

27 September 2008

By: Marius Oiaga, Technology News Editor

[IE8 Performance vs. Google Chrome and Firefox](#)



IE - Firefox - Chrome
Microsoft, Mozilla,
Google

JavaScript lessons

With Microsoft making headway towards the gold build of [Internet Explorer 8](#), the Redmond company has to face an ugly truth. Performance-wise, with emphasis on JavaScript performance, the software giant is getting ready to release a browser inferior to what is already available from rivals Google and Mozilla. Microsoft's aim is to make the next iteration of IE superior to what Internet Explorer 7 brought to the table back in 2006 on Windows XP and the start of 2007 on Windows Vista, and in this regard the company is on the right track. However, there is little focus on shifting Internet Explorer 8 into high gear and making it outrun Firefox 3.1 and Chrome. nbsp;

When he launched Internet Explorer 8 Beta 1 back in March 2008, Dean Hachamovitch, general manager Internet Explorer, revealed that "some of the tests we have done show pure JScript performance improvements up to 2.5 times. We also measured the performance gains on common Gmail operations, like loading the inbox (34%), opening a conversation (45%) and opening a thread (27%) compared to IE7."

The fact is that, with Google and Mozilla praising the JavaScript horsepower under the hood of Chrome and respectively Firefox 3.1, Microsoft cannot afford not to make "speed of the essence," although this is exactly what the company is doing. Stephane Kimmerlin, product marketing director, Windows client business group, Asia-Pacific, Microsoft, told [ZDNet](#) at the beginning of September that "when we designed IE8, we did not start with performance in mind."

[Christian Stockwell](#), IE program manager, said at the end of August 2008 that Microsoft would not be joining the chorus of browser developers trumpeting their product as the fastest in the universe. And believe it or not, there's a good enough reason why Microsoft is not applauding the performance superiority of IE8 over that of its rivals... because it's simply not there.

The need for (JavaScript) speed

The fact is that Microsoft has so far managed to avoid making their JavaScript benchmarks for Internet Explorer public. While Google has the V8 Benchmark Suite, the WebKit Team has SunSpider and Mozilla is offering Dromaeo, the Redmond company continues to remain loyal to its proprietary strategy with IE. In this regard, the conclusion presented by [Asa Dotzler](#), Mozilla's community coordinator in the past, is that Microsoft is simply falling far behind the developers of open source browsers when it comes down to speed. Dotzler noted that Microsoft simply "can't keep up" with open source projects and that it's "a shame that they're falling so far behind" with IE.

Is IE8 evolving in terms of JavaScript performance? Undoubtedly. Just as undoubtedly as the fact that Google Browser (Chrome) and Firefox 3.1 have already evolved past the stage where the next version of Internet Explorer is now. Whether Microsoft likes it or not, Chrome and Firefox 3.1 are "state of the art" in terms of JavaScript performance, while IE8 is lagging behind, with no consistent push from the company to make the browser measure up to the standards of its rivals.

What did Microsoft do with IE8?

"When we took a hard look at our goals and considered what we could do to build the best browser we were presented with a quandary. On the one hand, we could focus very narrowly on scripting performance, trusting that our investment would noticeably improve our users' browsing experience. Alternatively, we could invest more broadly in realistic scenarios, measuring heavily-used subsystems and investing our optimization effort accordingly. We opted for the latter approach," Stockwell explained.

In translation, Microsoft abandoned the idea of focusing on boosting JScript and JScript alone and went a different way, namely optimizing the browser for top usage scenarios. But in this context, Microsoft has left itself wide opened to a perception problem. And make no mistake about it; just as it was the case with Windows Vista, while poor performance is survivable, the generalized consumer perception of poor performance however acts as a deal breaker.

Firefox 3.1 TraceMonkey

For [Firefox 3.1](#), the successor of Firefox 3.0 and the next iteration of its open source browser, Mozilla introduced native code compilation JavaScript engine ("SpiderMonkey"). Just make sure to remember the key phrase "native code compilation." This created TraceMonkey. It's rather simple; Mozilla is cutting down significantly on the interpreting aspect and is increasing the focus on native code. The next generation JavaScript implementation in Firefox 3.1 uses a trace as the compilation unit.

By turning to traces in order to compile JScript "just-in-time" and renouncing to utilize functions or code files, Mozilla ensures that the JavaScript engine performs less interpreting and executes JS applications directly in native code. The raw beauty of the new "trace trees" technique and tricks that evolved SpiderMonkey into TraceMonkey is the loop optimizations made possible by the trace (sequence of instructions) for patch executed repeatedly which are no longer interpreted. [Mike Shaver](#), Mozilla's chief evangelist pointed out that Firefox 3.1 is competing directly against native code.

Google Browser Chrome V8

Remember the "native code compilation" key phrase for Firefox 3.1 and TraceMonkey? Guess what?! The same is valid for [Google Chrome](#). In fact, [Lars Bak](#), software engineer, revealed on the launch of Chrome that one of the cornerstones of the browser was the fact that its JavaScript engine was capable to compile source code directly into "native machine code." In this context, while Firefox 3.1 still performs some interpreting in addition to using traces, Chrome has no interpreter; compilation is done directly in native code. In addition, deploying virtualization and turning to "hidden classes and inline caches," Chrome delivers additional optimization, as the dynamic hidden classes streamline access to JavaScript objects.

Swimming in Native Code

This is what Microsoft needs to make Internet Explorer do, although it looks like it's already too late for IE8 with a reported Release to Web deadline set for November 2008. Microsoft's Christian Stockwell touted Jscript performance gains of 400% with IE8 compared with IE7 as far as the SunSpider benchmarking suite is concerned. The Redmond giant did introduce optimizations when it comes down to JavaScript-DOM and JavaScript Object Notation. But at the same time, while the company too should have focused on native code, it didn't.

Microsoft has indeed worked on IE8 performance, from runtime to memory optimization, but also on taking the AJAX subsystems a step forward, and the actual evolution of the existing JavaScript engine of the browser. However, this does not change the fact that Internet Explorer needs a native code compiler so as to at least keep the same pace as its rivals.

Internet Explorer 8 Beta 2 is available for download [here](#).

Google Chrome is available for download [here](#).

Firefox 3.1 Alpha 2 for Windows is available for download [here](#).

Firefox 3.1 Alpha 2 for Linux is available for download [here](#).

Firefox 3.1 Alpha 2 for Mac OS X is available for download [here](#).

[What's New for JScript in IE8?](#)