

## Backing Up With AppleScript

Extra large blockbuster feature for all you script lovers

Steve Harris

This article is one of many published in 1984-Online over the last year on the subject of AppleScript. I am not going to go over the basics of AppleScript, as this was done in issues 2 to 6 of 1984-Online, which are readily downloadable from 1984-Online's web site. Don't be put off, though, I always make the assumption that nobody bothers reading them anyway and write like you're all Class A novices. And rightly so.

Many articles about backing up your data begin with some patronising lecture about how you must be several brain cells short of a glass of water if you don't back up your data at least twenty times a day, perhaps accompanied by a predictable tale about some poor bloke who lost all his data and went bankrupt, then his wife left him, his children disowned him and his dog moved in with his mistress and so on, ad noisome.

Fortunately, we at 1984-Online Magazine are not here to patronise you or tell you stories, we're just here to help your Mac using life be as pleasurable jargon-free and acronym-free as is humanly possible. As a Mac user, you are spared endless hours tinkering with the anarchic and obscure bits of computers some so-called modern operating systems call 'power features'. The Mac OS and AppleScript prove that power and ease-of-use needn't be mutually incompatible. However, whether your computer is geek or chic backing up is a very good idea.

I know, I know, I know, it's so dull remembering to copy files around every so often, rooting around in the depths of your Mac's filing system trying to get it all together, but AppleScript is here to help. All you need is some form of removable storage — a Zip drive would be ideal and a real bargain at under £100 — and any version of the Mac OS / System with AppleScript installed.

So without wanting to sound patronising backing up your data, in English, means making a copy of it for safe keeping. Before I begin, though, I have a small tale to tell of

some poor bloke who didn't make backups, lost all his data and went bankrupt. Then his wife... (CUT!)

### Laziness For An Enlightened Generation

We are going to create an AppleScript script which backs up your data, and we're going to do it in several stages. The first stage will be to create a usable backup script, and the rest of the stages will look at polishing the script to make your life even easier. At every stage your script should be usable (if only just) so it's up to you how much polishing you do.

If this all sounds a bit daunting then don't worry, it won't be, I promise. Provided you know how to copy files and type a few words, it's a cinch.

### Getting Started

First work out what files / folders you want to back up. This is up to you. For my own purposes I'm going to back up my Documents folder (53MB) and my email folders (about 8MB). As this script is simple, you'll need to find out how much you're backing up and ensure that the space they'll occupy will not exceed the maximum space available on your backup disk. In my case I'll be backing-up about 61MB of data, which will fit nicely onto a 100MB Zip disk.

A few tips: if you're thinking of backing up your email files and space is tight, remember to use the compacting features included in both Netscape Communicator and Outlook Express before you backup as this can often (especially in Netscape's case) save megabytes of wasted rubbish. I can't speak for other 'proper' email programs as I never use them.

Netscape's Mail (and indeed all your Netscape settings) are stored within a folder with your user name in the Netscape Users folder in your Preferences folder in your System Folder (deep breath), whereas Outlook Express's mail is stored within a folder (probably called "Main User") within a folder called OE User(s) in your Outlook Express folder.

Once you've established what you want to backup and whether you've got enough space to do it in one script, we're ready to go.

Pop your disk in (I'm using a Zip) and give it an appropriate name. I've called mine "Backup Disk".

### Record Breaker

To create the main bulk of the script we're going to record the manual process of backing up. To do this:

Open the Script Editor (you should find it in the AppleScript folder of your Apple Extras folder), and click Record.

ow drag the relevant folders from your hard disk to your backup disk, if you've got Mac OS 8.5 or above, you don't have to wait until one set of files has copied before copying the rest.

Go back to the Script Editor and click Stop (you don't have to wait until the copying has finished). And voila! You have a usable backup script!

Well, sort of. The script makes a few assumptions. Firstly, it assumes that the backup disk is empty, as an AppleScript, recorded as ours was, won't replace files which already exist. Secondly, it assumes you want to see all the windows you opened and closed as you recorded the script opening and closing while you backup. It might look pretty, but it could become annoying. Thirdly, it assumes the files will copy before AppleScript times-out the operation and sulks. This last assumption needs some explanation.

The way AppleScript works is to give instructions to programs and then wait until they've done it. Unfortunately, AppleScript is impatient and wants everything done now, and will wait no longer than one minute before barking some curt remark and giving you the cold shoulder treatment. Sounds like someone I know rather too well, but whose name I daren't mention for fear of... OUCH! What was that for?

However, unlike 'she who must be obeyed', you can override this behaviour in AppleScript. Ah, but if only real life were so simple.

So what we need to do is cut down the script, whilst keeping its functionality intact and add a small command to stop it being impatient along with a minor tweak to replace existing files whilst copying.

Take The Rubbish Out, Dear

If you value the script you recorded, it would be sensible to save your script now, and make a copy of it to work on the next bit.

Amongst all the opening and closing of windows are the few important commands you really need to perform your backup. Let's cut out all the rubbish and see what we're left

with.

tell application "Finder"

activate

select folder "Documents" of startup disk

copy selection to disk "Backup Disk"

select folder "Steve Harris" of folder "Netscape Users" of folder "Preferences" of folder "System Folder" of startup disk

copy selection to disk "Backup Disk"

select folder "OE User(s)" of folder "Outlook Express 4.5 Folder" of folder "Microsoft Internet 4.5" of folder "Internet Applications" of folder "Internet" of startup disk

copy selection to disk "Backup Disk"

end tell

All I've done is taken out all the lines apart from the 'select' line before the 'copy' lines, and the 'copy' lines themselves. The lines I've removed are highlighted in red, below:

tell application "Finder"

activate

select startup disk

open selection

select folder "Documents" of startup disk

copy selection to disk "Backup Disk"

select folder "System Folder" of startup disk

open selection

close container window of startup disk

select folder "Preferences" of folder "System Folder" of startup disk

open selection

close container window of folder "System Folder" of startup disk

select folder "Netscape Users" of folder "Preferences" of folder "System Folder" of startup disk

open selection

close container window of folder "Preferences" of folder "System Folder" of startup disk

select folder "Steve Harris" of folder "Netscape Users" of folder "Preferences" of folder "System Folder" of startup disk

copy selection to disk "Backup Disk"

close container window of disk "Backup Disk"

close container window of folder "Netscape Users" of folder "Preferences" of folder "System Folder" of startup disk

select startup disk

open selection

select folder "Internet" of startup disk

open selection

close container window of startup disk

select folder "Internet Applications" of folder "Internet" of startup disk

open selection

close container window of folder "Internet" of startup disk

select folder "Microsoft Internet 4.5" of folder "Internet Applications" of folder "Internet" of startup disk

open selection

close container window of folder "Internet Applications" of folder "Internet" of startup disk

select folder "Outlook Express 4.5 Folder" of folder "Microsoft Internet 4.5" of folder "Internet Applications" of folder "Internet" of startup disk

open selection

close container window of folder "Microsoft Internet 4.5" of folder "Internet Applications" of folder "Internet" of startup disk

select folder "OE User(s)" of folder "Outlook Express 4.5 Folder" of folder "Microsoft Internet 4.5" of folder "Internet Applications" of folder "Internet" of startup disk

copy selection to disk "Backup Disk"

close container window of folder "Outlook Express 4.5 Folder" of folder "Microsoft Internet 4.5" of folder "Internet Applications" of folder "Internet" of startup disk  
end tell

We can go even further and replace the word 'selection' on the 'copy' lines with all the words on the 'select' lines apart from the word 'select' itself. Does that make any sense? Maybe seeing the script after that's been done will clarify it:

tell application "Finder"

activate

copy folder "Documents" of startup disk to disk "Backup Disk"

copy folder "Steve Harris" of folder "Netscape Users" of ↵

folder "Preferences" of folder "System Folder" of ↵

startup disk to disk "Backup Disk"

copy folder "OE User(s)" of folder ↵

"Outlook Express 4.5 Folder" of folder "Microsoft Internet 4.5" of folder ↵

"Internet Applications" of folder "Internet" of startup disk ↵

to disk "Backup Disk"

end tell

Note: You can split long lines to make the easier to read by pressing Command-Return.

This inserts a `↵` character to show the line's been split.

### Adding Patience

Next let's add the two lines which will stop AppleScript being such an impatient old nag. By default AppleScript times-out after 60 seconds. Let's set the timeout time to ten minutes (if you still have time-out problems, you can up the timeout period as appropriate):

After the line which reads 'activate' add this:

```
with timeout of 600 seconds
```

And before the line which reads 'end tell', add this:

```
end timeout
```

Now click the 'Check Syntax' button to make it look all pretty again.

### Replacing The Files As We Copy

In order to replace the files that are there, we need to fiddle with the copy lines in the script:

Firstly change 'copy' to 'duplicate'.

Now add 'with replacing' onto the end of the 'duplicate' lines, so they look something like this:

```
duplicate folder "Steve Harris" of folder "Netscape Users" of ↵
```

```
folder "Preferences" of folder "System Folder" of ↵
```

```
startup disk to disk "Backup Disk" with replacing
```

### The Final Draft

We've made a lot of changes, here is my script in its entirety:

```
tell application "Finder"
```

```
activate
```

```
with timeout of 600 seconds
```

```
duplicate folder "Documents" of startup disk to disk "Backup Disk" with replacing
```

duplicate folder "Steve Harris" of folder "Netscape Users" of ↵

folder "Preferences" of folder "System Folder" of ↵

startup disk to disk "Backup Disk" with replacing

duplicate folder "OE User(s)" of folder ↵

"Outlook Express 4.5 Folder" of folder "Microsoft Internet 4.5" of folder ↵

"Internet Applications" of folder "Internet" of startup disk ↵

to disk "Backup Disk" with replacing

end timeout

end tell

### Saving The Script

Before you save your script, it would be nice to put a description in the description box at the top of the script window as this can appear in a box when you run the script.

Here's my description:



In order that your script can be run like any other program, you'll need to save it as an application. To do this:

Choose Save As... from the File menu.

Choose where you want to save your script in the usual way.

Choose Application from the Kind popup menu.

Click Save.

When you run the script, you should see a message looking something like this:

Buttons For Punishment

Next month, for the brave and / or foolish amongst you, we'll look at further polishing of

the script, which will include how to handle things when something goes wrong, which, in programming terms can be as important as backing up. Remember the sad tale of that poor bloke who didn't handle his programming errors properly, lost all his data, and went bankrupt. Then his wife... (CUT!)

 Steve Harris

<[steve\\_harris@1984-online.com](mailto:steve_harris@1984-online.com)>