



Fighting talk

Operating systems: aren't you sick of them? Everywhere you look, someone is crowing about the advantages of OS X while putting the boot into OS Y. Howard Oakley has had enough.

In common with many people, I find the popular online game of OS Wars extremely tedious. Like it or not, though, Apple, Microsoft, Sun, Be and anyone else wanting to enter the fray has the vital task of convincing the public that their particular operating system is distinct from the others, in ways which are obviously advantageous. I think this is what our American friends refer to as "leveraging", which conjures visions of Archimedes trying to move the globe with the Greek antecedent of a caber.

You can split hairs and argue ergonomics over the human interface, but one area in which Mac OS is highly distinct, and which users and Apple adverts should be leveraging like a contestant in a strongest man contest, is AppleScript. Unfortunately most of the original AppleScript development team have long since left Apple, and I begin to wonder if anyone in marketing even remembers its existence. Look through the old development code examples which were provided to help code geeks implement support for AppleScript in their applications, and you'll see names of luminaries like CK Haun, who not only left Apple years ago, but also left Be some months ago.

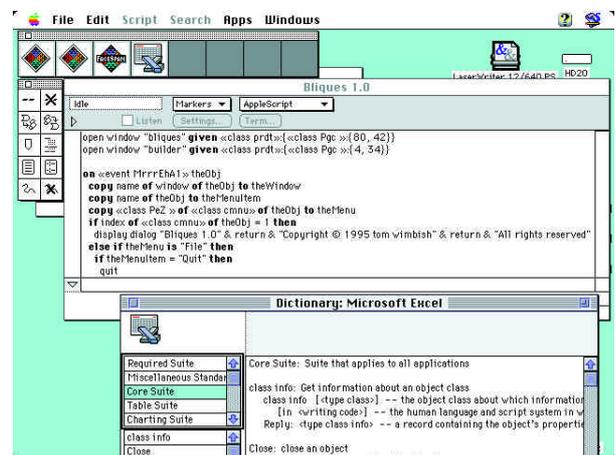
AppleScript is a sophisticated near-English programming language with which you, as an ordinary user, can control and co-ordinate shrink-wrapped applications. Introduced with System 7, it works with applications supporting AppleEvents, which are to AppleScript what keypresses are to the keyboard. In other words, when you run a script, it is converted into AppleEvent messages sent to the target application, which are handled by the target in among the mouse, keyboard, and other items

If you fancy writing your own AppleScripts, Scriptor offers all the tools you'll need, including a dictionary browser and a sophisticated debugger

which are able to generate events.

Part of the OS war game among many Mac fans is to take every opportunity to castigate Microsoft, in particular claiming that Office for Macintosh is all part of some conspiracy to convert Mac users to Windows 95. This theory is half-baked in many ways, not least of which is the fact that Microsoft applications include extensive support for AppleScript; in fact, armed with AppleScript and Visual Basic for Applications (VBA), Excel for the Mac is superior to its Windows-hosted sibling.

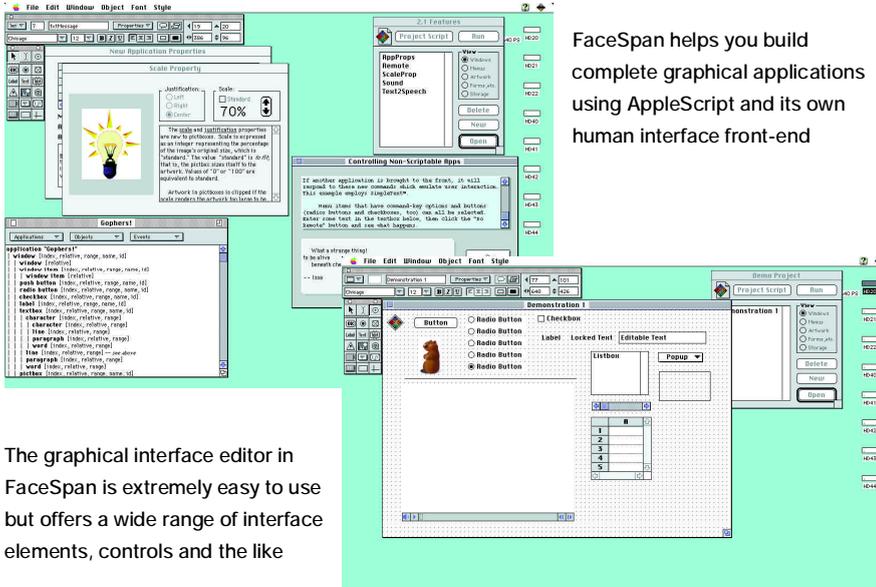
One of the reasons AppleScript has not set the world on fire is that, by some bizarre quirk of fate, Apple seems to have lost all its development team before it could complete the necessary suite of development tools. The rudimentary Script Editor which ships with each copy of System 7 is clear and simple, but falls far short of being a proper tool for script development. Another serious omission in older versions of Mac OS was that the Finder, arguably the most important application for scripting, was not scriptable (although this is not true of recent releases). Although Apple put a lot of early effort into defining the AppleScript language and suites of script commands and objects, and making them accessible via dictionaries, few applications became recordable. (That



is to say, you can step through a series of actions and have them automatically recorded as a script).

One of the first and most essential features of any script development tool is the facility to debug code which is not working as it should. This is not just for experts; indeed, good debugging facilities are more important for beginners, so that they can track down their errors rather than just getting frustrated and resorting to something more pleasurable, like poking themselves in the eye with a broken bottle.

Various third parties came up with improved development environments, including Paul G Smith, one of the co-founders of Full Moon Software Distribution, purveyors of fine scripting and other development tools to the UK and quite a bit of Europe. But the most durable, and now by far the best, is Scriptor 2.0. If you're daunted by the idea of programming your Mac in any way, you should take a look at Scriptor because it is not written to satisfy the needs of nerds. It is a highly accessible script development tool designed to work with FaceSpan, one of the most wonderful



The graphical interface editor in FaceSpan is extremely easy to use but offers a wide range of interface elements, controls and the like

pieces of software I have ever seen. Like all the best and most innovative software, you wonder why no-one had thought of FaceSpan before. It is an application generator which uses AppleScript as its programming language, giving you free rein to do what you will in your own application, and to control others.

To illustrate how recursively wonderful this is, you can even script the FaceSpan development environment using FaceSpan.

An integrated graphical interface editor, of the kind popularised but not invented by Microsoft's Visual languages, helps you to construct the user interface, behind which are gobbets of AppleScript to make it all work. You can run FaceSpan projects within the FaceSpan application, or extrude them as standalone applications. Attesting to the wisdom of this approach are the Apple Network Servers, which are administered remotely using AppleScript

FaceSpan helps you build complete graphical applications using AppleScript and its own human interface front-end

with a FaceSpan front-end. So armed with Scripiter and FaceSpan, and a selection of standard books including *Danny Goodman's AppleScript Handbook* (Second Edition, Random House) or *The Tao of AppleScript* (Second Edition, by Derrick Schneider, Hayden Books), plus *Apple's AppleScript Reference: English Dialect* (Addison-Wesley), the world is your oyster.

Perhaps the biggest shortcoming with AppleScript, however, has been what you *cannot* do with it. If an application has support for, say, copying all worksheet cells greater than ten, pasting them into the next sheet and sorting them into rank order, then scripting that sequence is straightforward. But many major applications fall short in various respects, most commonly in handling dialogs. It is not uncommon to find yourself blocked by a dialog, a brief piece of interaction which mars a script.

FaceSpan 2.1, the latest version, overcomes this problem in an obvious but ingenious way. It provides two new script commands: "click as user" to mimic mouse actions, and "type as users" to mimic keyboard input. Armed with these, you can get past most previous stumbling blocks.

Innovation lives!

A frequent riposte in the OS Wars game is that Apple no longer innovates. I recently stumbled across one of many active projects which attest to continuing innovation: Project X, a "fly-through" web navigator which links neatly to the web browser of your choice.

It is worth checking out hot and new items on a range of Apple's sites from time to time so that you keep abreast of these new ideas. Some, like the Communications Toolbox, will fly for a while and then die. Others, like OpenDoc, promise to change the face of computing.

Next month I'll look at how OpenDoc is taking off, and how it fits into Mac OS 7.6 and beyond.

A quick guide to Apple network protocols

Whether you're trying to decipher an error message or get a mixed network up and running, these are AppleTalk's catchily-named components:

LocalTalk A physical connection using a variety of different cabling systems to connect the printer serial port to a network. Not to be confused with AppleTalk itself.

EtherTalk A physical connection using any of the ethernet cabling schemes.

TokenTalk A physical connection using a Token Ring cabling scheme.

AppleTalk The suite of networking protocols which can be run across any of these physical connections.

LAP (Link Access Protocol) The lowest level software protocol which is run over the physical connection: it comes in different varieties for EtherTalk (ELAP), LocalTalk (LLAP), and TokenTalk (TLAP).

DDP (Datagram Delivery Protocol) The next lowest level software protocol, this time common to all AppleTalk variants, which provides and exchanges data between sockets.

NBP (Name Binding Protocol) Discovers named entities on the network and then supports their proper addressing.

ZIP (Zone Information Protocol) Supports the division of large networks into zones, and communication between them.

RTMP (Routing Table Maintenance Protocol) Acquires and maintains a list of routes to support the transmission of messages over the network.

ADSP (AppleTalk Data Stream Protocol) Provides a "pipe" for the reliable exchange of data between two applications.

AEP (AppleTalk Echo Protocol) Supports reliable communications over DDP.

ATP (AppleTalk Transaction Protocol) Provides reliable transfer of data between network entities: typically used to move files and other data between two systems on the network.

ASP (AppleTalk Session Protocol) Extends ATP by helping it transfer an ordered sequence of data.

AFP (AppleTalk Filing Protocol) Supports the cross-system and cross-platform sharing of filing system information, including file sharing and program linking.

PAP (Printer Access Protocol) Connects workstations and printers, and supports communications between them, including the sending of PostScript documents to the printer.

SNMP (Simple Network Management Protocol) A cross-platform system for managing local and remote network connections. Although AppleTalk can support this, so do most other networking systems.

PCW Contacts

Howard Oakley is pleased to hear from Mac users. Contact him at mac@pcw.vnu.co.uk
Apple Computers: phone 0181 569 1199; web home pages www.apple.com and www.euro.apple.com.

Scripiter 2.0, costs £169. FaceSpan 2.1, costs £207.50. Distributed in the UK by Full Moon Software Distribution 01628 660242, which can also supply the AppleScript books. Project X is available free of charge from mcf.research.apple.com/ProjectX (All prices are exclusive of VAT).