Macromedia Flash Professional 8
Keith Alcock, TCS Member
tcs@keithalcock.com

## Introduction

Macromedia Flash Professional 8 "is the industry's most advanced authoring environment for creating interactive websites, digital experiences and mobile content," according to its new owner, Adobe Systems Incorporated. Much as Adobe Acrobat creates PDF files viewable with Adobe Reader on nearly every personal computer in the world, Flash Professional produces SWF files playable on any computer running the ubiquitous Flash Player, usually inside a web browser. Chances are that you and the majority of your audience already have the player installed and it is the player's market penetration that bestows the authoring tool much of its perceived value.

Flash isn't the only game in town, of course. There are alternative ways to deliver interactive content over the web. Many programming languages, both popular and obscure, include web browser plug-ins that execute code written in those languages. Java, with its applets, was one of the more popular, but they also exist for Smalltalk, Tcl/Tk, and lisp to name a few. DHTML and now AJAX offer some functionality overlap with Flash without requiring plug-ins at all. Microsoft is said to be coming out with a product called Microsoft Interactive Designer, otherwise known as Sparkle, to compete with Flash. For the moment, however, Flash has the most widespread player, the best animation, and in its eighth version is hardly vaporware. Let's take a look.

## Getting Started

Changes in the review policy required me to download a 30-day demo version just like everyone else. The 110 MB download includes both Flash Professional and Flash Basic. You don't choose between the two until after installation. Not many customers read the license agreement, but they should. One clause states that you may be audited for compliance with license terms and charged for the audit if violations are discovered. Read this carefully if it concerns you. The software did attempt Internet access without asking my permission, but perhaps this behavior is limited to the demo version.

The product offers a quick tour, tutorials, and extensive documentation right off the bat, which guide the new user up a steep learning curve. The quick tour made things look easy, and the first tutorial was straightforward. My work on the second tutorial looked less and less like it was supposed to until I finally abandoned it. I did run across bad hyperlinks, references to non-existent menu items, and misnamed directories elsewhere in the documentation, but tutorial instructions are probably correct. From a programmer's perspective, long lists of click here, resize that, and move this there are problematic. They are difficult to remember, reproduce, compare, and verify. Graphic artists and designers, however, probably feel right at home. Luckily for me, most Flash presentations involve a fair amount of code, ActionScript, which is more my style.

**Show Time**

My "Hello, world!" application/movie is an investigation into some of the text manipulation abilities of Flash. It is a(nother) ransom note generator that accepts text from the user and then distorts the letters as if they were cut and pasted from different sources. I had hoped to use horizontal and vertical scaling to resize letters, but these operations are not supported for individual characters of a longer text. Flash does support HTML formatting of text, but is severely limited in the styles it can use. Neither letter-spacing, word-spacing, nor font-stretch is included in its vocabulary. In the end, I made do with font size and letter spacing in non-HTML text to produce the variation. It's not an acceptable compromise for my real project, but suffices for a review. These attributes can easily be controlled from code using a TextFormat object and applying it at different positions in a TextField. Flash keeps track of all the text runs (consecutive letters with the same format) on its own so that in the end the code is very simple despite all the experimentation required to write it.

For anyone familiar with languages based on C syntax such as JavaScript in general or ECMAScript-262 in particular, ActionScript is reassuringly easy to learn. ActionScript 2.0 adds type information to the earlier ActionScript 1.0

```
function Ransomize(aTextField:TextField):Void {
    var myString:String=aTextField.text;
```

which enables accurate and useful code hinting. Flash even attempts to infer data types from comments and variable name suffixes for hinting. It tries valiantly to read your mind. Occasionally I was reminded that we don't think alike, though. Unknown variables, functions with the incorrect number of arguments, and function calls undefined on a particular data type don't bother ActionScript. Some typos are apparent only from the misbehaving program, which is very late. In this case the debugger comes in handy and I was pleased with its functionality. The editor's syntax highlighter is sometimes deceptive and the compiler's error messages do not include column numbers, so nirvana it isn't. Flash interfaces with Microsoft Visual SourceSafe, according to the documentation, and ActionScript class files are text so that differencing is possible. The FLA files, where some minimal amount code must remain, are binary and thwart simple change tracking measures, unfortunately.

To add motion to my application, I had to abandon the text editor and go graphical. It seemed easy at first and for experienced designers it probably stays that way, but I ran into problems not just figuring out how to specify what my application should do at what time, but also convincing Flash to get it done. For example, to make two rectangles the same size, I entered dimensions into the convenient space provided. I typed 10 and pressed enter. The program responded with 9.8. I typed 10 again. The program compromised with 9.9. I typed 10 again and finally the program conceded. As an alternative I tried snapping the size to a grid and that failed. At times I noticed that two texts having the same font, font size, character spacing, and letters would differ in width. What caused it and what fixed it are both still mysteries, but it certainly took a long time fiddling before it worked. To move an object between two points on the stage, you use

a "tween," which Flash happily generates automatically. They worked wonderfully in the tutorials, but it turns out that the default tween changes my text to a graphic, hiding the letters from the ActionScript code. They're just not there anymore and ActionScript proceeds without complaint, waiting for the programmer to notice something wrong. It took a long time, but now that it's over, I'm satisfied with the result. Just be prepared for a long learning curve and a fair amount of frustration.

Flash automatically generates an HTML file containing tags required to display a movie. Simply transfer the HTML and SWF files to the web server. No special configuration is required at my bare bones service provider and I'm happy to avoid that hassle. The movie plays fine on both the Windows and Macintosh computers that I tested. However, there's always a "but." I noticed that one can copy and paste the ransomized text, but only unformatted text is available from the Windows clipboard. Foiled again!

**Extending the Environment**

ActionScript code runs in the Flash player, which usually displays in a web browser. I was excited to learn that Flash offers a way to extend the development environment using JavaScript and potentially C on the developer's machine. This means that pre-determined text can be ransomized prior to delivery, so I tried it out. Flash includes a JavaScript editor with syntax highlighting, although no syntax checking, that will produce a mostly text JSFL file. Debugging is not supported. The JavaScript interface to Flash objects is somewhat different from the ActionScript version, so the code above changed to

```
function Ransomize(aText) {
        var myString=aText.getTextString();
```

which is a fairly straightforward conversion. If the JSFL file is placed into the right directory, its functionality is readily available from Flash menus. The programmer in me really appreciates this. For complex or time-consuming operations, or simply code reuse, one can call out from JavaScript to C functions placed in DLLs. I compiled and tested the example provided and it does seem to work, so there is one more exciting possibility.

Flash also extends to the Macintosh. I moved my project to a Mac, opened it in the Flash Professional 8 demo on that machine, and verified that the movie ran. It did. However, I couldn't continue development since all but the first of my frames had disappeared. Since the movie ran properly, the frames were hiding somewhere, but definitely not on the timeline. The development environment is apparently not as portable as the player. Installed with Flash is another program, the Macromedia Extension Manager, which offers additional possibilities. Extensions can be downloaded from Flash Exchange. I could find no information, however, on how to create an extension myself.

**Conclusion**

Some Flash movies, presentations, and games out there are so impressive that I had very high hopes for the development environment. It appears that my particular application hit a weak spot in Flash's abilities and my review may have come at an inopportune time in the acquisition process between Macromedia and Adobe. Both features and quality one would expect from a product with such a high price tag seem lacking. My inclination would be to wait until the next release before purchase. Also, third party products that target the SWF file format are taking advantage of Flash's success and may fill in the functionality I miss. As usual, your mileage will vary. I wholeheartedly recommend the demo version and encourage you to first try and then buy.

**Company/Product information**

| | |
|---|---|
| Vendor: | Adobe Systems Incorporated |
| URL: | http://www.adobe.com/products/flash |
| Price: | $699 online from Adobe |

**System requirements**

| | |
|---|---|
| Processor: | 800 MHz Intel Pentium III (or equivalent) and later |
| Operating System: | Windows 2000 or XP |
| Memory: | 256 MB RAM |
| Graphics: | 1024 x 768 x 16-bit |
| Disk space: | 710 MB |

The Flash project (FLP), document (FLA), movie (SWF), and web page (HTML) files for the ransom note generator are available from my website at http://www.keithalcock.com. So too is the JavaScript command file (JSFL) for ransomizing text from within Flash.