

Browser Power for Flash

Robert Taylor & Tyler Wright

Browser Power for Flash: The power to offer internet users a superior experience across internet technologies (or, building kick butt websites with Flash and JavaScript).

The Session

By combining the strengths of Flash with those of the browser you can overcome the obstacles that keep you from providing a great user experience. This session will cover the tools and methods that we've acquired to this end. Topics include greater browser control from within Flash, searchable Flash for SEO (Search Engine Optimization), and the utilization of the Flash Player to dynamically enrich HTML. The goal of this session is to inspire new solutions through covering concepts and presenting concrete examples. This presentation will not cover ActionScript or JavaScript basics, including *ExternalInterface*.

Who Should Attend?

If you're interested in building more capable HTML websites, if you'd like your Flash content to be marketable through search engines, or if you want to give your Flash movies ultimate control over their browser environment, this session is for you.

Designers, developers and businesses alike will benefit from learning the *why's* of Browser Power and the techniques to make it happen.

Protocols Reviewed

getURL

getURL was one of the first scripts introduced to Flash and seems to be supported universally (with the exception of IE 4.5 on the Mac). *getURL* is primarily used to navigate between web pages and communicate with the server. However, with some creativity it can be a powerful tool in taking control of the browser environment. Its communication is only one direction, from Flash to the browser.

ExternalInterface

ExternalInterface is the simplest solution for direct communication between JavaScript and Flash to date. It is widely supported by all major browsers, can send complex data types, and its calls are synchronous. It is also fully packaged with Flash Player 8, requiring little additional ActionScript. We will use *ExternalInterface* for all of the demonstrations in this session.

Browser Power for Flash

A Content Driven Internet

Background

In the past the internet has been described as a lake of information, miles wide but only an inch deep. The successes of cooperative communities such as Wikipedia and search technologies like Google have made the web an invaluable source of information. Even now with the advent of rich internet applications the web is primarily content (read HTML) driven.

Adding a Touch of Flash

In any HTML website there are limitations that can be overcome through combining technologies. By decorating the HTML page with a touch of Flash users can better enjoy their search for content. It is your responsibility to uncover the solutions by evaluating the strengths of Flash and applying them in a content driven internet.

The Strengths

Flash began as an animation tool and continues to provide rich media with a small footprint. It also presents a variety of development tools not yet available to browsers.

Utilities such as a file upload tool can be built in Flash allowing users to now appreciate queuing up multiple files and watching the progress of an upload.

Dynamic Fonts: Flash can embed fonts and use them dynamically

Rich Graphical Representations: The Flash drawing API and tweening abilities are the beginning of endless solutions

MP3 Player: Flash can load and play (even stream) MP3's within the page

Video: With high quality compression and support of alpha channels Flash can add interactive video to websites and ads

Live Streaming Audio/Video: Through microphones and/or webcams Flash can strengthen web communities.

Socket Communication: Through Flash a socket can remain open, sending and receiving live data (live support)

Local Storage: Flash has a 100 K cookie storage that doesn't clear with the internet cache. It will even serialize objects.

File Upload/Download: Through the *FileReference* in Flash the user is presented with a better experience when uploading or downloading files from the server.

Browser Power for Flash

Searchable Flash Content

The Challenge

It is important in today's content driven internet to make that content accessible to the user. The user relies heavily on search engines to find the content they're looking for. Because of the interactive nature of Flash, the SWF format will always have trouble being indexed by search engines. Flash must rely on the browser for Search Engine Optimization.

Choose Your Own Adventure

Traditionally there have been mirrored websites that require users to choose between a Flash or non-Flash experience. Without the "choose your own adventure" splash screen it is still important today to have valuable content available in some form as HTML. Through a multi-tiered solution to this content presentation we will determine the users capabilities and intelligently (invisibly) choose the highest quality experience. One of these content tiers is the most helpful for search engines: HTML. As users see in color (images and styles) search engines see only black and white (list item and header tags).

Guiding Principles

There are multiple solutions for realizing Flash SEO but all of them should be used under these guiding principles.

Always something: No matter what the users' capabilities they should always receive *something*. The user who has Flash installed will get the Flash experience. The user without may receive upgrade instructions along with the styled HTML version of the site. If the HTML is structured properly then users who are using text-only browsers (on a mobile device, for example) can get well formatted, organized content. It is this type of structured HTML that is optimal for search engines.

Sharing content: Whether they share a common content management system (CMS) or another source of data, layers should share content. The approach becomes more manageable to maintain and will make the tiers of experience consistent.

Moment of Choice: Most often a user visits a site with a purpose. That purpose is *not* to choose between forms of content that are offered. Make the choice invisibly, based on system capabilities and bandwidth. It can be beneficial to offer the higher experience discreetly.

Choices Everywhere: The ability to present the best user experience should be built into every page of your website that uses these solutions. In this way a user can link to a page from a search and still receive the correct experience.

Browser Power for Flash

Flash Browser Power

Taking Control

So far we've covered various ways to use Flash in the browser. Now it's time for Flash to take control of its environment. Through *ExternalInterface* and *getURL* it can *define* and *drive* JavaScript and utilize information from its container. No longer at the mercy of the browser Flash will extend it to overcome inherent weaknesses and achieve the greater user experience. With JavaScript as the tool for browser control we can evaluate its strengths (even those that are hidden) and discover how the technology can best compliment Flash.

The Strengths

The browser is JavaScript's domain and Flash is simply a guest. Though implementations vary between browsers, with standards and frameworks it has become possible to build robust applications that succeed across the internet.

Universal: JavaScript is available in all major browsers natively, requiring no download or installation.

Native Language: JavaScript can communicate with and drive the browser directly.

Creation and Manipulation: All aspects of HTML and CSS can be modified and even created at runtime through JavaScript, including the Flash object.

Open: JavaScript has a strong open source community behind it with countless frameworks and development resources available.

Embedded JavaScript

A powerful and little known capability of Flash (since Flash 3) is to define JavaScript from within ActionScript. Through Flash's *getURL* entire methods and packages of JavaScript code can be *created* in the containing browser at run time. By porting the JavaScript with ActionScript in a single code base we are no longer dependant on the HTML page. The SWF can be place inside any HTML container without consideration for external script files. The code also benefits from the automatic SWF compression which is not guaranteed for JavaScript loaded via the HTML.

Example:

In this example a function is defined on the webpage called *jstrace*. The function will be called when the mouse is pressed, displaying a message through the browser alert.

```
getURL("javascript:function jstrace(str) { alert(str); }");

function onMouseDown()
{
    var message = "Hello World!";
    getURL("javascript:jstrace(' " + message + "')");
}
```

Browser Power for Flash

Hybrid Components

Flash components or classes that contain JavaScript can be added at anytime during the development of a Flash project providing immediate results. There are several considerations to make when compiling JavaScript with the SWF. First, it is important to define unique methods and properties under a common scope in JavaScript. Because the code is hidden within the SWF it can be difficult to foresee naming conflicts with other scripts in the page. Second is to make sure that whatever functionality that relies on JavaScript will not ruin the user experience if JavaScript is disabled. This can usually be avoided by initializing with several test calls between Flash and the browser to ensure communication available. Lastly it is advised to build and test the JavaScript in the browser environment where it can quickly be modified and validated across browsers. By following this development process you will more easily isolate the bugs rather than debugging across several technologies. It's also good practice to detect whether the SWF is running in the test player (*playerType* "External") and disable the *getUrl* and *ExternalInterface* calls. It gets bothersome to have your browser popping up a blank page each time you test a movie that has embedded JavaScript.

Browser History and Bookmarking Control: Give your Flash movie the ability to respond to the browser back and forward navigation, load appropriately to bookmarks and other link-keeping.

Custom Right-Click: JavaScript can help ActionScript disable the native Flash Player right-click menu, allowing you to create custom menus or alternate functionality.

Frame Control: Create and manage iframes that appear to reside *inside* your Flash movie. Flash just became capable of displaying HTML.

Universal Trace: A simple method that will *trace* output to an HTML popup depending on a test-mode flag.

OnUnload: respond to the *Window.onUnload* event, allowing Flash to do any last second clean-up.

Others: Full auto-update solution from within Flash, style sheet changes, accessing the GET string from the page url, easily resize the Flash Object, and many more.