

What's up with Flash?

Phillip Kerman

If nothing else, Flash's greatest asset is the ubiquity of its plug-in. There are other attributes which make it great too: small file size (fast download), anti-aliased graphics, scalable to any screen size, MP3 compressed audio.

But what can Flash do? (*a lot*)

How sophisticated are the programming capabilities? (*very*)

Finally, are there other uses for Flash besides creating web animations? (*yes*)

Flash 3 added two sophisticated features: "Tell Target" and "Load Movie". Tell target allows for (at runtime) commands (called "actions") to be directed to specific "movie clips" (sub-timelines—similar to Authorware's bitmap sequence "movie icon" or Director's "film loop"). Load Movie is a way to modularize your creation into separate Flash files which can be individually "loaded" (and thus downloaded) at runtime. These two features allowed for surprisingly sophisticated creations. To reach further into "programming" the only avenue was via JavaScript—where a "dumb" Flash movie could "call" a JavaScript function when the user clicked a button or the animation reached a particular frame. (Also, JavaScript could, at any time, "tell" Flash to *do* any one of its built-in features.)

Flash 4 now has programming capabilities provided in the form of custom user variables and a handful of built-in functions. It may seem with just a handful of functions programming would be easy—but it takes an even more creative programmer to work within the limits of Flash's *Action Script* language (or, if you will, *Flingo*). Every additional component to Flingo adds to the overall size of the Flash plug-in—which remains an attractively small size of less than 200K. Although I've seen (and created) some amazing stuff in Flash 4, it remains limited. Let me attempt to clarify the range of possibilities within Flash... although, any time you hear someone say something's "impossible" with software you shouldn't believe them.

The range of possibilities within Flash 4:

Respond to user events: mouse events (mouse down, mouse up, mouse over, mouse leave, etc.) and key press.

Respond to system events: when an animation frame is reached; when a frame or loaded movie is fully downloaded; via scripting you can also access a system level millisecond timer.

Act: Flash actions (placed "under" a button or in a keyframe) can cause changes to occur. Simple things: Stop; Play; GoTo a particular frame; and GoTo another URL (optionally targeting a specified window). Movie clips can have actions applied to them: "tell target" sends actions *to* a particular movie clip; "duplicate" movie clip to create instances at runtime; start or stop "dragging" a movie clip (so it follows the mouse); and a variety of "SetProperty" controls to change attributes of a movie clip (like location, transparency, and size).

Similar to actions, sounds can be placed in keyframes and, can be synchronized to the timeline (visual changes) or respond to events.

Conditionally: Basic programming controls can be included in Flash "action scripts". "If / then / else" and "loop while" are the two biggies. With the addition of a few simple operators (standard math, comparisons, and string operators) and some built-in functions (like random(), integer(), and string functions like subString()) you can actually "program". A programmer accustomed to a larger set of functions, however, will find it a little tricky. For example, since there's no absolute value function instead you'll need to write "if < 0 then multiply by -1". Authors can create their own custom variables with very lax "type" rules. Indeed there are local and global user variables. Text fields (user editable text) are always associated with a user variable so an author can ascertain the contents of a field (or fill the contents of a field). Finally, authors can't write their own functions (like Director's "handlers"), though, the built-in "Call" function is a powerful way to effectively create "sub-routines" in the form of actions contained in frames (that are kept 'offline'—in a frame the user needn't ever visit).

Tell me about it: If all this wasn't enough... the capability for Flash to "talk" to the rest of the world comes in many forms. The simplest (though it *does* require a CGI script to be "listening") is the standard FORM GET and POST. Just specify an URL (of your CGI script) and send it all the parameters you want. Flash can even load variables received from a CGI script. Flash can also "talk" to JavaScript in two basic forms: via the "FSCommand" action, and through the less supported "embed" technique (which executes JavaScript directly). FSCommand executes a single JavaScript function which can then parse any number of arguments sent from Flash. There are a couple specific rules to follow—but it's a pretty easy technique (for more, see www.teleport.com/~phillip/ucon99/presentation/). The "embed" trick involves using a Flash "GetURL" action (where you normally specify an URL to which you're hyperlinking)... though instead of specifying an URL, you simply type: `javascript:then the javascript code`. For example, try getURL with the following in the URL field: `javascript:alert("hello");` (The semi-colon ends a line in JavaScript.) This will invoke JavaScript to pop up an alert box with the text "hello".

A guest of a different "host": The above FORM and JavaScript commands only work when the Flash movie is embedded inside an HTML document (that is, "hosted" by the browser). But Flash can be hosted by the applications Director, Authorware, and even Dreamweaver. There's an Xtra for both Director 7.02 and Authorware 5.1 which enables a Flash ".swf" file to be imported or linked. There's an interface for the author providing about 10 basic settings (like anti-aliasing, scale, whether to include sound, etc.). Another approximately 70 properties can be modified at runtime with Director's Lingo scripting or Authorware's calculation icon. You can either "get" a particular property of a Flash member (in Director) or icon (in Authorware) or "set" the property. For example, you can "get" the current frame of the Flash movie... or you can "set" the current frame (effectively doing a "GoTo" that frame). The technique is not difficult... the syntax for Authorware is:
`SetSpriteProperty(@"FlashIconTitle", #theProperty,value)`
`SomeVariable:=GetSpriteProperty(@"FlashIconTitle", #theProperty)`

In Director it's based on the sprite number in which the Flash member is placed:
`sprite(5).rotation=value`
`someVariable=sprite(5).rotation`

Some properties affect the single instance (sprite) of the Flash member... others affect the member globally. This same issue is encountered in Authorware as some properties are set with "SetIconProperty" instead of "SetSpriteProperty". It's easier to understand the difference in Director because one member can be used in many places as a sprite.

So much for talking to Flash (even if "get" is more like asking Flash to return information—so far we've looked at the host initiating communication). But how does Flash send messages to a host? For Flash to talk to Authorware, use the "getURL" action and in the URL field type a string. Inside Authorware you'll need an "event" response hanging off an interaction icon where you identify which Flash icon(s) you're listening to as well as specifying the "getURL" event name. This event response will act like an invisible button that gets clicked when the getURL action sent from your Flash movie. To gain access to the string which was "sent" from Flash's URL field use `EventLastMatched[#urlstring]`. The EventLastMatched list variable contains more information, but likely all you'll want to see is that URL string.

For Flash to talk to Director use the same getURL action, but type `lingo:LingoEvent`. For the *LingoEvent* portion you can call a built in handler like `go to the frame +1` or you can call your own function like `kickit()`—provided the host Director file has a handler called "kickit".

Is that all?: Of course not. There are other (less traditional) things you can do with Flash—like export animated GIFs. Also, QuickTime 4 supports a "Flash track" (kind of like another channel of audio—but this one can contain Flash graphics. Just "import" a finished QuickTime into Flash's timeline. Add layers of buttons and graphics, then you can export a QuickTime (no, you can't export a .swf with QuickTime in it). If the button has a GoTo a frame action, it will jump to that part of the video too!