

Browser Wars II: The Rebels Strike Back

By [Brennan Spies](#) - Posted on October 8th, 2008

Tagged: [Browsers](#) [Google](#)

In the heady days of the mid-1990's, a company called [Netscape Communications Inc.](#) rode the wave of the emerging Internet excitement to technology stardom. With its flagship product, Netscape Navigator, controlling over 90% of the web browser market and a stock price that reflected that, the company seemed destined for greatness...but, alas, dreams can disappear quickly in the tech industry, and Netscape's dream was a particularly brutal example. Microsoft, sensing an opportunity in the emerging Internet as well as a potential threat to its mainstay desktop business, launched itself into the browser market in late 1995 with the release of Internet Explorer 1.0 (for Windows 95), based on code licensed from Mosaic. The first browser war had started in earnest.

The First Browser War

The first browser war was really a war to determine the shape of the Internet experience. During those early days of the web, most pages were simple text and hyperlinks; the race to provide developers and users with a richer Internet experience became the central battleground between the two browsers. Much like the Cold War, it was also about achieving parity: JavaScript, Java applets, media tags--if one side had a particular technology or feature, the other had to get it quickly or risk "falling behind". Often this was done with proprietary extensions, in part because organizations like the W3C could not create standards at the "speed of the web" but also no doubt in an effort to lock web authors into focusing on a given browser platform. (In a sense, some of the biggest casualties of the war were probably the web standards themselves--something we still feel today.)

In terms of product development, the two sides moved in opposite directions: while Netscape focused on being the "office suite" of browsers (its Communicator product), Microsoft focused on being lighter and faster. The second approach ultimately turned out to be what most users really wanted. Furthermore, because of the breakneck speed at which many of the features that Netscape introduced were developed, they were often criticized for being unpolished and buggy, something which did not help the image of the Navigator platform.

The turning point came in 1997, when Microsoft released Internet Explorer 4.0, a faster and more standards-compliant browser than Netscape's Navigator. (It was also, somewhat controversially, the first version that Microsoft integrated into the Windows operating system itself.) Simply put, it was a better browser than Navigator. Coupled with the fact that most users simply used the browser that "came with their PC", and Internet Explorer's competition with Navigator quickly became a rout. By 1998 the writing was on the wall. In a last ditch attempt to stave off Microsoft, Netscape sold itself to AOL in that year. Ultimately, this did little to prevent the hemorrhaging of Navigator's market share to IE and, to add insult to injury, AOL was even forced to continue using Internet Explorer as the basis of their product line because of a pre-existing agreement with Microsoft.

The rest is history. Internet Explorer 5 and 6 went on to dominate the browser market (achieving 96% of the browser market in 2002), "Netscaped" *became a verb*, and the victorious giant finally rested, secure in the knowledge of its complete dominance. After all, who would seriously challenge them in the browser market again?

The Second Browser War

One of the other things that Netscape did in 1998 was to release their browser code as an open-source

project, a move which led to the formation of the [Mozilla Foundation](#). Though this did not keep them from losing the mindshare of the Windows-using computing mainstream, it did keep Navigator relevant in areas (such as Linux) where Microsoft did not care to go. It also helped spark developer interest in the project, despite the complexity of the [Gecko](#) code base.

Enter Firefox. One such developer was [Blake Ross](#), a young programmer who began contributing to the project soon after it was open-sourced. Together with Dave Hyatt, he created the first version of the [Firefox](#) web browser*, a lightweight and much improved version of the Gecko browser. Version 1.0 was released in November of 2004. With nice features such as a dedicated search toolbar and multi-tabbed browsing, Firefox was the right browser at the right time. Helped by Microsoft's general neglect of Internet Explorer as well as a string of well-publicized security vulnerabilities in the dominant browser, Firefox's market share began to chip away at IE's market share. As of September 2008, Firefox's estimated share of the browser market was just under 20%, an amazing turnaround. Version 2.0 of Firefox, released in October 2006, added a phishing filter as well as the ability to restore browsed web pages after a browser shutdown/crash.

Shaken by the sudden and stellar rise of Firefox, Microsoft went back to work on Internet Explorer, releasing version 7.0 of the browser in the same month as the release of Firefox 2.0 with "me too" versions of the search bar, multi-tabbed browsing, and phishing filter. But IE 7.0 did little to improve on IE 6.0's poor standards compliance (especially in the area of newer CSS standards), and the browser is noticeably slower than the previous version. It also did nothing to address the developer-unfriendliness of the browser: decent JavaScript debugging required an investment in Visual Studio, a stark contrast to the excellent and freely available [Firebug](#) plug-in for Firefox (though they did finally improve DOM and CSS inspection with the [IE Developer's Toolbar](#)). Despite its still-dominant market position, IE had suddenly found itself in the uncomfortable position of having to play "catch up" with Firefox.

Mozilla struck back in June 2008 with version 3.0 of Firefox, featuring significant speed improvements in Gecko 1.9 as well as other nice features like the ability to "tag" bookmarked pages, a preview of some WHATWG (later to become [HTML 5](#)) API elements, [Acid 2](#) compliance, and an implementation of [JavaScript 1.7](#). Current work on Firefox 3.1 includes more support for HTML 5 (specifically and tags), full CSS 3 selector support, as well as a brand-new and wickedly fast JavaScript engine code-named TraceMonkey (covered in more detail [here](#)). But issues such as Gecko's high memory usage--related to its extensive use of [XPCOM](#)--continue to be an issue in Firefox, hampering optimizations as well as making the browser unsuitable for embedded use. Though big strides have been made in reducing both XPCOM's and Firefox's memory footprint in Firefox 3.0, Mozilla has stated that improving/removing XPCOM APIs will be a major focus of the 3.1 and 4.0 releases.

Microsoft, for its part, is not standing still. In August 2008, they released the [second public beta](#) of Internet Explorer 8. Numerous small improvements to standards support (most notably full support of CSS 2.1) allowed IE 8 to finally pass the Acid 2 test. Though these changes also break many pages written against older versions of the IE browser, an IE 7 and "quirks" mode allows these pages to co-exist with more standards-compliant ones. On the downside, however, IE 8 remains [one of the slowest](#) browsers.

But as Microsoft and Mozilla continue to slug it out in successive versions of their browsers, the sidelines have been anything but quiet.

Enter Webkit. When Apple chose the KHTML engine (powering KDE's browser-cum-file manager [Konqueror](#)) over the more well-known Gecko engine as the basis of its [Safari](#) browser in 2003, the decision raised some eyebrows. The [crux of the decision](#) essentially boiled down to the complexity and memory-consuming nature of the XPCOM-based Gecko code; for these reasons, the engineers at Apple felt that the long-term maintainability and extensibility of their browser development would be better with the

KDE-developed code. Apple, in turn, released its improvements to the KHTML engine as the [WebKit](#) project.

Apple's decision would turn out to be prophetic when Google's long-rumored entrance into the browser market turned out to be based on WebKit. With the release of [Chrome](#), Google put its substantial weight behind the lightweight browser. But Chrome is not just WebKit with a pretty face. Google changed the browser so that by default each tab would run its own process, a feature touted as contributing to greater browser stability and security. They also developed their own fast JavaScript engine for Chrome, dubbed "[V8](#)". Finally, Google spent a considerable amount of time optimizing the WebKit browser itself, a very noticeable improvement when one considers the speed of Chrome (or the WebKit nightly build) in comparison to Safari 3.1.

With the recent [announcement](#) that WebKit is the first browser to fully pass all [Acid 3 tests](#), the project has proven that it is not just fast, but a serious contender in the area of standards compliance as well. Of course, compatibility in the real world is a mixture of standards and quirks, so being an everyday-use browser requires enough uptake from end users that developers make sure to test their sites on that browser. This is the hurdle that WebKit/Safari/Chrome still must clear.

The Battle of JavaScript

The [announcement](#) of a new JavaScript engine by the WebKit team, dubbed "SquirrelFish", in June 2008 set off a race for the fastest JavaScript implementation. Mozilla fired back with [TraceMonkey](#), a new trace-based JIT compilation implementation (more [here](#)). Numerous [benchmarks](#) showed that Firefox would not be bested in the JavaScript game. Then Google got into the game with its [V8 engine](#), which itself was a real [barn-burner](#). Then the WebKit team announced "[SquirrelFish Extreme](#)" with [benchmarks](#) showing it to be much faster than either V8 or TraceMonkey. And it isn't over yet...with TraceMonkey still in the early stages of development, expect more big announcements from the Mozilla camp.

Who's the fastest? Depends on when you ask that question (and who's benchmark you use).

The Battle of Standards

Poor support of web standards resulting from the first browser war galvanized groups like [The Web Standards Project](#) into action. Its Acid tests have become one of the key benchmarks of standards compliance recognized by every browser maker. They have helped visibly promote compliance to key specifications such as XHTML 1.0, DOM 2, CSS 2/2.1/3, and JavaScript/ECMAScript. With the acceptance by the W3C of WHATWG's [HTML 5](#) standard, support for key features of the specification (e.g. `<canvas>`, `<video>`, `<audio>`) will soon become a key success criterion for any browser as well.

The Story Now

Firefox. Even with all the improvements to Gecko embodied in Firefox 3.0, the browser still struggles with memory usage requirements due to continued dependence on XPCOM. Mozilla has learned its lesson from the first browser war, though; expect them to continue stripping out or optimizing XPCOM-based code in versions 3.1-4.0 to make Gecko lighter and faster (and more embeddable). If they can do this, the maturity of Gecko's code base and the plethora of available Firefox plug-ins give them a slight edge over WebKit. TraceMonkey will make JavaScript code lightning fast in Firefox 3.1, and this will open new doors to web developers.

Internet Explorer. Inertia favors Microsoft. Users who won't be bothered downloading a new browser (outside of Windows updates) are likely to stay with IE. But this is no longer as great an advantage as it was in the days when 56k modems were considered fast. Furthermore, with more and more users becoming used to storing photos online, writing blogs, or in general living on the web, expect alternate browsers that offer

distinct advantages over IE to continue to grab more of a share of the market. How much IE 8.0 can stem this loss is hard to say: the need to backwardly support pages written against IE 6 or IE 7 may well become the browser's Achilles' heel, leading to code bloat and generally slower performance. IE is still very much behind on general browser and JavaScript performance: the IE team has their work cut out for them.

WebKit. The little browser upstart has been on a roll lately, pumping out SquirrelFish Extreme and Acid 3 compliance in rapid succession. Having Google engineers on board the project has no doubt helped. But the Safari and Chrome browsers still have their work cut out for them in supporting *real world* web pages, quirks and all. The excitement surrounding the Chrome release may well bring enough web developers to start testing their pages against WebKit, which in turn will give the browser more "everyday use" credibility. The lightweight code base (about a tenth of Gecko code) and browser speed will give them an advantage in the embedded market, but Chrome/Safari will have to attract a more vibrant plug-in community in order to really displace Firefox as the "alternative" desktop browser of choice.

Opera. Back in the late 90's Opera was known as *the* lightweight browser, and actually made a decent living selling browser technology at a time when Netscape was unable to do so. (Disclosure: I used to work for Be Inc.--which had the dubious distinction of having *almost* been the next operating system for Apple's Macintosh--and we had a relationship with Opera where its browser was embedded in our Internet Appliance O.S.) Though it has been noticeably absent from all the hubbub surrounding Firefox and Chrome, an internal build of Opera has *quietly* become the second browser to pass the Acid 3 test. Opera continues to produce a nimble and mature browser which is very embeddable (Opera Mini).

Summary

It's an exciting time to be a web developer. The stagnation that had gripped the browser market is finally starting to lift in a dramatic fashion. JavaScript speed is approaching that of native C code, enabling the developer to create some pretty amazing things right in the browser without the need for plug-ins. Unlike the first browser war, the second is likely to vastly improve the story of standards compliance, a welcome development for anyone who has ever had to spend time on quirksmode.org. Yes, browser incompatibilities will still exist, but anything that can be done to minimize them is welcome relief.

And for those of you who are curious, I am writing this article on Chrome, but my everyday browser is still Firefox. Long live the war!

* *Originally (and more appropriately) named Phoenix, after the mythical bird that rises from its own ashes.*

[Web Browser Timeline \(Wikipedia\)](#)
