email program or a Web browser.

Fortunately, you can save a workflow as a regular, double-clickable application, if you like, or turn it into a Service, as described above, or embed it in shortcut menus all over your Mac. In fact, when you fire up Automator, the first thing it wants to know is: What *is* the desired destination for this workflow?

You're offered seven options in the template screen shown in Figure 7-4:

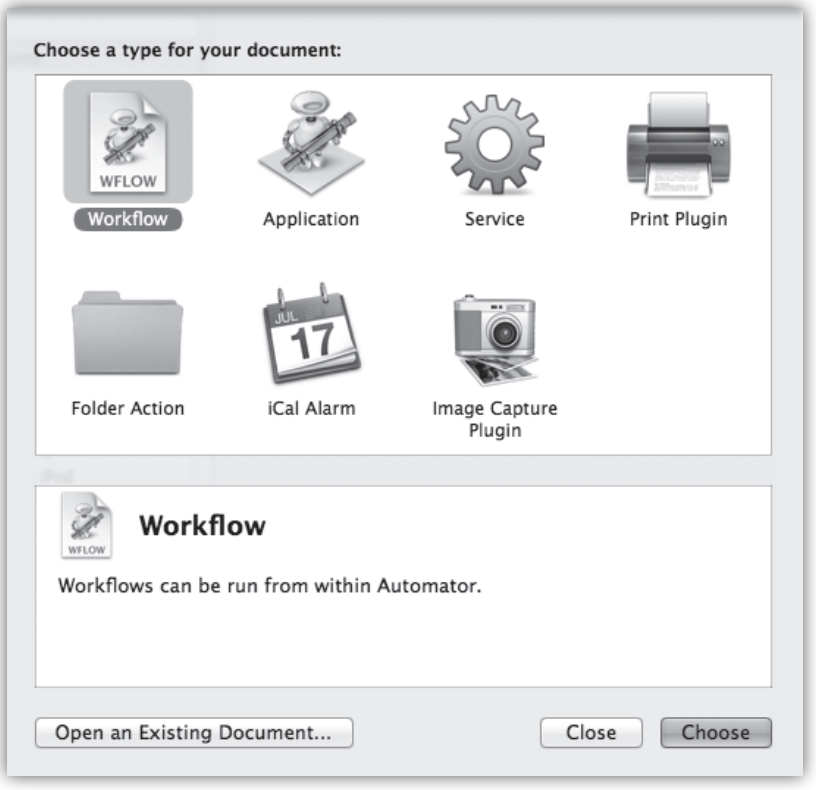

Figure 7-4: *Automator has seven choices for the final form of your workflow. If you're not sure which format you'll want to use for a particular workflow, choose Workflow for now. You can always save the result into a different format (or multiple formats).*

- **Workflow.** An Automator Workflow file (with file name extension *.workflow*) is what you create if you want to study and refine your project. It's not the best format for *using* your homemade app, though, because to run it, you have to open Automator and click the Run button.

- **Application.** A workflow saved as an application acts just like a normal OS X program. You can double-click its Finder icon to launch it, drag it to the Dock for one-click launching, or just leave it in your Applications folder.

  So why wouldn't you just save your workflows as applications *all* the time? First, workflow applications are *much* larger than workflow documents—as much as 20 times larger. That's worth remembering if you plan to distribute your workflows online or via email.

  Also, when a workflow application runs, it doesn't give you access to the Workflow *log*—a big downside if you want to monitor the progress of your workflow in minute detail. Instead, you get a dinky action indicator in the menu bar, as shown in Figure 7-5.

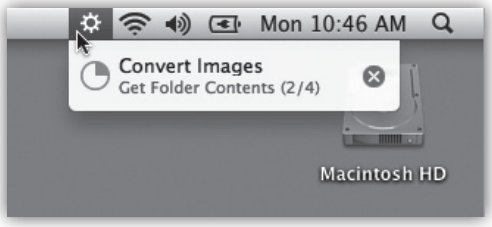

Figure 7-5:
The menu-bar indicator tells you that the workflow application is, in fact, running (the spinning ✿). When you click it, a menu tells you which action is running and how far along the workflow is (the little pie chart). If you want to cancel the workflow halfway through, click the ⊗ button.

Finally, it's a pain to edit a workflow application if something goes wrong; you have to open the whole thing in Automator and resave it.

- **Service.** Yes, that's right: You can now build your own Services, as described above, and give them all the privileges of Apple's own Services. One sample benefit: You can assign your own keyboard shortcuts to them, as described on page 211. The beauty of Automator-created Services is that they show up *instantly* in all the places where Services show up: in the Services menu, in the shortcut menu that appears when you right-click or two-finger click a Finder icon or selected text, and in the Finder's ✿ menu—all ready to use. In other words, enhancing and debugging your workflow is incredibly easy, because the Services you create are immediately ready to try out in the "real world" of your Mac.

- **Folder Action.** Folder Actions are workflows that run automatically when you add files to a *folder* in the Finder. Folder Actions are extremely powerful for image processing, network backup jobs, and much more, but they're also extremely complicated. For an explanation of the ins and outs, read this chapter's bonus AppleScript appendix (on this book's "Missing CD" page at *www.missingmanuals.com)*.

- **Print Plugin.** Print workflows show up as menu items inside the Print dialog box's PDF menu (page 596). If you save a workflow as a Print workflow plug-in, you can easily run the workflow on a document you're about to print—applying a ColorSync filter to the document to compensate for slight discolorations, for example.
- **Calendar Alarm.** This powerful plug-in format lets you schedule workflows to run at specific times. When you save a workflow in this format, Calendar opens and creates a new event named after your plug-in; you're supposed to drag and edit this event to whatever time and day you want the workflow to run. (You can even use Calendar's "repeat" pop-up menu to have the workflow run every day, week, and so on.)

  This is a fantastic tool. If you have a workflow that plays an iTunes song, for example, you can easily build an alarm clock using nothing more than Calendar alarms.
- **Image Capture Plugin.** Image Capture is a program for importing photos from a camera or scanner. If you save a workflow as an Image Capture plug-in, you can choose to run the workflow whenever you import new photos with this program—so you can easily shrink the new images to a smaller size, for example.

  You can click these icons one at a time to read capsule descriptions. You don't have to make a choice now, though; you can start by just saving a Workflow file and then change it to something else later, when you decide what you really want to do with it.

Now, go forth and automate!

Tip: You can find extra Automator resources, including useful workflows and actions, at sites like *http://macscripter.net* and *www.macosxautomation.com.*

Automator Tour

Once you've chosen a finished format for the workflow you're about to create, you wind up in Automator itself. A tour may be in order:

Toolbar

At the top of the Automator window, the toolbar offers six fairly self-explanatory buttons (Figure 7-6).

Tip: To save screen space, you can hide the entire Automator toolbar by choosing View→Hide Toolbar.

Library list

The Library pane is the entire left section of the Automator window. It includes the search box, Library list, Description field, and the list of Actions or Variables. These are described on the following pages.

Tip: You can resize either of Automator's left-side columns (Library and Actions) by dragging the vertical divider lines between them.

Search box

Like OS X's other search boxes, this one works in real time—it displays matching actions or variables as you type.

If you start by selecting a category in the Library list at left, you're telling Automator, "Search only within this category of actions or variables." If you want to search for actions that can process files in the Finder, for example, click Files & Folders in the Library list, and then type *file* in the search box.

Tip: You can tell Automator to group actions by application, instead of category, if you prefer. Choose View→Arrange Actions by→Application. (To switch it back, choose View→Arrange Actions by→Category).

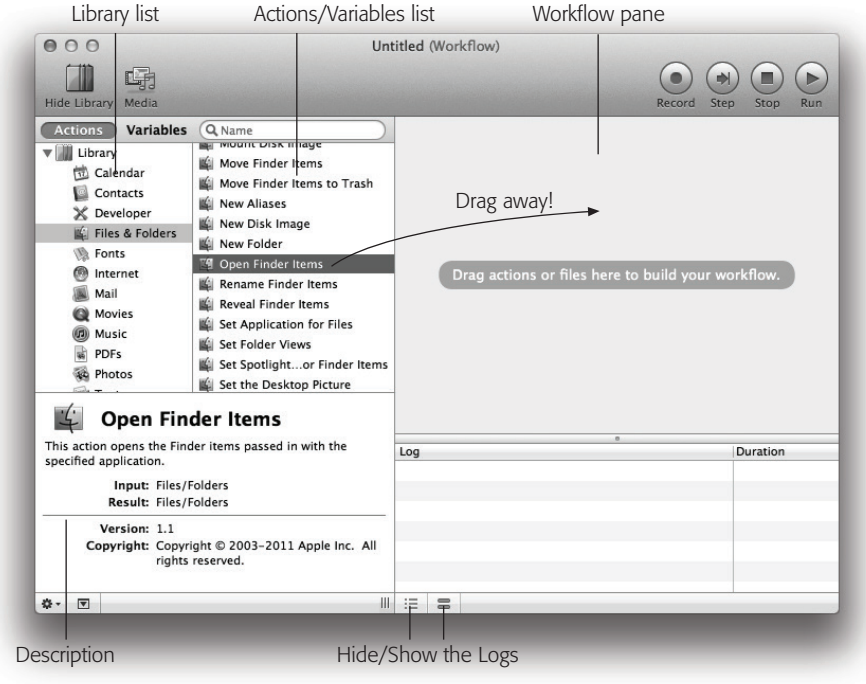

Figure 7-6: *Automator's screen design comes in big slices: a Library list at left, individual Actions in the middle, and the actual workflow-building area at the right.*

Or, to search *all* Automator's categories, click Library at the top of the Library list before searching. (That's the best way to find an action or variable if you're not sure what category it's in.)

Your search results appear in the Actions/Variables list, sorted alphabetically. You can begin dragging actions directly into the Workflow pane at the right side to build a workflow, as described shortly.

Click the ⊗ button in the search box to return to the complete list of actions or variables.

Library

Above the Library list, two buttons appear that govern what's displayed in the Library pane:

- **Actions.** When you click Actions, the Library lists all the features and data that Automator actions can control: Files & Folders, Music, Photos, Text, and so on. (Or, if they're arranged by application: Contacts, Finder, Calendar, Mail, and others.) When you click a category, the Actions list on the right shows every action related to that type of data. For example, when you click Photos in the left-side list, the right-side list of actions offers steps like Flip Images, Crop Images, and so on.

  When you find an action you want to use in your workflow, drag it to the right into the large Workflow pane.

- **Variables.** *Variables* are memorized info chunks that you can reuse in an Automator workflow, exactly as in real programming languages. The Variables list is divided into categories like Date & Time (today's date, today's month, and so on), Locations (the paths to various folders on your Mac), and User (your name, phone number, and other information).

  Later in this chapter, you'll see how it's useful to incorporate these information tidbits into your workflows.

Note: Variables whose icons look like a boxed V are variables that you can *change*, once you've added them to a workflow. For example, you can change the formatting of the "Current time" variable by double-clicking it in the variables log area at the bottom of the Workflow pane.

Variables with ⚙ icons are predefined and unchangeable.

GEM IN THE ROUGH

Open Any Program with a Keystroke

It's a feature that's been achingly missing from the Mac since Day One: assigning keyboard shortcuts to *opening programs*. You couldn't set up Control-W to bring up Microsoft Word, Control-S to open Safari, and so on. (At least not without the help of add-on shareware.)

With Services, now you can.

First, use Automator to build a simple, one-step workflow. The first and only action should be Launch Application. Use its pop-up menu to specify *which* program you want to open from the keyboard.

At the very top of the workflow pane, change the pop-up menus so that they read, "Service receives no input in any application" (where the underlined phrases are pop-up menus). That "any application" part is important; otherwise, your keystroke will work in only one program.

Save the result and name it "Open Word," or "Open Contacts," or whatever you like.

Finally, open System Preferences→Keyboard→Keyboard Shortcuts. Click Services. Find your new Service and give it a keystroke as described on page 287.

And that's it! You've just created a keystroke that can open the desired program—using only OS X's built-in tools. Hurrah!

Description field

When you click an action or variable in the list, the Description box provides some terse, superficial information on how to use it. You might see what the action does, what kind of data it expects to receive from the previous action (*input*), and what the action sends on to the following action (*result*).

If the variable is editable (it has a V icon), you get to see what parts of it you can change.

✿ button

The ✿ button is a pop-up menu. Its four commands let you create and delete customizable collections called *groups* and *smart groups*. They behave exactly like playlists and smart playlists in iTunes. For example, you can create a group that holds the actions you use most often, so you don't have to keep hunting for them.

Smart groups are constantly updated with actions that match the criteria you set for that smart group. (They're available only for actions, not variables.) For example, you can create a smart group that lists only actions that work with iPhoto, or actions with input types that contain the word "image." Add more criteria by clicking the **+** button.

Note: Automator's Library list comes with two factory-installed smart groups: Most Used, which displays the actions you've used the most in your workflows; and Recently Added, which displays actions added by newly installed applications, for example, or actions you downloaded and added yourself.

Workflow pane

The Workflow pane is Automator's kitchen. It's where you put your actions in whatever order you want, set any action-specific preferences, and fry them all up in a pan.

The Workflow pane also shows how the information from one action gets piped into another, creating a stream of information. That's how the Workflow pane differentiates Automator from the dozens of nonvisual, programming-based automation tools out there. Figure 7-7 shows what a piece of a workflow (an action) might look like in the Workflow pane.

When you drag an action out of the Actions list into the Workflow pane, any surrounding actions scoot aside to make room.

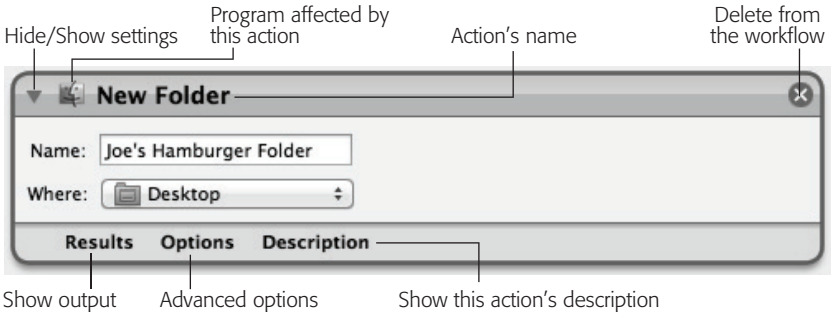

Figure 7-7: *Anatomy of an Automator action. You can drag the action by its title bar to move it to an earlier or later position in your workflow.*

Tip: If you double-click an action in the Actions list, Automator inserts it at the *bottom* of the Workflow pane. (Pressing Return when an action is highlighted does the same thing.)

Log viewer

Under the Workflow pane on the left are two tiny buttons, identified in Figure 7-6. They hide and show two useful pop-up panels that contain *logs* (mini reports): the Workflow log, which shows which actions ran successfully, which failed (if any), what each action did, and so on; and the Variables list (Figure 7-8), which shows all the variables used in your workflow.

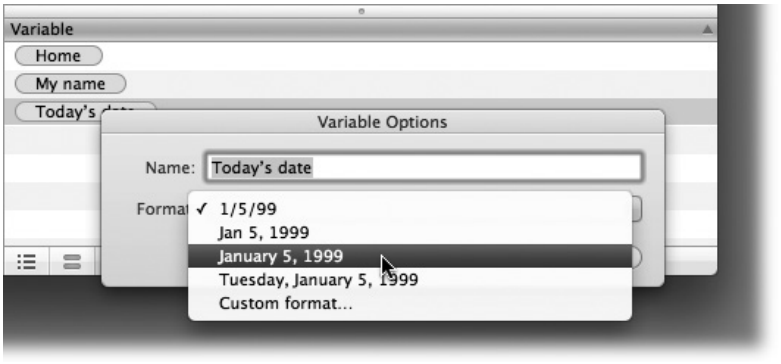

Figure 7-8:
The Variables list shows all the variables you're using in your workflow. The light blue variables are editable; double-click one to modify the variable's name and its settings (in this case, the format for "Today's date").

Opening Existing Workflows

Automator comes with three prebuilt workflows that show off its capabilities. Find them by clicking Help→Open Examples Folder; open one of these workflows for inspection by dragging it to your desktop and double-clicking it. (If you don't drag it to your desktop, thus making a copy of it, before opening it, Automator will start barking that you don't have access to modify the workflow.)

Tip: You don't actually have to *open* an Automator document to see how it's set up. Simply select a workflow in the Finder and hit the space bar to activate Quick Look (page 55), which gives you a preview of the entire workflow—actions and all.

Note, though, that this works only with workflows saved with Automator in Snow Leopard or later. Since Apple's own example workflows were created in the older 10.4 version and never updated, Quick Look won't properly display them until they're opened and *resaved*.

Sadly, the included workflows aren't especially useful. In fact, two of them don't work consistently. Here's a rundown:

- **Process Images** applies special visual effects to whatever photos you specify in the workflow. This is the most useful of these examples. No, people haven't exactly

been clamoring for an automated way to create mirror-image camera shots, but this workflow does show you the basics of batch-processing photos. You can use it as a template for building your own "Convert to JPEG" or "Scale to 640 × 480 pixels" workflow, for example. Later in this chapter, you'll find a complete walk-through of how it works.

- **Import MobileMe Photo Album into iPhoto** didn't work even when the MobileMe service was still around. You can, however, use this workflow on other Web sites that include linked images, such as NASA's Astronomy Picture of the Day (*http://apod.nasa.gov/apod/)*. The concept of the workflow is simple: It slurps linked photos from a Web page and copies them into iPhoto. From there, you can view a full-screen slideshow of the images, or even edit them.

Tip: You can download a *working* version of this workflow (renamed to Import NASA Picture of the Day into iPhoto) from this book's "Missing CD" page at *www.missingmanuals.com.*

- **Display Trailers in Safari** is intended to show you how you can create workflows that work with Web addresses. This one sneaks out to Apple's popular movie-previews Web site, rounds up the list of the latest movie ads, and opens each in a new Web-browser tab.

Tip: Apple provides more Automator examples at its Automator Web site, *www.macosxautomation.com.* "Welcome to the Party!" for example, cleverly demos two complex features of Automator—variables and the Loop action—by showing you how to create a workflow that takes photos of your friends and turns them into a cool, party-ready screen saver. The instructions are a little outdated, but if you can manage to follow along, this one still works like a charm in Mavericks.

Understanding a Workflow

Before you build your own workflows, it's a good idea to understand how actions work together to process information. Here's a step-by-step analysis of the Process Images workflow described above (and in Figure 7-9), which will give you deeper insight into building your *own* workflows.

1. **Ask for Confirmation.**

   This common action, available in the Utilities category (in the Library list), produces a dialog box that tells the innocent bystander what's about to happen (Figure 7-9). It's often smart to begin each of your *own* workflows with a box like this, to remind yourself (or your minimum-wage minions) what the workflow actually does.

   In this case, the message informs your audience that the workflow is about to take some of your desktop pictures, apply some wacko effects to them, and then open them up in Preview to display the results.

Tip: If you want the dialog box to appear with a bright warning sign—to inform you, for example, that you're about to erase your entire hard drive—click the robot icon in the upper-left corner of the Ask for Confirmation action. Automator swaps in a robot-with-yellow-triangle icon.

Keep in mind, too, that the entire Ask for Confirmation action is 100 percent customizable. Not only can you change the text that appears in the dialog box—you can even change the names of the Cancel and OK buttons.

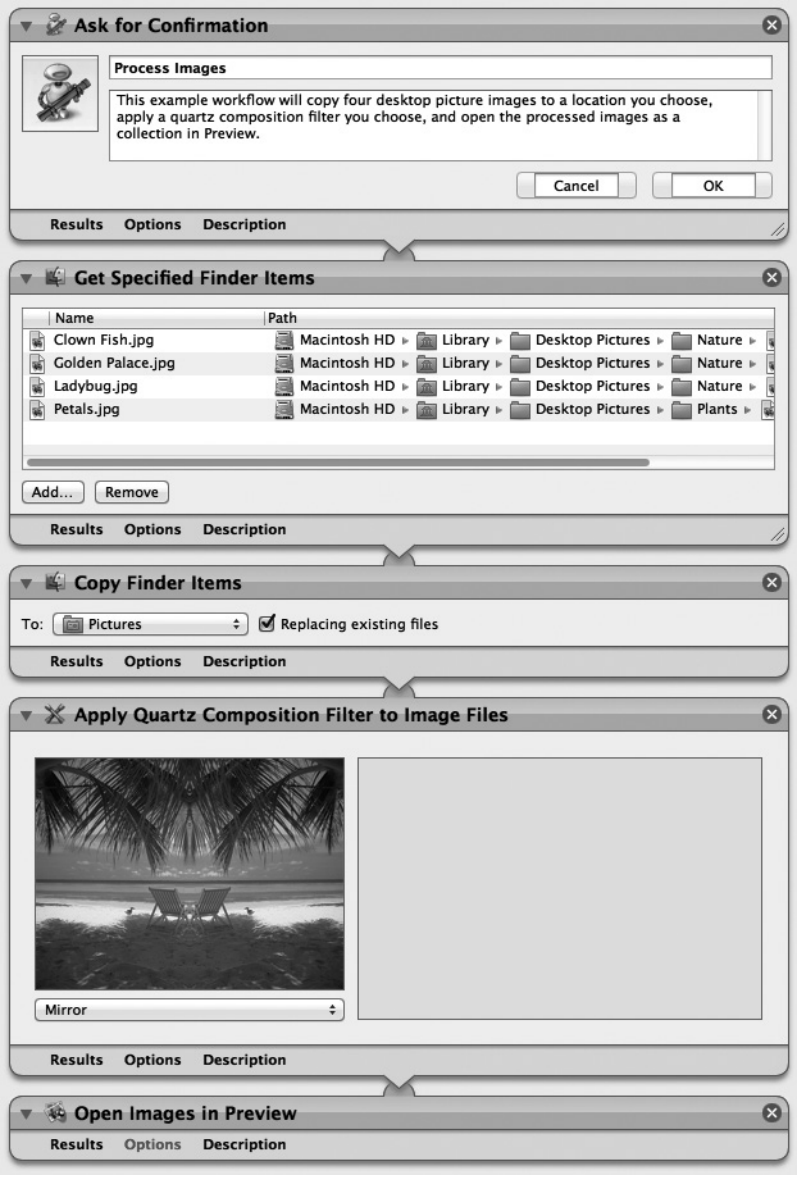

Figure 7-9: *The Process Images workflow consists of only five actions. (The first action, which just displays an explanatory dialog box, shouldn't even count.) Still, this simple action does in 10 seconds what would take most humans at least 5 minutes: applying the same photo effect to several images.*

2. **Get Specified Finder Items.**

   The next step in the Process Images workflow comes from the Files & Folders Library category. Here's where you specify which files you want your workflow to operate on. Apple has set it to use some of your desktop pictures, but you can use the Add and Remove buttons to edit the list—to add your own images to be mirrored, for example—or you can drag files straight from the Finder into this list.

   When this action is finished, it passes on a list of files and/or folders to the next action, ready for further processing.

Tip: This example workflow always operates on the same set of files. But if you were to substitute the Files & Folders→Ask for Finder Items action instead, Automator would *ask* which files to process *each time* you ran the workflow, which is a heck of a lot more useful.

---

3. **Copy Finder Items.**

   This is a very important Files & Folders action: It makes a *copy* of the specified files and folders (in this case, the ones you identified in step 2) so you don't gum up the originals. You can change where you want the copies stored by editing the To pop-up menu in the action. The menu lists obvious locations like Pictures and Desktop, or you can choose Other to select any folder you like.

Note: The "Replacing existing files" checkbox tells Automator that, if there are old files in the Pictures folder with the same names as your new files, you want to delete the old files automatically.

   If you click Options in the action, you see that the "Show this action when the workflow runs" checkbox is turned on. That way, when your workflow runs, Automator will ask where to store the copies so that the destination can be different each time. Otherwise, the files will automatically be copied to whatever folder you select in the pop-up menu right now.

4. **Apply Quartz Composition Filter to Image Files.**

   This action (which came from the Photos category) processes the newly duplicated images from step 3; in this case, it applies a mirror filter to them. The action then passes the newly mirrored images on to step 5. If you prefer, you can choose a different filter from the pop-up menu—to make the image look like a comic-book drawing, for example, instead of applying the mirror filter.

Note: The Quartz Composition Filter, a piece of OS X's Quartz display technology, can modify images and photos in real time. Photo Booth uses this technology, which is why many of Photo Booth's effects are also in the action's Filter pop-up menu. (See *http://developer.apple.com/technologies/mac/graphics-and-animation.html* for the incredibly nerdy details on Quartz.)

   Since "Show this action when the workflow runs" under Options is turned on, you'll have the chance to choose a different filter each time this workflow runs.

(By the way, the image of the beach is intended to demonstrate the filter's effect; it's not actually one of your photos.)

Note: The big box on the right side of the action isn't there because Apple had nothing to fill the space. Certain filters have settings you can modify, which appear in that box. The Glow filter, for example, lets you specify *how much* glow you want applied to the image(s).

---

5. Open Images in Preview.

This final action, which also comes from the Photos category, takes the post-filter images from step 4 and opens them in Preview. From there, you can flip, resize, or resave the images.

Try running the workflow by clicking Run. The bottom of the Workflow pane tells you which step of the workflow is running at the moment. As each action finishes, a green checkmark appears in its lower-left corner.

Note: If something goes wrong while your workflow is running (or if you click Cancel in a dialog box), your workflow stops in its tracks. To identify the offending step, look for the red X in an action's lower-left corner, or check the Log.

Unfortunately, if your workflow shuts down in the middle, you can't restart it from there. When you click Run the next time, the workflow plays from the beginning.

AppleScript

AppleScript is a powerful computer language that's been around since the days of Mac OS 7. If you're an everyday Mac fan—as opposed to some computer-science PhD—AppleScript is by far the easiest programming language to use for automating your Mac.

You can think of AppleScript programs (called *scripts*) as software robots. A simple AppleScript might perform some daily task, like backing up your Documents folder. A more complex script can be pages long. In professional printing and publishing, where AppleScript has hard-core fans, a script might connect to a photographer's hard drive elsewhere on the Internet, download a photo from a predetermined folder, color-correct it in Photoshop, import it into a specified page-layout document, print a proof copy, and send a notification email to the editor—automatically.

Even if you're not aware of it, you use the technology that underlies AppleScript all the time. Behind the scenes, numerous components of your Mac communicate by sending *Apple Events,* which are messages bearing instructions or data that your programs send to one another.

AppleScript has several important advantages over Automator—not the least of which is its even greater power. It comes with a dedicated program just for writing out these scripts. It's the AppleScript Editor program that's sitting in your Applications→Utilities folder.

Still, AppleScript is a *very* deep subject—so deep, in fact, that you'd need an entire book to do it justice. This chapter is an appetizer; a book like *AppleScript: The Missing Manual* is the seven-course meal.

Tip: You can also download an entire chapter about AppleScript from this book's "Missing CD" page at *www.missingmanuals.com.*

Two Sample AppleScripts

Here are a couple of very simple AppleScripts, just to give you a taste:

Open a folder

The first one opens your Applications folder. Sure, you can do that in the Finder easily enough, but this one works no matter what program you're in.

FREQUENTLY ASKED QUESTION

The Scripts Menu

Hey, what happened to that handy Scripts menu? There were some pretty cool little tools in there!

Don't worry. It's still there; out of the box, it's just hidden. You can turn it back on with AppleScript Editor.

Open the program, go to AppleScript Editor→Preferences→General, and click "Show Script menu in the menu bar" (shown here). If you want to see some sample scripts that Apple has kindly provided for you, click "Show Computer scripts." Voila! The menu is back! Now you've got a nice little place to organize your scripts and workflows (you can add Automator workflows to the Scripts menu, too).

If you've used the Scripts menu in the past, you might remember there were around 100 sample scripts in there. Some of those scripts have gone to the great hard drive in the sky, but there are still some gems, like Convert to PDF (under Printing Scripts), which you can use to change a selected JPEG on your Desktop to a PDF. Some of the missing ones you might remember have been reincarnated as Services. Or they're functions you can now do with Automator, like renaming or renumbering Finder files en masse (check out the Rename Finder Items action). Others are available as Services you can download from *www.macosxautomation.com*.

For an overview of some of the scripts listed in the Scripts menu, download the "What's in the Script Menu" appendix from this book's "Missing CD" page at *www.missingmanuals.com*.

Open up AppleScript Editor (which is in your Launchpad, under Utilities). Type this:

```
tell application "Finder"
activate
open folder "Applications" of the startup disk
end tell
```

The result looks like Figure 7-10.

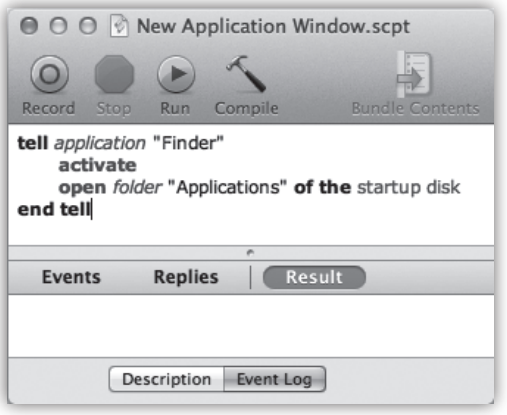

Figure 7-10:
If you've never seen an AppleScript before, you may be surprised at how simple it looks. As you can probably guess from the commands in the window, this script simply opens the Applications folder in the Finder.

Now try it. Click the Run button. The first thing AppleScript Editor does is check your script for typos and syntax; if all is well, you'll see the script format itself, putting the verbs in bold, indenting pairs of matched commands, and so on. If that part went well, the script runs—and your Applications folder appears before you. Woo-hoo!

Now you can save your script (File→Save); choose Application as the file type. In the unlikely event that this was an actual *useful* script, instead of a lame sample, you could stash the new program on your Dock for anytime access.

So what do those commands mean? Here's the rundown:

- **tell application "Finder"** tells OS X which program is supposed to obey the following commands.
- **activate** brings the Finder to the foreground, much as you would by clicking its Dock icon.
- **open folder "Applications" of the startup disk** tells the Finder to open a new window, displaying the Applications folder on your main hard drive.
- **end tell** directs the Finder to go about its regular business, ignoring further AppleScript commands.

Tip: You can also edit this script to suit your needs. Try replacing "Applications" with "Users," for example, to make the script open the Users folder instead.

The metronome script

OS X comes stocked with dozens of programs—everything but the kitchen sink. All right, everything but the kitchen sink and a *metronome.* How are you ever going to play the piano in even rhythm without a steady clicking sound provided by your Mac? Sure, sure, you can use GarageBand's metronome in a pinch, but that's like using an industrial pile driver to kill an ant.

Instead, you can use AppleScript to do the job for you. Open a new document in Script Editor (File→New, or ⌘-N), and type this:

```
display dialog "Welcome to the AppleScript Metronome"
set bpm to the text returned of (display dialog ¬
"How many beats per minute?" default answer 60)
set pauseBetweenBeeps to (60 / bpm)
repeat
beep
delay pauseBetweenBeeps
end repeat
```

Note: Don't actually type the ¬ character. That's programmerese for, "This is really all supposed to be on the same line, but I ran out of space on the page."

When you run this script, you see a dialog box that asks how many beats per minute you want the metronome to tick. Whatever number you type (for example, *120)* gets stored in a variable—a temporary holding tank within the script—that you've named *bpm.*

Next, the script calculates how long it must pause between beeps, and puts that fraction of a second into the "pauseBetweenBeeps." If you told the script to beep 120 times per minute, for example, "pauseBetweenBeeps" would be 0.5, since the script would have to pause half a second between beeps.

Finally, the script creates an endlessly repeating loop: beeping, pausing for the proper period, and then repeating.

Click Run to test out your script, and click Stop when you've had enough beeping.

Note: Believe it or not, even grandpappy AppleScript gets updates from time to time.

For example, AppleScript Editor now lets you start writing a new script from a prebuilt template (choose File→New from Template). Templates are included for writing droplet scripts, file processing scripts, iChat scripts (which have been updated to work with Mail rule action scripts, and some others. It's a great way to get started, but you still need to know enough to fill that template wrapper with the guts of your script.

Furthermore, for the geekiest of the geeky, it's now possible to write Cocoa-AppleScript programs right in AppleScript Editor. Before Lion, you needed Apple's developer tools to do this. Warning: This requires an understanding of AppleScript and Cocoa, the programming technology used to write serious Mac programs. (In fact, it's what Apple's own programs, such as Mail and iPhoto, are written in).

AppleScript vs. Automator

AppleScript has hundreds and hundreds of uses: automating layout workflows that are too complicated for Automator, controlling programs that Automator doesn't recognize, and programming things that Automator can't handle.

Yet if all you do is look at AppleScript as a second choice to Automator, you're missing out on a lot of power. Truth is, AppleScript lets you do more than Automator will probably let you do in the next 10 *years;* it's just a lot more powerful.

In the end, stick with Automator for simple things. If you need to use AppleScript to automate some aspect of your Mac, though, take pride; you're stepping up to a true power tool. (You can even combine the two, building AppleScripts right into your Automator workflows, thanks to the Run AppleScript action.)

Happy automating!