

# Protect your Privacy with AppleScript

by Johnathon Williams



## WHAT YOU NEED

- Mac OS 10.2.3 or later (\$129, [www.apple.com](http://www.apple.com))
- Apple GUI Scripting (free, [www.apple.com/applescript/GUI](http://www.apple.com/applescript/GUI))
- Apple Mail (part of Mac OS X) or Microsoft Internet Explorer 5.2 or later (free, [www.microsoft.com/mac](http://www.microsoft.com/mac))

Welcome to the **MacAddict School of Script-Fu**. The bad news: **They are watching you.** The good news: **One script can shred your virtual paper trail.**



PHOTOGRAPH COURTESY OF APPLE

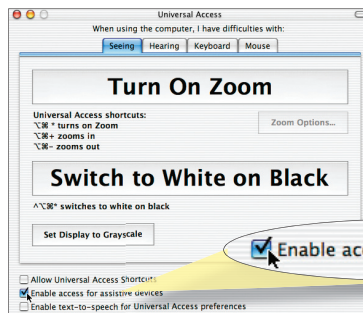
In 1984 (the book, not the year—or the Van Halen album), author George Orwell imagined a future where technology eradicated privacy—you know, all that “Big Brother is watching” stuff? Orwell never lived to see a personal computer, and maybe that’s a good thing. The shock might have killed him prematurely.

Why? Your Mac records just about everything you read, write, listen to, and watch on it. It does this for your convenience, to spare you from having to repeatedly hunt down your favorite documents and Web sites. Unfortunately, what is convenient for you is also convenient for snoops. Luckily, we Mac addicts have a safety net.

It’s called AppleScript.

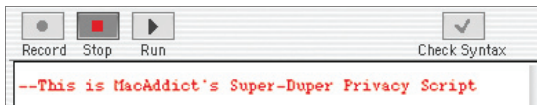
For the uninitiated, AppleScript is an English-like scripting language built into the Mac OS. With a little know-how, you can use it to create a digital watchdog for your own desktop. If privacy is something you covet, here’s how to create a script that will obliterate your deleted email in Mail, clear Safari and/or Internet Explorer’s browser cache and history, make the contents of your Recent Items menu disappear, and then empty your Trash. It won’t prevent a totalitarian government from one day ruling the world, but hey—you gotta start somewhere.

**1 Go Get GUI** Late last year, Apple released a beta of its GUI Scripting (aka UI Scripting) software, an AppleScript addition that allows scripts to click menus and buttons just like a human. Download the beta release from Apple’s site, but don’t install it just yet. To make the scripting addition work with your system, open System Preferences and click the Universal Access preference. In the resulting panel, check the Enable Access For Assistive Devices box, and close the window. Install the UI Scripting software by double-clicking the System Events package in the UI Scripting folder, and you’re ready to rumble.



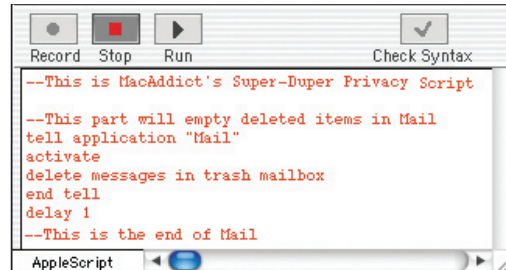
To allow AppleScript to click buttons and menu items, check the **Enable Access For Assistive Devices** box in the Universal Access preferences.

**2 Start Your Script** OK, let's start scripting! Launch Script Editor (inside Applications > AppleScript) to call up a blank script window. The top Description box allows you to type what your script will do. Filling this in is optional. The box below it is where you actually write the script. Click in the bottom box and type **--This is MacAddict's Super-Duper Privacy Script**. The double-dash (--) preceding the text tells AppleScript that the text string that follows is not a command but a comment, and to ignore it. Use it to make notes about segments of your script for yourself and others who may study your code. Then press Return twice to skip down a couple of lines.



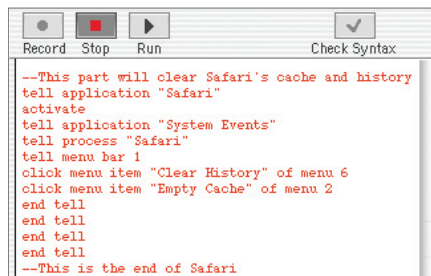
To post reference notes about parts of your script, add a double-dash before a text string.

**3 Create a Mail Delete Script** First up, create a script segment that tells Mail to delete all email in its Trash mailbox. (If you're not a Mail user, skip to step 4.) With AppleScript, you can write script text almost as if you were speaking it. Copy the script shown in the screenshot below exactly. The delay 1 command following the end tell command tells AppleScript to pause for a second before it tackles the next task. After typing a closing comment, press Return twice.



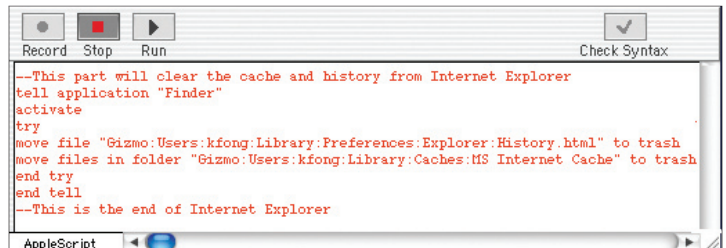
AppleScript language looks just like plain-old English—as spoken by Captain Kirk.

**4 Make Browser History** Most Web browsers keep records of every Web site you visit. This can be helpful, as it speeds up Web surfing. It can also be incredibly revealing. Therefore, create a script that deletes your browser history and cache. We show code for Safari and Internet Explorer—if using the latter, skip to step 5. Safari users: To create the actual script, copy the text exactly as it appears in the screenshot below. At the end, press Return twice, and then skip to step 6 (unless you also use Internet Explorer).



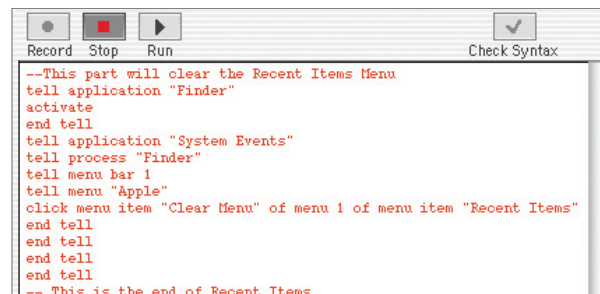
Copy this text exactly as shown to create a script that deletes Safari's browser history and cache.

**5 Script for Internet Explorer** Because GUI Scripting doesn't play as nice with Internet Explorer as it does with Safari, we will have AppleScript move certain IE files to the Trash instead of clearing history. Check out our screenshot below and type the code as displayed with the exception of two lines. In the line that begins with *move file*, replace *Gizmo* with the name of your own hard drive and replace *kfong* with the name of your own user directory. Replace the same variables in the text string on the next line. Finish by typing the remaining text as shown and press Return twice at the end.



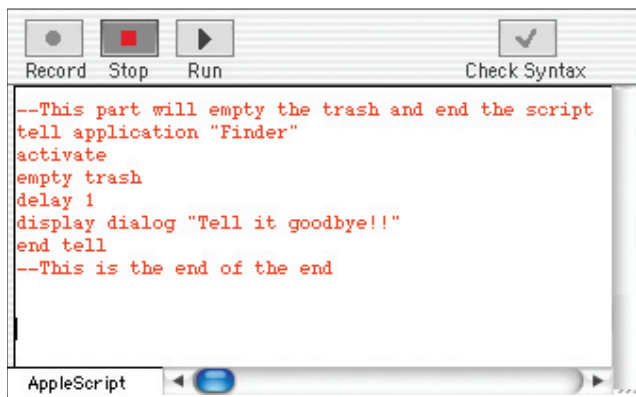
Because AppleScript is pickier than a new mother-in-law, be sure to type all directory names exactly as they appear when typing script (the code is case-sensitive).

**6 Make Recent Items History** Now create a script that clears your Recent Items from the Finder. Although having stuff in Recent Items can save you from hunting for your files and apps, you might not want snoopy coworkers or other eyeballs browsing through its contents to see what you've been up to. Again, copy our text verbatim, as pictured in the screenshot at right. Remember that AppleScript is case-sensitive. When you refer to an application or directory, make sure you cap or uncap appropriately (type names exactly as they appear in the Finder) except for comment text—anything goes there. When you're finished, press Return twice.



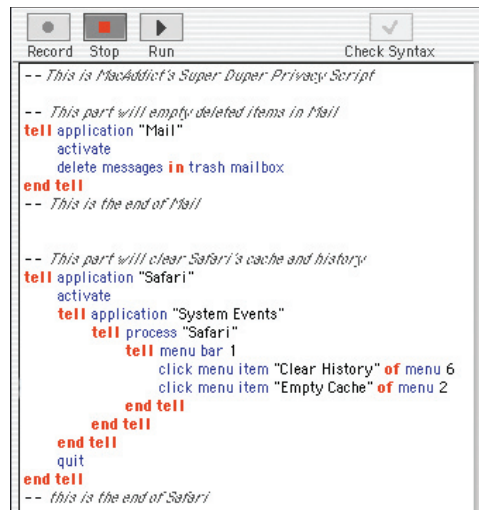
This script banishes all recently opened files—like your secret collection of old love letters—from the Recent Items menu.

**7 Take Out the Trash** Only one thing remains: creating a script segment to empty the Trash. Here's how to create one that not only does just that, but also notifies you when it's done. Once again, type the code exactly as shown in the screenshot below, but feel free to replace the quoted text on the *display dialog* line with whatever you want—this is what AppleScript will display in the dialog that appears when the script finishes running.



We saved the trashy stuff for last—here's the code.

**8 Cross Your Fingers and Run** Before you have AppleScript check your script, read through it carefully for any typos. AppleScript, like any scripting language, is very picky about syntax. When finished, cross your fingers—the moment of truth has arrived. Click Check Syntax to have AppleScript verify your work. If the script's good, the editor formats the text (and will launch some of the apps you scripted). If not, it gives you an error message—recheck your typing and try again. Once the editor formats your script, click Run to test it. If everything works, save the script—if not, recheck your text for directory misspellings or path errors, and try again. Now whenever you want to cloak your Mac-dabbling in secrecy, double-click your AppleScript file. (One note: If the script runs but doesn't clear off something you told it to, try running it again. GUI Scripting is still just a beta, and it can be cranky.)



If you typed everything correctly, the Script Editor formats your script nicely and colorfully.

## SCRIPTING STRUCTURE

by Johnathon Williams and Kris Fong

If you're a scripting neophyte, prepare to get hooked. AppleScript is so easy to learn that it's made scripting junkies out of technophobes. Because its language core is built around plain-old English, you can get a general understanding of how things work by merely studying scripts. If you're gung ho to roll your own, here are some common AppleScript statements and commands, and how to use them.

### tell

Almost all scripts begin with a tell statement that specifies a variable and item. This puts the chosen item on alert for whatever command follows, as in this script, which launches iTunes. (Always close a tell statement by typing **end tell** at some point in the script.)

```
tell application "iTunes"
activate
end tell
```

### open

This application-specific command instructs an app to open a file or files. For example, this script imports all the songs in our MP3 Dump folder into iTunes.

```
tell application "iTunes"
open file "Gizmo:Music:MP3 Dump"
end tell
```

### quit

Use this to quit any app you opened. For example, if we wanted iTunes to quit after it imported our songs in the script above, we'd write this. (We typed **delay 30** to pause the script for 30 seconds, giving

iTunes time to import our songs before the script quits iTunes.)

```
tell application "iTunes"
open file "Gizmo:Music:MP3 Dump"
delay 30
quit "iTunes"
end tell
```

### move

Use this to transfer an item from one place to another. Start the syntax with the command followed by the path to the source object (in quotes), the word *to*, and a reference or path to the new directory. For example, this script trashes an unwanted file.

```
tell application "Finder"
move "Macintosh HD:Users:johnathon:Documents:microsoft-sale" to trash
end tell
```

### set

This command tells AppleScript to set a variable as another value. When writing this, place the variable first and the value that it represents last. For example, this script sets a text title to a link.

```
set Best_Mac_For_UM to "http://www.macaddict.com/forums/"
```



Johnathon Williams is an avid script writer who collects conspiracy theories in his spare time.