CHAPTER 1

What Is Automation?

IN THIS CHAPTER

Before getting into any of the details concerning Mac OS X and automation, it's time to take a breather and get some perspective. There's plenty of time to go into all the intricacies of AppleScript, Automator, and all the ways to combine the two to automate workflows in later chapters. If this were any other book, then I could certainly get away with just a how-to manual.

But this is a Bible, and that means letting you know not just how-to but also why-should-we, as well as why-do-we. By no means is this book meant to imply that you must do this instead of that because people who do *this* are good, and people who do *that* are bad, but it is meant to be a one-stop, comprehensive look at the topic.

There is going to be just a little bit of preaching, but rest assured, it will all be confined to this section of the book. Many of you are already converted, but some still need some prodding.

After this section, I'll get back to the details, but for now, it is important to understand the whys and wherefores of automation, and that means taking a short trip through history.

Why You Should Use Automation

I'm going to return to an important theme expressed in the Quick Start chapter: Simply put, people are good at some things, and computers are good at others. Humans have reason, intuition, and can apply their judgment, experience, and insight to the data they sense with their own eyes and ears.

Computers, on the other hand, are good at rapid calculation and data crunching; unsurpassed in the areas of storage and retrieval; and very, very good at transforming one set of data into another. What computers can't do very well (yet) are all those things that humans are good at: giving all of that data some context. Your iMac may be very good at resizing photographs, but it can't tell you which photos are the best ones you took on your last vacation.

Fortunately (and unfortunately) for us in the early twenty-first century, computers are everywhere. They're on our desktops at home and work (something that was deemed unthinkable 25 years ago). Portable laptops are finally small and light enough to carry in messenger bags. Processors are jammed into iPods, car engines, traffic control systems, industrial turbines — you name the place, and there's a little electronic brain analyzing input and spewing out data.

What all of this means is that we're faced with a geometric increase in the kinds of data we have to process. Graphic designers may have to process 1,000 images at one go, doing the same tasks over and over again: Resize to no bigger than 300x300 pixels, grayscale, rename the file, and then create a thumbnail. System administrators have to pick through 100,000 lines of log files to figure out what happened if a server is hacked. Marketing professionals are confronted by thousands of data points from all across the business — Web analytics, open rates from e-mail marketing campaigns, number of Twitter followers and retweets, network analyses of their LinkedIn and Facebook groups, and all of it compared to their sales pipeline and forecasts.

It's easy to forget that there's a purpose to all of this data: What we're meant to do with the data is to make decisions and gather insight, not just wallow in it. But there's so much data, so many files, so much to do that those of us without the proper skills in automation, well, we just sift data through a sieve all day long.

I once sat and watched as a young graphic designer, working on an iMac, went through the painful manual processing of thousands of images. It took her most of a day to perform all of her tasks, with an average processing time of two to three minutes per image. I've never seen anyone so beat from a full day of sitting at a desk in an air-conditioned office. I won't even talk about how sore her mousing hand was!

My job at the time didn't include intervention, just assessment, so all I did was report back to the agency she worked for. And she was by no means the exception to the rule. Just about every single graphic designer in that department had little or no exposure to automation, even though every single Mac came with Automator and AppleScript installed.

Simply adding a very short Automator workflow to their process would save countless person hours of image processing. When I suggested this change to management, I got the immediate response: "This will save us a lot of time and money, but what do we do with these people now that they have all this free time?"

I was astonished by the question. My response was curt and immediate: "Put them to work doing something valuable! They're all graduates of design programs. Put them to work designing great things for your clients." Any low-level intern with minimal training can spend their entire day crunching through a series of repetitive tasks. It isn't fulfilling, it doesn't add that much value, and therefore, it needs to be automated as quickly as possible.

It's important to remember, though, that process automation is by no means an end unto itself. Some tasks cannot be fully automated, nor should they be. The point is to provide repeatable, automated workflows where they make sense. It's about leveraging the strengths of the computer, particularly in the arenas of quick calculation and task completion, such that the human has more time to do what they do best: Apply expertise, intuition, and skill.

In the case of the designer working at the agency, yes, of course, automation will mean fewer billable hours for resizing digital images, but it will also mean a lot more time left over each day for client meetings. It's the face-to-face time with clients, the back and forth, the design process that makes a designer a more effective professional, not the time spent head down with repetitive tasks.

And it's not just about making creative people more productive, either: Think about all the accountants, programmers, and other left-brainers out there. Come tax time, accountants can benefit greatly from adding automation to their work processes. And there's hardly a single programmer left on the planet who hasn't benefited from some kind of automation, whether it is in build management or some other area. The same goes for those who have to manage a lot of people and the data they produce: sales managers, casino floor bosses, storeowners, and more.

A Brief History of Automation

If you look up the word *automation* in Merriam-Webster's Collegiate Dictionary, you get something close to the following definition:

1. The technique of making an apparatus, a process, or a system operate automatically. 2. The state of being operated automatically. 3. Automatically controlled operation of an apparatus, process, or system by mechanical or electronic devices that take the place of human labor.

The term automation itself was coined in 1912, but it wasn't really popularized until after World War II. Of course, other related terms, such as automatic (as in *semi-automatic pistol* or *automatic reflex*), have been around long before the advent of computers (the former since 1898, and the latter since the late 1600s). The ancient Greeks had a word, *automaton*, that described a self-moving, self-operating apparatus (what we would call a robot); later on, this word evolved into the French *automatique* in the seventeenth century to denote a mechanism designed to follow a predetermined sequence of operations.

So, well before the computer age, we had this notion of machines or systems doing repetitive tasks. That's because anyone who sat down to analyze the kind of work people did day in and day out would recognize that properly automating a process greatly increases the speed at which the work is done and reduces the error rate.

Differences between mechanization and automation

In areas of agriculture or industrial production, mechanization is often conflated with automation, but the two aren't the same. *Mechanization* is all about giving human operators the tools and machinery to assist them with physical labor — for example, in the early days of industrialization, a steam-powered tool like a lathe dramatically decreased the time it took to turn out table legs, but the lathe still required a human operator. The same goes for water-powered mills, windmills, or foot-powered threshers. These mechanized industries make you more productive, but they don't necessarily involve automation.

Automation, on the other hand, is a step beyond mechanization and is all about reducing the need for human intervention. It's about requiring less human sensory and cognitive input into some kind of process, system, or task. An automated teller machine is a good example: Here you have a networked appliance that allows users to carry out basic transactions without the need for a human teller. The use of sophisticated lab equipment to automate the analysis of genes, cells, and tissue samples has led to many incredible breakthroughs in medical science (not to mention the launch of various blockbuster CSI franchises on network television). Autoresponder e-mail systems are yet another example of how a small business can use technology to provide better service and response times to customers.

Regardless, automation and mechanization are most linked to the Industrial Revolution. That this wasn't your typical revolution with armies and uprisings is beyond the point — it was a paradigm shift, one that started innocently enough. First you had the use of the steam engine in the 1760s to drive machinery, mostly in the textile industry, and then that turned into a whole bunch of applications across a wide swath of human work.

Of course, some historians have pointed out that you had all kinds of mechanization long before then, ranging from the aforementioned windmills to foot-powered threshers, but from a historical perspective, the Industrial Revolution wasn't just about mechanization; it was about how we organized ourselves to work. Over the past three centuries, there's been an inexorable shift from small villages and farms to urban centers, from handcrafted goods to mass-produced widgets.

Since 1946, automation has increasingly been applied to the world of information technology, data processing, and computers. Each year, the amount of data grows and grows, and so does the need for improving how we store, handle, secure, and transform (and any other verb you can think of) all of this data. Without automation, without some tools for sifting through or processing data, we would never get to the point where we could actually add value to the system: giving that data some context, turning it into knowledge, and transforming that knowledge into action.

In industrial contexts, the focus of automation has shifted over the past 20 years from mere quantity to more focus on quality output and flexibility on the line. Whereas hand-installed pistons for truck engines might involve a 1-percent to 1.5-percent error rate, automating that process has reduced errors to 0.00001 percent, according to Wikipedia.

Negative connotations of automation

Of course, you can't have a discussion of automation without mentioning the naysayers, those who, like the ad agency referenced earlier, argue that "increased automation puts people out of work." It's an old argument, one that we can take back to the Luddites of the early nineteenth century.

The Luddites were textile artisans who protested the use of mechanized looms. The looms, they argued, put too many skilled artisans out of work, because they could be operated by relatively unskilled (and cheap) labor. The Luddite response to the paradigm shift of mechanization was to wreck the machinery, which led to a harsh retaliation by the British government (the ringleaders were executed publicly). Some historians argue that the Luddites were anti-technology and anti-progress; others argue that they were just a pragmatic group of people trying to preserve their way of living (after all, churning out more product meant lower prices and lower wages).

Regardless of your position on the Luddites, there's no stopping a paradigm shift. The factories were coming, and industrialization would not stop short at textiles. The steam engine, the railroads, the steel mills, all the changes that would come would lead to more men leaving their villages and seeking work on the factory floors. That in turn would lead to more products out there in the world, and to a higher demand for services related to those products, as well as to more people being involved in the paperwork of commerce.

More recently, it is true that manufacturing jobs fled North America during the 1980s and 1990s, but that's only partially due to the rise of automation. Yes, we have more robots and automatic assembly lines putting products together, but we've also seen a shift in consumer values: We want cheap products. Cheap products don't get made by expensive workers in expensive plants. They get made in dollar-a-day factories in China.

Running parallel to the manufacturing decline, we've experienced a giant boom in the IT sector. Those jobs pay just as well, with great benefits and safer work environments than their counterparts in manufacturing. In fact, we've added entire sectors of professional specialties unimaginable in the 1940s and 1950s. For example, in a typical corporation you have business system analysts, enterprise architects, solution architects, customer relationship managers, Web designers, event coordinators, online community managers, intellectual property lawyers, and a whole host of other specialties. The only drawback is that all of these specialties require special skills and education; it isn't as easy as graduating from high school, walking across the parking lot to the tractor plant, and applying and getting a job as an apprentice welder or line worker.

Positive changes due to automation

You could argue that the widespread use and availability of technology today, and a professional technical class who know how to use that technology, has led to an unprecedented number of freelancers and consultants who have economically viable (and personally fulfilling) careers without much need to work for the Man.

In turn, this shift has created a hyper cottage industry of professionals that are turning out great work. It doesn't matter if the end product is a Web site, a business management assessment, or a carbon footprint calculation; what we're seeing as a result is a return to a kind of work we haven't seen since the giant factories started pulling millions of people away from their villages and farms and into the urban areas.

What we're seeing, in fact, is technology driving yet another paradigm shift. Increased automation is allowing more and more of us to make a living how we want to, using tools like Automator and AppleScript to do the work of 5, 10, or 15 people. It's kind of wonderful how the cycle has turned, that we're at this point where a highly trained artisan (for example, a consultant with 15 years' experience) can be as productive as 10 to 15 low-skilled workers.

Examples of Automation

It's time to leave the dry discussion of history to the past (no pun intended). Let's talk about something more relevant to what you're trying to do right now. You might be wondering what you can automate, and a good way to go about that exercise is to look at the list of tasks, processes, and systems that others have been able to automate successfully.

Of course, this entire section will be circumscribed by a simple filter: I'm not here to talk about automation of car factories, as I assume you don't have one of those handy. Instead, I'll focus your attention on how computer-based tasks have been automated by people and small businesses. The discussion will be largely limited to AppleScript and Automator, but keep in mind that any tool or system (for example, shell and Perl scripts, proprietary vendor tools, and others) that has simplified a process or system can also apply.

Data entry or collection

It's not something you think about every day, but one of the original boons of business computing was moving processes from paper-based to computer-based forms. In many cases, all that was needed was physically laying out the paper form onto the screen. Error rates fell because data entry clerks didn't have to parse out horrible handwriting. In many cases, a simple data entry screen removed various steps from an impossibly complex workflow, providing more cost savings.

Later on came further advances in usability, giving us more powerful and intuitive electronic data gathering systems that took full advantage of the medium. Why ask for a user's name and other personal data if they've already logged in and therefore identified themselves? Why ask for a date when the system already knew the date?

Data collection grew in other areas as well, as more and more applications joined the landscape, each with their own logging tools. Soon we were awash in error logs, messages, and all manner of electronic notes about all kinds of system, application, and network events. Soon we had more data from systems and applications than we did from humans. A single day on the stock market could generate more data than a month of operations in a pre-computer age corporation.

Data analysis

All that data meant an increased need for automated analysis. We needed some way to sift through all of that data in a quick fashion, looking for rough patterns, and trying to mine insight from the slew of messages. If we knew, for example, that an increase in traffic to a Web site at certain times of the day meant something different (say, more traffic from Europe than from Japan), we could make smarter decisions — like offering more Europe-specific messaging or special deals to the content mix of our site.

Of course, there's a big difference between real-time, near real-time, and historic (or even forensic) data analysis, but insight is insight, even for the little guy, because everyone from the small retail shop owner up to the head of a multinational organization now has at their disposal more data than they could ever consume in 20 lifetimes.

Put simply, the best data analysis tools allow the human operator to sift through great quantities of data and reveal patterns, trends, and relationships in that data. Are there certain days when our store makes more sales of certain products? Are there certain times of the day in which more sales occur? Do we make more profit from certain types of work, or when we work with certain types of customers?

Many data analysis tools are used to gain insight into customer behavior and understand what competitors are doing. Many of these tools are plugins or add-ons to existing software like spreadsheet and database programs.

Why is data analysis so crucial? Because good analysis is hard to come by, with too many organizations relying mostly on gut or instinctual calls. Think about the last time any company you worked for went about redesigning their Web site. Did they rely in any part on data analytics? For example, did they optimize for Firefox over Internet Explorer (as revealed by the preponderance of Firefox-using visitors in their log files)? Did they include information on their Web sites based on the actual needs, problems, questions, or issues posed by their customers?

Probably not — they most likely made a series of subjective decisions about their design and hoped for the best. The same is probably true about most of the millions of dollars spent on TV advertisements, but that's a whole other book.

Data munging

Data munging has a long and storied history in the information technology industry. Munging is all about writing simple scripts that do a specific task quickly and heuristically. Some malign munging as being the "poor man's ETL," but sometimes you can't get to the more sophisticated tools without first playing around a bit. And sometimes, there's just no point in building some perfectly automated tool that removes all human involvement from every step of the process.

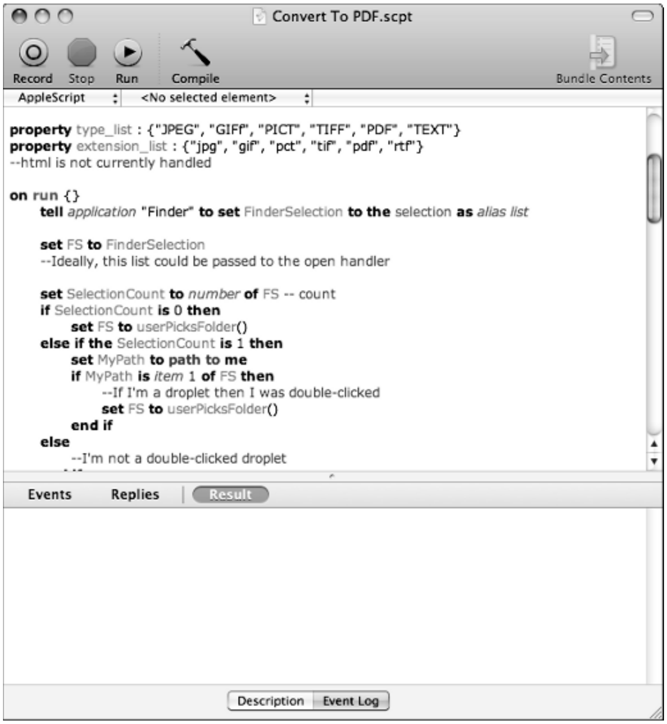
In other words, sometimes all you need is a shovel, not a backhoe. You might need to open 10 log files with 10,000 lines of data each, scan each line for a certain character sequence, and if found, pull out the first 50 characters of each matching line. Then you might need to put all those matching lines into another file where another script, run every night at midnight, might compare those lines to the previous night's log. If a pattern or trend emerges, then that occurrence might cause another script to load certain data into a spreadsheet that is e-mailed to an analyst for the next morning's analysis and briefing session.

Yes, your organization could go out and buy a huge enterprise system that does all this, or you could build the same system using Perl, UNIX shell scripts, AppleScript, Automator, or any other "simple" tools.

In fact, many organizations do just fine with data munging, mostly because they involve small pieces that can be mixed, matched, and adjusted many times without too many adverse consequences. In fact, the example I gave two paragraphs ago might be too involved, as many data munging efforts usually involve extracting data from certain files and then spitting out a result set that matches some simple search criteria. In Figure 1.1, I've included a look at a script that converts different image files to PDF files. Another popular munging activity is transforming one set of data, such as a comma-separated data file, into something else, like a set of SQL queries for database loading.

FIGURE 1.1

Sometimes automation is about converting from one format to another.



```
property type_list : {"JPEG", "GIFf", "PICT", "TIFF", "PDF", "TEXT"}
property extension_list : {"jpg", "gif", "pct", "tif", "pdf", "rtf"}
--html is not currently handled

on run {}
    tell application "Finder" to set FinderSelection to the selection as alias list

    set FS to FinderSelection
    --Ideally, this list could be passed to the open handler

    set SelectionCount to number of FS -- count
    if SelectionCount is 0 then
        set FS to userPicksFolder()
    else if the SelectionCount is 1 then
        set MyPath to path to me
        if MyPath is item 1 of FS then
            --If I'm a droplet then I was double-clicked
            set FS to userPicksFolder()
        end if
    else
        --I'm not a double-clicked droplet
```

A lot of what you're going to do in this book could be categorized as munging. That's okay. I don't consider munging a pejorative term, and neither should you, particularly if a little bit of munging saves you hours of boring work.

Data extraction

Extraction is the E of ETL (extract-transform-load), and it usually involves handling either binary or unstructured data files. In other words, you start with a bunch of images, audio files, word processing documents, calendar events, e-mails, and other data and end up with structured data (database records, XML, SGML) at the other end of the process.

You normally see data extraction working in concert with other steps (such as data transformation, which I'll talk about next), but it's important to understand that automating extraction can be a valuable, beneficial exercise.

For example, one of the more time-consuming extraction exercises is pulling text out of Web pages or PDFs. Many tools (both commercial and open source) have emerged that allow users to quickly scrape the text out of Web pages and PDFs. In fact, Automator offers actions that allow you to do either with relative ease and comfort.

Instead of spending hours manually copying and pasting text from PDFs, now you can use Automator (and other tools) to painlessly grab what you need and then hand it all to another script, action, or tool.

Data transformation

Transformation is the "T" of ETL. Just like extraction, transformation is quite a big deal. In many cases, what you're trying to do is transform one kind of data (comma-separated values or HTML) into another format (SQL queries, XML, and so on). In other cases, you're not worried about converting, but about adding metadata, like you did with the Automator workflow in the Quick Start chapter. Sometimes all you'll be doing is mapping from one type to another — for example, you might have one spreadsheet with certain columns and rows that needs mapping to another structure.

Regardless of the kinds of transformation, you will probably see more munging activities than pure transformations, but that's okay.

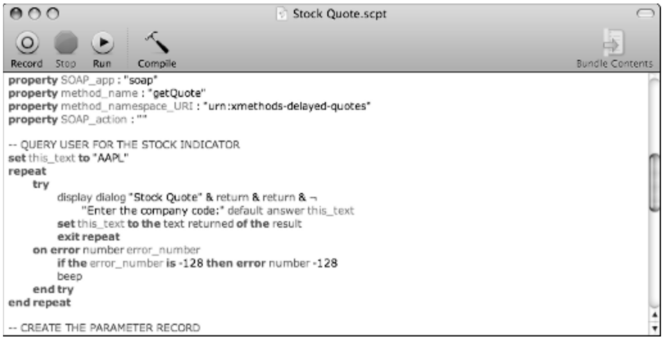
Data integration

Whenever you have data coming from more than one source (like databases or applications) and then present that data in a unified view, you have a data integration opportunity. A typical use is to grab data from different data sources (such as XML files or database tables) and then create a data view on a Web or application dashboard.

You'll have plenty of opportunities to perform data integration tasks, as you pull from different data sources and combine them into custom views. Figure 1.2 shows a snippet of AppleScript that uses SOAP to get a stock quote. SOAP stands for Simple Object Access Protocol, and it's used to provide access to services or data without opening up the entire application stack to outsiders. For example, to use the AppleScript stock quote script, you don't need to know anything about how the stock quotes are stored on the remote server. All you need to know is how to get to the stock quote and how to retrieve the ones you're interested in knowing more about.

FIGURE 1.2

Sometimes automation calls for the use of SOAP or other Web services to get data.



```
property SOAP_app : "soap"
property method_name : "getQuote"
property method_namespace_URI : "urn:xmethods-delayed-quotes"
property SOAP_action : ""

-- QUERY USER FOR THE STOCK INDICATOR
set this_text to "AAPL"
repeat
    try
        display dialog "Stock Quote" & return & return & ¬
            "Enter the company code:" default answer this_text
        set this_text to the text returned of the result
        exit repeat
    on error number error_number
        if the error_number is -128 then error number -128
        beep
    end try
end repeat

-- CREATE THE PARAMETER RECORD
```

Launching applications and scripts

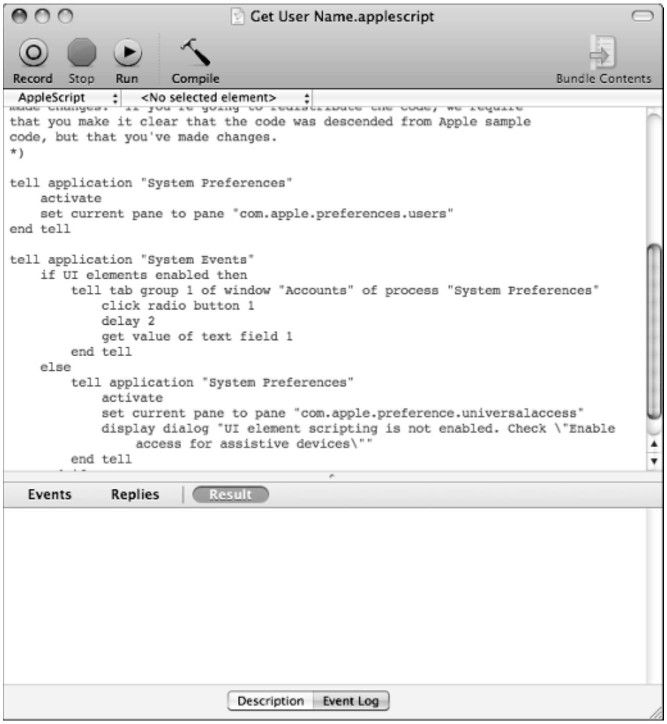
One of the easiest and most common automation tasks involves launching applications and scripts. For example, you might want to open and close different applications at different times of the day; starting every day at 8am, you might want to launch Mail and iCal, then turn off all applications and put your Mac to sleep at 6pm.

The same thing goes for custom AppleScripts. You may want to launch a script manually, using a workflow action in Automator, as part of an iCal event, or as the result of another AppleScript that is itself scheduled to run at a certain time. You'll have ample opportunity to explore all these possibilities.

Figure 1.3 shows a snippet of AppleScript that opens the Accounts pane in System Preferences.

FIGURE 1.3

Sometimes automation simply opens a window or application.



```
made changes.  If you're going to redistribute the code, we require
that you make it clear that the code was descended from Apple sample
code, but that you've made changes.
*)

tell application "System Preferences"
    activate
    set current pane to pane "com.apple.preferences.users"
end tell

tell application "System Events"
    if UI elements enabled then
        tell tab group 1 of window "Accounts" of process "System Preferences"
            click radio button 1
            delay 2
            get value of text field 1
        end tell
    else
        tell application "System Preferences"
            activate
            set current pane to pane "com.apple.preference.universalaccess"
            display dialog "UI element scripting is not enabled. Check \"Enable
                access for assistive devices\""
        end tell
```

Communications

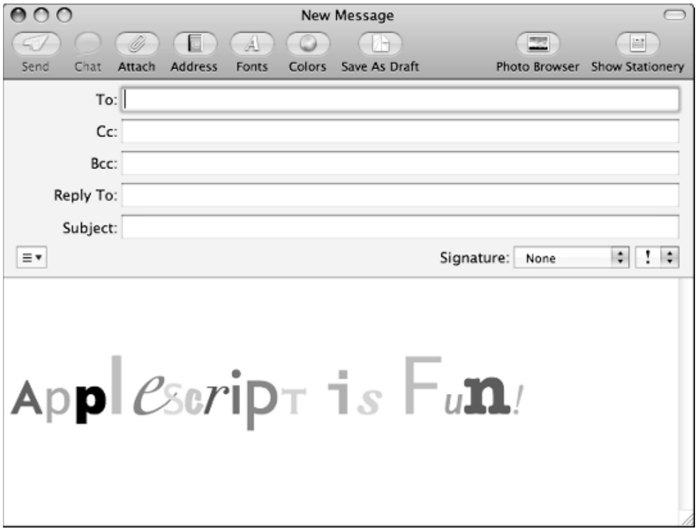
A useful candidate for automation is communications — whether by sending scheduled e-mails to a client database (such as a weekly newsletter every Monday morning) or responding to or sending e-mails based on certain events or criteria — for example, setting up an autoresponder for when users send messages to a certain e-mail address.

In this case, think of Automator and AppleScript as extremely powerful additions to Mail, Address Book, and iCal — together, they can be tied together into extremely powerful and customized solutions. For example, you could track customers by using groups and metadata in Address Book, use an AppleScript to pull out customers by group, schedule the AppleScript to run through iCal events (every two weeks, for example, you could e-mail an invitation to a free workshop), and then use an Automator workflow to process the data from users who do attend the workshops.

Figure 1.4 shows a rather silly example, admittedly, of what happens when you use AppleScript to format the fonts in a mail message. You probably won't have much use for this kind of outcome, but you will most likely need to control the appearance of your messaging at some point, and so knowing about the capability is important.

FIGURE 1.4

Sometimes automation is just about having some fun.



AppleScript is Fun!

Setting up reminders

One of the easiest things to automate is reminders, but don't fool yourself into thinking that only events in iCal have reminders. You can also use AppleScript to create reminders to revisit files with certain labels or Spotlight Comments (such as pending, draft, or in review). You can also create reminders that go far beyond the basic rules and flags associated with Mail, or tie together different applications to set up more custom reminders.

The idea is that you spend a lot of time working on files and data, but often not in ways that allow you to complete the work. It's also true that users tend to work with some files and data in an intermittent fashion, visiting the same folders and applications once a week or every quarter. The only way to stay sane is to set up reminders, and sometimes you need to go far beyond the simple reminders available in iCal.

Systems maintenance and backups

Mac OS X does an excellent job of providing system updates and backups (through Time Machine) on a regular basis, but there will come a time when you will need to create automated backups and maintenance programs of your own, even if all you're doing is mounting a network server once a day to check for updates or to back up files off site.

Why Automate?

Now that you have some idea of what can be automated, the question becomes, well, why should you automate? In some cases, automation would seem to require more time than would be required to set up an AppleScript or Automator workflow.

That supposition is only true if you consider that your work is a one-off deal, never to be repeated again. Given that view, yes, if you spend half an hour to automate what would normally take you 15 minutes to do manually, then you're wasting time. However, if you're going to perform the same workflow every day for the next week, or five times a month for the next year, then you'd be remiss in not at least considering automation.

Larry Wall, the inventor of Perl, said that one of the virtues of a great programmer is laziness. If you had to do the same thing more than once, a good Perl programmer would probably write a script and then be done with it. It might take them an afternoon to write the original processing script, but after that, the manual task that took an hour takes only two or three seconds.

Regardless, there are only a few good reasons for automation, but you're bound to hit one or another when you stop and consider the task before you. The reasons, broadly speaking, are as follows (and in no particular order):

- You want to save time
- You want to save effort
- You want to simplify a process
- You want to reduce errors
- You want to save on manpower
- You want to save money

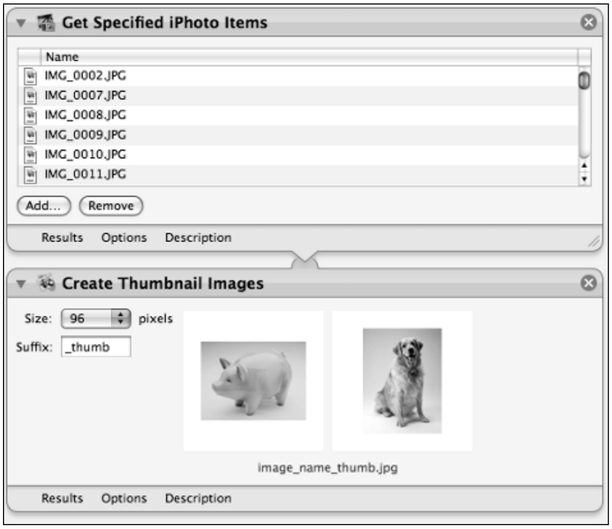
Save time

Saving time is a primary reason for automating a task. Instead of slogging through hundreds of images, documents, or e-mails by hand, you could build a workflow or script to zip through the pile in mere seconds.

Figure 1.5 shows a simple Automator workflow that creates thumbnails of images — a process that takes a lot of time when done manually.

FIGURE 1.5

A simple Automator workflow that creates thumbnails can save a lot of time.



Get Specified iPhoto Items

Name
IMG_0002.JPG
IMG_0007.JPG
IMG_0008.JPG
IMG_0009.JPG
IMG_0010.JPG
IMG_0011.JPG

Create Thumbnail Images

Size: 96 pixels

Suffix: _thumb

image_name_thumb.jpg

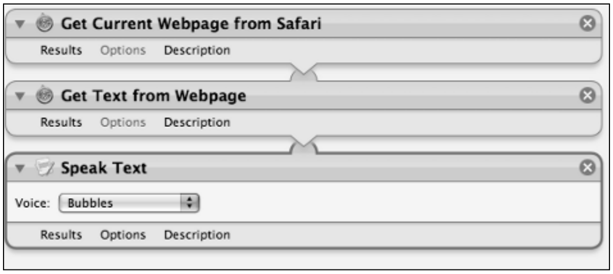
Save effort

Saving effort is not the same as saving time, not at all. You might have a process that takes just a little bit of time, but it might involve a lot of different people or different applications, or different types of interactions at different times of the month. Any of these combinations will quickly ramp up your level of effort.

Figure 1.6 shows an Automator workflow that extracts text from a Web page and then speaks that text out loud — normally, extracting raw text from a Web page is a laborious process, but Automator simplifies it easily.

FIGURE 1.6

Another Automator workflow might extract text from a Web page and speak it out loud, saving a lot of effort.



Voice: Bubbles

Simplify a process

Simplification is another easy win. If you have ten different manual steps, automating even half those workflow steps will create a good return on your investment. Even if your efforts result in three or four different workflows and scripts working together, you end up with a much simpler process.

Reduce errors

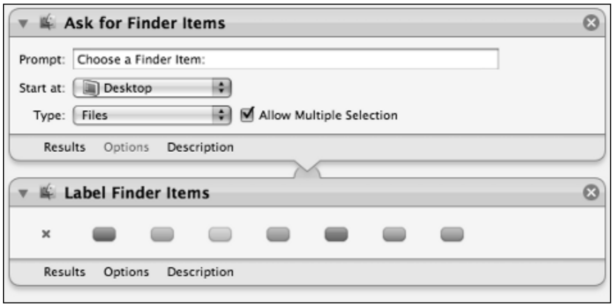
Given enough steps in a manual workflow, or enough time spent doing manual tasks, eventually it will lead to one or more errors. It happens to everyone. Errors happen because workers get tired, or get distracted by a phone call or a visitor, or because they lose interest. Maybe they didn't understand the initial instructions ("Oh, I thought you wanted all these images cropped to 400 x 400!").

In any case, setting up a repeatable workflow or script turns the task over to the computer, and the computer never gets tired, never gets distracted, and never loses interest. Whatever instructions you give it, rest assured that those instructions will be followed to the letter.

Figure 1.7 shows an Automator workflow for properly labeling files based on a user's selection in the Finder.

FIGURE 1.7

Another Automator workflow might, in two easy steps, label files for you.



Ask for Finder Items

Prompt: Choose a Finder Item:

Start at: Desktop

Type: Files ☑ Allow Multiple Selection

Label Finder Items

Save on manpower

If you've got five people working on a manual task (processing thousands of files by hand) and you automate that task, maybe all you need is one person to analyze the results of the automated processing, and the other four can be retasked to more important jobs.

Save money

At the end of the day, just about any automation will save you money, because it will generally simplify processes, reduce the amount of time it takes for task completion, and reduce the number of errors that cause so much pain. The point is, how much money will you save? For example, if you reduce a task from 50 person-hours a month to 1 person-hour a month, you've just saved at least $2,400, if you count each person-hour at $50/hour.

Other tasks, like reducing the number of steps in a marketing workflow involving communications with customers, might be harder to tally up, but after a while you'll learn how much time is spent by marketing staff after automation compared to before automation, and you'll be able to work out the savings.

Summary

In this chapter, you learned a bit about automation — what it is, what kinds of things you can automate, and why you should do it. Your Mac sits at the tail end of a long history of automation — you now have more power at your fingertips than entire corporations had 40 or 50 years ago.