

Overcoming the Limitations of the Browser

Phillip Kerman

Overview:

"The map is not the territory"... the browser is not the web!

Using the web without a browser can be a difficult concept to grasp. However, "browserless" web applications avoid all the incompatibilities between Netscape and IE (as well as various version numbers), poor performing technologies (like Java), and dreaded plugin issues—while still benefiting from all the apparent advantages of the web: up-to-date information, global reach, and interactive communication. This presentation discusses browserless web applications, provides examples, suggests when such applications are viable, and provides some direction as to the tools available (primarily Macromedia software).

Concern	Browser		Browserless	
	Advantages	Disadvantages	Advantages	Disadvantages
Ubiquity	Everyone already has a browser... they invest the time to learn the interface and that investment is paid back over time.	Not everyone has the same browser... plug in, etc.	Even though the user base may be limited—this might be an advantage if you want to limit access to only users with your player.	You have to get your application <u>to</u> the users (possibly via the web). But certainly you can't expect they already have your player.
Ease of production	If you keep things simple, they can often be simple to produce. Even advanced capabilities <i>can</i> be automated.	A more sophisticated application is difficult because you are restricted to browser limits. If you want to do something new, it can be near impossible.	A custom application (once created) can allow for very efficient (even automatic) production.	Sometimes you have to create something from scratch rather than leveraging off existing tools.
Layout control	Only through "window.open()" or fullscreen= yes (IE only). Typeface <i>can</i> be specified, but short of creating graphic text or Flash text, this is not completely controllable.	Probably the bane of web design. If typography didn't matter the web would be entirely mono-spaced text.	Pixel perfect control. Even typefaces can be included (or installed) with today's tools.	Only vector solutions can scale to the user's screen dimensions. (e.g. Flash.)
Continual updates	Pretty much works... just requires the user to actively revisit a site.	"Refresh" doesn't always work, plus exactly when cache is recycled or disposed is not clearly apparent.	All the advantages of the web... just a tad bit more control.	The user has to re-launch your application. Not exactly part of someone's everyday pattern (where launching a browser is).
Skilled people available	Many of people who know conventional HTML skills. Plus, learning is a good investment because there's a need for people.	Learning is constant. Plus you must learn esoteric differences between browsers. If you don't, you'll end up creating pages which don't reach all users. (e.g. use only "FrontPage" and your creation may not look right in all browsers)	Generally, the skilled people who <i>do</i> exist are often very skilled—and aren't limited by the tools.	Much more limited resources. Much fewer Director programmers than HTML editors, for example.
True interactivity	JavaScript not very interactive... Java <u>can</u> be as interactive as you want.	JavaScript capable browser required (and turned on). Additionally, there's a long (15 -30 second) load time. True Java has poor performance (and can be difficult to produce).	Even the simple tools (like Flash) enable true interactivity with limited skills.	When very sophisticated features are desired, it may take more skills—but it's possible.

Solutions to tasks you may want to include in browserless web applications

Launch Default browser

Director: open "http://www.domain.com/" with "c:\program files\netscape.exe" *only works with the path to your browser.

Third party (buddy API http://www.mods.com.au/) solution: `baOpenURL("http://www.domain.com/file.html", "normal")`

Authorware: `JumpOut("", "http://www.teleport.com/~phillip/index.html")` *only works if you include the extension—and that extension is associated with a browser. Third party (buddy API) solution: `baOpenURL("http://www.domain.com/file.html", "normal")`

Flash: Version 3 solution: FS Command, Command= Exec, Arguments= start /m
http://www.domain.com

Notice: The blank () is an "alt-Number Pad 9" (see <http://www.flashzone.com/tutorials/nobat.html> for more)

Temporary Download of media (linking media)

Director: Import a media file using "Link to External File" option into member "TheFile" (This will just be the default file).

`Member("TheFile").filename="http://www.domain.com/filename.jpg"`

Authorware: Import a media file using "Link to File" option. (This will just be the default file).

In the field "File", type an explicit URL... or simply =variableName and define the value of "variableName" before encountering this icon.

Flash: Currently the only "linked" feature (loadMovie) is not available in the stand-alone projector mode.

Reading text from the web

Director: `getNetText("http://www.domain.com/somefile")` (should also use `netDone()` and `netError()` to confirm everything has downloaded)

Authorware: `result:=ReadURL("http://www.domain.com/somefile", 30)`

Flash: Currently no GET capabilities

Downloading files

Director: `downloadNetThing "http://www.domain.com/somefile.jpg", "newname.jpg"`

Authorware: Only available via web player.

Preloading

Director: `downloadNetThing` can be executed any time. You just have to decide when it's appropriate.

Authorware: Only available via the web player (plug in)... but your server configuration can actually preload media based on the "popularity" of certain files others have downloaded (via "Advanced Streamer"). But you have plenty of control within Authorware as well.

Flash: Currently the only "preload" feature (loadMovie) is not available in the stand-alone projector mode.

Uploading data

Director: `postNetText("protocol://domain.com/cgi-script.cgi", dataList)` or
`postNetText("protocol://domain.com/cgi-script-name?arguments", "data")`

Authorware: `PostURL("protocol://domain.com/cgi-script-name?arguments", "data")`

Flash: Currently no POST capabilities.
