

Brendan Eich

The new browser war

JavaScript's creator details the challenges of getting to Version 2.0

YOU COULD ARGUE THAT BRENDAN EICH has had as much, if not more, influence on modern-day browsers than anyone. At Netscape Communications, he created JavaScript for the Netscape Navigator browser. He was co-founder of Mozilla.org and helped spin out the Mozilla Foundation, where he served as lead technologist and a member of the board of directors. In 2005, Eich became chief technology officer of Mozilla Corp. and is currently readying Firefox 3.

Eich is also hard at work on JavaScript 2, a job complicated by disagreements within the Ecma International standards body, which standardized JavaScript as ECMAScript and is working on Version 4 of the language.

Some of Eich's disagreements are with Microsoft's participation in the process—seemingly opening up old wounds from the days when Microsoft killed off Netscape.

Eich sat down with

to talk about a range of topics. This excerpt from their discussion focuses on JavaScript and where it is going.

What's up with JavaScript 2? Where does it stand, and when will we see it in browsers?

Since we're working in this context of Ecma [International], the standards body, it's really not possible to say when you'd see it in products; it's up to the various members that are working on it.

'We're not trying to have any JavaScript 2 differentiation vis-à-vis the standard, as Netscape might have done.'

Senior Editor Darryl K. Taft at Mozilla's headquarters in Mountain View, Calif., earlier this spring

There was a big, well-known flap about it last fall, where it became clear that Microsoft and Doug



Crockford at Yahoo were against the big improvement to the language. We were trying to do the language under the fourth edition number. ECMAScript 4 and JavaScript 2 are the same thing. And, actually, in our detailed spec where we talk about versioning, we make those two the same thing.

So we're not trying to have any JavaScript 2 dif-

ferentiation vis-à-vis the standard, as Netscape might have done once upon a time.

... It goes back to 1999, when the third edition was finished, and a lot of the development and design work that was done under Waldemar Horwat, who is at Google now but was at Netscape then, fed into actual languages—

[CONTINUED ON PAGE 48]

MOLLY E. HOLLSCHLAG/MOLLYCON

EICH FROM PAGE 47

mainly, ActionScript in the Flash player and Jscript.net, Microsoft's ASP.NET, which is in the CLR [Common Language Runtime].

Waldemar is the guy I gave the keys to the kingdom to in 1998, when I went off to found Mozilla.org. I'd done JavaScript and taken it through standardization at Ecma, which hadn't really done much software before then.

A lot of people contributed. Sun contributed, Microsoft certainly contributed. And [Microsoft was] very interested at that point in gaining market share on Netscape. They were the minority browser, so they were saying, "What's this Netscape churn that's going on from release to release?" We want standards. And they helped make the standards. And they helped make them be real-world standards. Some of the personalities at Microsoft were very bright, and they would have liked to change some things. But as we got into it, they realized that any change meant that if they made the change in their implementation first, as the minority browser vendor, they might not work on a certain page. And then Netscape might renege or be late, and they'd lose market share.

Well, there's the same dynamic now, but it's Firefox and Safari that are

on the minority side of it, and we're trying to gain market share. And we're interested in real-world standards. So, you think we would be super conservative.

But it's a funny thing that's happened to the Web: The Web is a mess. It's formally unsound. It's a bunch of de facto and du jour standards; browsers kind of agree, and content authors worry about the top few browsers when they write something and roll it out. You

thing—which was Phoenix instead of Firefox—and making a usable browser and just a browser, we timed the market. We were ready when the time was right. And that got us huge market share because IE was in terrible shape—the thing with security holes and pop-ups and so on.

But Microsoft improved on that.

Yes, and now people are saying Microsoft's back in the game. They're doing more work on standards,

'The Web is a mess. It's formally unsound. It's a bunch of de facto and du jour standards.'

still see stuff rolled out on some of the big sites that doesn't work in Safari right away. But it's getting better. So there's this sort of intersection between the browsers and what works—like the Venn diagram and the intersection semantics. And it moves over time. And it can be moved if you put some energy into it. Web developers do want to do new things. They do want to make their life easier, and they do look at certain browsers first.

The problem is Internet Explorer—Microsoft used its monopoly and stagnated the Web. And that was huge for us because by simply persevering and focusing on the right

[but] they're doing some of their own proprietary things, too.

Like, in IE 8, they have ways of doing cross-site requests—background I/O—and they did not propose it to the [World Wide Web Consortium]. They did not work with the W3C in the standards group that's trying to do that.

So, it isn't really clear where they'll go. Because whatever they do in IE, they're very concerned about compatibility. So, they don't want to "break the Web," as they put it. Or they don't want to break their intranet customers. All the enterprises that used IE with an ActiveX markup—that's an even

bigger constraint on what they can do with the browser than the public Web is. So, [Microsoft is] very conservative about some things, but they're moving forward with IE, which is good and makes for competition.

On the other hand, you see Silverlight, and you see real innovation coming in from the CLR/.Net world with the graphics stack [Microsoft] did for WPF [Windows Presentation Foundation]. And that seems to be where they like to focus. Because it's their thing, they can move it faster, and they can take advantage of the OS. And what we try to do with Mozilla, at least—and I think we're probably in alliance with the other minority share browser vendors—is not only say, "Let's not break the Web, but let's move the intersection semantics forward fast, but in a cross-browser way that kind of competes with that Silverlight stuff." Why should you have to get locked into that? Why shouldn't you be able to just use SVG [Scalable Vector Graphics] and better JavaScript?

But it's tricky because, in moving that forward, you kind of have to get the browsers to all jump at the same time. And if IE doesn't jump, you have to figure out what to do about it. And we have actually thought about that a lot.

So, for JavaScript, there's a real problem—as you

evolve the language, you're going to introduce new syntax, new ways of saying things, that don't work in old browsers. They'll cause the old browser to say it's a script error, and maybe it'll keep it to itself because there's no need to bother the user, but it will stop the page from working in the way it would in a new browser. So you have to be willing to make old scripts for the old browser and new scripts for the new browser that use the new language features. And that's twice the work at the limit, and who wants to do that?

... Some things have improved—the ability to update your user base in order to patch security bugs or because you have a Windows update or OS update mechanism. That's pretty key. And that's something Mozilla's interested in keeping pure. We don't want to use it to channel extra products to you or force things down your throat, because then you might reject the security patch, and we really do want you taking the security patches.

So if we have a better update story and we have a good way of getting people to go to the next major version of the browser and it has some new features, maybe JavaScript 2 will get out there faster. But if Microsoft is saying, "No, we're not going to do it, we don't think it's good for developers, and we'd like to do other things that

we think are higher priority," then it's going to be a problem.

So what do you do?

One of the projects that I started last summer is Project Screaming Mon-

completeness before we start to build a community around it and make it downloadable and pre-built so you don't have to build the source. And you can try it out in your own IE on Windows, and

'[IE is] an extensible platform, and we're going to use that to extend it to support JavaScript 2.'

key, which is this project to take the engine for a language—ActionScript, which is a precursor to JavaScript 2—and make it an active scripting engine for IE. And that project is coming along nicely. Mark Hammond, who is a big Windows Python guru and well-known hacker in Australia, is working for us, and he's been making that work.

At Microsoft MIX [in Las Vegas in March], [Microsoft] showed a chess demo to boost Silverlight and to beat up on JavaScript and the browser being slow. Well, Mark has Screaming Monkey working and running that chess demo, and it's not quite as fast as C#, because I think [Microsoft] pulled out all the stops to make that version fast, but we're running, like, within an order of magnitude of the speed, and we're coming up. So, Screaming Monkey lives, and it just needs a little more polish and

you'll have something like ES4 [ECMAScript 4] or JS2 [JavaScript 2].

It's actually using a compiler written in ES4—so, a self-hosted compiler. And that's always been an issue with Flash. Flash ActionScript was this run-time-only version of the JavaScript language that took out the ability to compile a program at run-time and evaluate it. That feature has been wanted. That's coming back through this self-hosted compiler. So you will be able to use it just like you use JavaScript today, and all we're doing now is making sure the performance is right. And that's going to take another little while.

But if Screaming Monkey is capable and performs ... And, here's the other if—and I can't tell you if this is going to happen—but if Adobe decides to ship that ActionScripting thing along with Flash 11 or whatever, then you could wake up in a year or two

and see that there is support in IE for JavaScript 2 and Microsoft didn't have anything to do with it.

IE is such an extensible platform. [Microsoft] did do a good job of that. Microsoft does know how to do platform. And they did it for their own purposes and their own enterprise customers as much as anything. They wired all of the mid-'90s automation interfaces—COM and OLE—into IE, and they built IE on that. And they built that into the Windows shell. This all came out in the anti-trust case because they said, "We can't remove it," which wasn't quite true. But [IE is] an extensible platform, and we're going to use that to extend it to support JavaScript 2 without their help.

Microsoft will have no say in the matter?

Well, they could try to break us. They could say, "Oh, I can't support this; this is a security hazard. Screaming Monkey is going to bite my nose off, and I'm going to outlaw that MIME type." That probably gets them into some antitrust concern, because it looks anticompetitive. It looks like they're trying to fire-wall Silverlight and the CLR.

If they try to break the general app description facility, they're going to break a bunch of other customers and enterprise people who've added their

[CONTINUED ON PAGE 50]

EICH FROM PAGE 49

own scripting engine like CobolScript or whatever.

So, I don't think they can do it. I think they're just going to have to take the risk and try and say we have better tools, we have better run-times. And for some definition of better, for a lot of professional developers, they do. There's no disrespecting Microsoft on that front. That's been one of their strengths, and it will continue to be. VisualStudio is a powerhouse.

But the Web is a low-cost disruption that's always chipping away at that and always saying: Well, do you really want to lock yourself into that and only target Microsoft and what Microsoft supports? Sure, Silverlight runs on the Mac. They've got Miguel [de Icaza of Novell] doing a Linux version. It's kind of like they took out the old "Kill Netscape" playbook and they're using it against Adobe Flash. The same thing happened with Netscape, and of course they dropped the Mac port. And the Unix port of Internet Explorer was contracted out and died a long time ago.

So, I wouldn't be shocked if in the worst case that happens again with Silverlight. But I'm here to prevent that with whatever efforts I can put into it. And part of

the effort is making JavaScript 2 widely available. So we don't want it to be just a Firefox thing. We hope Apple and Opera support it. And Opera's been great in the Ecma group. The people they've brought to bear have been really top-notch. Apple has not really been that involved, but I've been talking to them for over two years, and they're following it. And we're

just using their own YUI [Yahoo User Interface] or Dojo name as an object to hold other names. And it's by convention, it's nice and it's very low integrity. People can attack it. You can make a mistake that stomps on somebody. So there are both security threats and just error and tolerance problems with that. And we'd like to make that better. We'd like to make it possible for developers to

make more static judgments about code for security, for instance, or for knowing what names mean so nobody can mess with you.

It sounds as if JavaScript can take on a lot.

Well, the fear is that JavaScript will turn into Java, and it will become this very bounded and disciplined language, and it will be painful because programmers will always have to declare everything, and it'll be used in a way that is rigid and inflexible. We're not doing that. It's still JavaScript; it's still a dynamic language. There's no Java mandatory requirement to always write down your types and declare your variables. You can do anything that goes today. It's backward compatible.

But as you begin to scale up your code, you're going to want to lock a few things down, like your APIs that you present to the other libraries and to

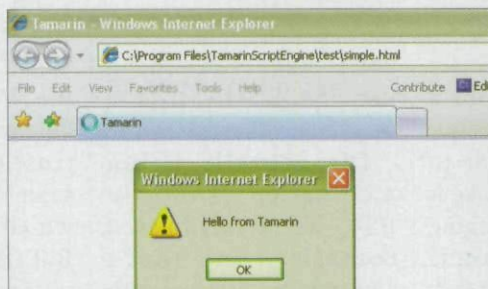
other applications—those generally you start out prototyping. You really don't know for sure what you're doing and how to express what you're doing in the language. So you play around and you do some prototypes. As you evolve that, especially if it becomes a hit, then you start to get a better idea of what things are. And for that version at least, you ought to be able to make things a little more controlled.

So, we're trying to improve programming in the large. And that's the real goal.

So JavaScript 2 is really going to take until this fall to really get the spec done. And we're on track. We've cut some things. We've made some important sacrifices, I would say, to get it done soon. Part of that is because we can't keep having it go on and become like a Perl 6.

What else needs to be done to push JavaScript 2 forward? Well, I can tell you a little bit more. In order to try to make Microsoft happy inside Ecma—even last year, when I was the so-called convener of the task group working on this—I said, look, you guys can do a smaller iteration on the existing standard, the third edition. I'm not particularly interested in working on that because we've kind of moved beyond that in Firefox, but if you want to do that, go for it.

[CONTINUED ON PAGE 53]



ScreamingMonkey lets Mozilla's Tamarin run in IE.

hoping that they'll come on-board at some point.

What we're trying to do in JavaScript 2 is mainly not about performance; it's more about, "Can I write my package or whatever you want to call it, throw it over the wall to you, and have you use it or mash it up with something Alice or Bob wrote and have them not step on each other?" And that's hard. That's the real goal with JavaScript 2, in my opinion.

But can't people do some of that today?

People do that with JavaScript today using conventions. They have the AJAX [Asynchronous JavaScript and XML] libraries share the global namespace by

EICH FROM PAGE 50

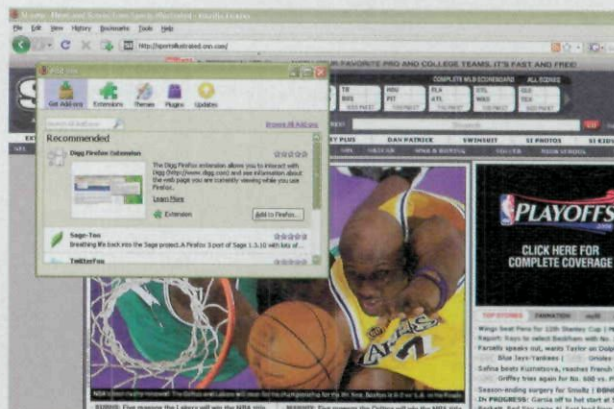
And so there's been an [ECMAScript] 3.1 effort so-called.

But we're actually on schedule for this year with the fourth edition [of ECMAScript], and there's an agreement that the fourth edition be strictly a superset, or the 3.1 version be a subset, of 4. This is so that there would be no divergence and we're not forking the language. And I have to say that's difficult because we've made this agreement. And it makes the language that's supposed to be the subset have the ability to add something that we don't want in the fourth edition. Then we have to have this negotiation and argue about it, and that's happening right now.

And sometimes I wonder, is Microsoft just trying to wrong-foot us? We're playing fair in this Ecma setting. We're trying to have covenants about the subset relation. And suddenly they're coming in and start changing their mind or adding things and start growing this thing on the side that we don't want. And we're going to get slowed down arguing with them. And that's a risk that we're taking.

On the other hand, if they do too much of that, they won't finish this year. And what's the point of having a 3.1 if the fourth edition is already done?

So ECMAScript 3—would that be equivalent to Java-



Eich is working on Firefox 3, due for release in mid-June.


Script 1.9?

It's even less. Like, they didn't want to go as far as we went in Firefox with some of the features we added in Firefox 2. I'll tell you straight: I think what they want to do is have "getters" and "setters"—these little metaprogramming hooks for making the property call a function on get and a different function on set. That's it. If they had a standard that said that, they'd ship that in IE 8. They don't have an IE 8 beta where they have "getters" and "setters" yet. And they've gotten a lot of flack for not having it.

You can see that that would be easy for them to add. Their heart's really in Silverlight. But it wouldn't hurt them a whole lot to add "getters" and "setters," and they probably will. They want a standard that allows them to wrap themselves in the mantle of [ECMAScript] 3.1. They want to fix some errata in 3.0; that's great. Guess where those errata were hosted

for the last nine years? Mozilla.org. Ecma didn't care that it had a bunch of bugs in its pseudocode spec; we did. Now they're suddenly saying we must spend a lot of effort putting these errata back into the smaller iteration of the standard.

[They want us to do this] among all the other work we could do that might be important, but it's less important than moving the Web forward. So we've compiled those errata.

So when you see the shoe on the other foot you have to wonder. And it's not a good environment. Before this big falling-out last year ... we had a high-quality committee. We had a good core group for a while, and it was collegial. Once we got into this 3.1 versus 4.0 thing it's been a little bit rough. And it makes me sad because we lost that old high-quality, small group of chefs cooking in the kitchen. Now it's more like food fights and cleavers being waved. 

EWEEK CORPORATE PARTNERS

eWeek's Corporate Partner Advisory Board comprises IT decision makers from government, manufacturing, education, retail, publishing, financial, utility, medical, telecom and nonprofit sectors. The Corporate Partners represent eWeek readers and work on an ongoing basis with eWeek editors, reporters and Labs analysts.

ADVISORY BOARD MEMBERS

Sam Inks	Aerjet
Francine M. Siconolfi	Aetna
Glenn Evans	AT&T
Kevin Levesque	ATX II LLC
Frank Calabrese	Bose
Kevin R. Baradet	Cornell University
Carl S. Ashkin	Darby Group
Randy Dugger	Dugger & Associates
Kevin Wilson	Duke Energy
David Bray	Emory University
Keith Carter	Estee Lauder
Ed Benincasa	FN Manufacturing
Tom Miller	VNUS Medical Technologies
Gary Gunnerson	Gannett
David Milkovich	Lockheed Martin
Bill Conati	Maax Spas Arizona
Karl Herleman	Miami Dade College
Robert Rosen	National Institute of Arthritis and Musculoskeletal and Skin Diseases
Erich Berman	Northwestern Mutual
Francis Rabuck	Rabuck Associates
Michael Skaff	San Francisco Symphony
Pankaj Chowdhry	Third Pillar
Michael Schwedhelm	United Labor Bank
Susan Nowicke	U.S. District Court
Bruce Brorson	University of Minnesota, Crookston
Judy Brown	University of Wisconsin System
Gregory Smith	World Wildlife Fund

Copyright of eWeek is the property of ZDNet and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.