AppleScript Basics

by Jerry Stratton

Copyright © 2011

http://www.hoboes.com/NetLife/AppleScript/

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 published by the Free Software Foundation; with no Invariant Section, with no Front-Cover Text, and with no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”

Getting started 1
Why AppleScript 1
Script Editor 1

Recording Actions 2
Carefully arrange the desktop 2

Simple Scripts 4
Comments 4
Dictionaries 4
Create a news screen 4
Automatically Reload Web Pages 5
Shell Tell 6
Repetitive Tasks 6
Talking Clock 6
Drag and Drop 8
Other Things You Might Do 9

AppleScript and the Command Line 10
Running AppleScript from the Command Line 10
Calling the Command Line from AppleScript 10

More Information 11
AppleScript: The Definitive Guide 11
AppleScript on Negative Space 11
Introduction to AppleScript Overview 11

Open Source License 12
Gnu Free Documentation License 12

GETTING STARTED

Why AppleScript

AppleScript can automate monotonous, repetitive tasks, or tasks that need to be performed at a specific time. Mac OS X can (using it's built-in scheduler via programs such as Apple's iCal or Embraceware's Awaken) run any script on the minute you need it to run.

AppleScript makes it easy to "talk" to standard Macintosh applications. It can "glue" applications together so that each application does what it does best.

AppleScript is easy to understand; it's almost English:

```
tell application "Safari" to make new document with properties {URL:"http://
            www.hoboes.com/"}
```

This is a valid AppleScript statement. If you use it, Safari will create a new document, and inside that document it will open the hoboes.com web site.

Script Editor

Every Macintosh comes with a program called "AppleScript Editor" that allows you to create AppleScripts. AppleScript Editor is in the Applications folder, inside of the Utilities folder.

RECORDING ACTIONS

If an application supports it, recording actions allows you to quickly build up a basic AppleScript. For example, if you go into AppleScript Editor and hit the "Record" button before making a new folder, you might end up with something like:

```
tell application "Finder"
  activate
  make new folder at folder "Desktop" of folder "jerry" of folder "Users" of startup
           disk with properties {name:"untitled folder"}
  set name of folder "untitled folder" of folder "Desktop" of folder "jerry" of
           folder "Users" of startup disk to "Quick New Folder"
end tell
```

This tells the program ("application") called "Finder" to first activate, then make a new folder, and then change the name of that folder to "Quick New Folder".

Unfortunately, most applications don't support recording; if they do, however, it is often the easiest way to make a new script.

Carefully arrange the desktop

Let's say that you want to always start up your computer with three folders open: your Desktop, your Documents, and your Downloads.

Go into Script Editor and get rid of whatever is currently there. Click "Record" and then switch to the Finder.

Hold down the Option key, pull down the Finder menu, and choose "Close All".

Use Command-N to open a new window, and from that window open your Desktop folder.

Move your Desktop window to the upper left, and then arrange it to take up a bit less than a quarter of the screen.

Do the same for your Documents folder, but put it in the upper right.

Then do it again for your Downloads folder, but put it in the lower left.

Go back to AppleScript Editor and click "Stop". You should have something like this:

```
tell application "Finder"
  activate
  close every window
  close Finder window 1
  make new Finder window
  set target of Finder window 1 to folder "Desktop" of folder "jerry" of folder
           "Users" of startup disk
  set position of Finder window 1 to {99, 44}
  set bounds of Finder window 1 to {99, 44, 995, 699}
  make new Finder window
  set position of Finder window 1 to {1008, 44}
  set bounds of Finder window 1 to {1008, 44, 1934, 700}
  set target of Finder window 1 to folder "Documents" of folder "jerry" of folder
           "Users" of startup disk
  make new Finder window
  set target of Finder window 1 to folder "Downloads" of folder "jerry" of folder
           "Users" of startup disk
  set position of Finder window 1 to {102, 730}
end tell
```

Next, test it. There's one odd thing in it: after it closes every window, it tries to close each window individually. But it can't do that, because it just closed all of them.

Delete the lines that look like "Close Finder window 1". Your script will look like:

```
tell application "Finder"
  activate
  close every window
  make new Finder window
  set target of Finder window 1 to folder "Desktop" of folder "jerry" of folder
           "Users" of startup disk
  set position of Finder window 1 to {99, 44}
  set bounds of Finder window 1 to {99, 44, 995, 699}
  make new Finder window
  set position of Finder window 1 to {1008, 44}
  set bounds of Finder window 1 to {1008, 44, 1934, 700}
  set target of Finder window 1 to folder "Documents" of folder "jerry" of folder
           "Users" of startup disk
  make new Finder window
  set target of Finder window 1 to folder "Downloads" of folder "jerry" of folder
           "Users" of startup disk
  set position of Finder window 1 to {102, 730}
end tell
```

You can save this as "Arrange Folders" with File Format "Application", and every time you run it it will close all of your Finder windows and open up just the three you want, arranged the way you want them.

Click here to download full PDF material