

[topics](#)[using sendmail](#)[interviews](#)[articles](#)[conferences](#)[sites](#)[sendmail, inc.](#)our commercial
products and services[sendmail.org](#)the sendmail
consortium site[home](#) > [interviews](#) > tim o'reilly

Q&A: Tim O'Reilly

There may be places where Tim O'Reilly requires an introduction, but this isn't one of them. With a superlative publishing catalogue and a series of history-making conferences to his credit (and more to come), he and the rest of the team at O'Reilly & Associates have earned the good will of hackers of all stripes, from junior webmonkeys to Perl gurus. We were consequently delighted when Tim said he'd take part in a Q&A for sendmail.net.

O'Reilly & Associates is going in with [VA Linux](#) and [SGI](#) to offer a support and documentation package for [Debian Linux](#), and making the O'Reilly book on [Debian](#) available under the OPL. How much does the fact of Debian's "pure" Open Source distro figure into your decision to collaborate with them? What do you hope to achieve?

At bottom, this project was at VA's initiative, and they should get most of the credit for it. And one of our primary interests was in partnering with VA, which has done such a great job of supporting the ideals of the Linux community. However, I do also support the goal of the project, which is to make sure that the Debian project gets more visibility in what is shaping up to be the Linux distro wars. As you point out later in this email interview, Open Source projects can play a large role in maintaining open standards. And while so far all of the commercial Linux distributions have done a great job of continuing to support the free software ideals behind Linux, the Debian project is the only one to make that its explicit goal.

And because Debian is not itself out to create a company and supporting infrastructure, there was a fear that it

[talkback](#)[comp.mail.sendmail](#)newsgroup
(via Google)comments? ideas?
give us [feedback](#)[store](#)buy the [knife](#)buy the [t-shirt](#)

might get lost in the shuffle, known and loved by the hardcore Linux community, but hidden from the wider community that is now adopting Linux. So when VA suggested that it would be a good time to put together a packaged Linux distribution, we were glad to sign on. Obviously, at the price VA has set, this distribution is barely going to cover the costs of manufacturing and distribution, but that's consistent with the original vision of the [FSF](#) and the Debian project, which aims for availability and freedom, not for profits.

While a lot of the corporate talk about Open Source is undoubtedly vapor, there are companies that have shown a real commitment to the Open Source model and an understanding of its benefits. Can you give some examples of big companies that seem to get it? What do you think differentiates them? Executives with vision? Engineers with clout?

Clearly, among computer industry giants, IBM, HP, and SGI are leaders in this area, and Intel has made some extremely significant investments. All of these hardware companies have a clear strategic imperative in supporting Linux and other Open Source projects, though, in that it gives them a software platform free of what people have come to call "the Microsoft tax." It doesn't take a lot of vision to embrace a new supplier who gives you an enormous competitive advantage by lowering your costs! That being said, I do think that these companies do in fact have a lot of vision. I've been incredibly impressed, for instance, with the "tiger team" that IBM put together to explore the Open Source opportunity. These are brilliant, dedicated guys with a lot of industry experience, but with the freshness and openmindedness of hungry youngsters! This may be true at the other large companies as well; I've just spent more personal time with the IBM team.

As to how this tiger team got put together - you'd have to ask its members. My guess is that it's a combination of bottom-up and top-down, with a lot of give and go in both directions.

I also want to put in a good word for Sun. While they get

some amount of grief from the Open Source community because the SCSL is not a true Open Source license, I think that they are grappling with real issues that other companies may also have to face one day. As Bill Joy likes to point out, Microsoft has already tried to undermine Java, and with a pure Open Source license, they'd be free to do so. The right to fork, which is so key to Open Source developers, is a powerful tool for good or ill, and in the hands of a hostile company can possibly cause serious problems.

In any event, I see a lot of serious thought about these issues at Sun, and I applaud their courage in taking an unpopular position, and in trying something a little different. And as I like to point out, if Open Source is science and not religion, we're going to learn something from their experiment. Either they'll be able to spur collaborative development by the community of their users or they won't. Telling them that they're wrong before the experiment has been tried impoverishes us all. If they succeed, we'll know something about collaborative development that we don't now know; if they fail, we'll also learn something very important.

(I also do think that we'll be seeing Sun making some more explicit contributions to the Open Source community proper over the next year. They clearly realize that some of their code is not strategic for them, and would be very useful to others, and I'm hopeful that they'll do the right thing and contribute some significant pieces of code to the Linux community.)

I think that we're in a period of intense experimentation about licenses, as we try to figure out what causes the positive effects that we've all recognized from the Open Source model. I have no problem with experiments, as long as we all look honestly and clearly at the results.

Speaking of noble experiments, you have to give a huge amount of credit to Netscape. They've had a trial by fire with [Mozilla](#), and despite the naysayers, I think that the experiment is succeeding. What's important is that it's succeeding in unexpected ways, perhaps not the ones that

Netscape's management originally hoped, but significant nonetheless. And after all, isn't that one of the benefits that we tout about Open Source, that it produces valuable unintended consequences?

The strategic focus on Linux has been hugely successful in promoting the Open Source model to executives, analysts, the markets, and the press. At the same time, the pitch to the mainstream has largely overlooked the other technologies (such as Apache, BIND, sendmail, Perl) at the basis of the Internet/Open Source revolution. O'Reilly has done more than anyone I can think of to present Open Source as an ecosystem (rather than, say, a star system). Is it worth advancing that idea more forcefully beyond our own community? Or would that undermine the overall effort to promote Open Source?

I do think that this is an incredibly important message. Focusing on Linux is good in many ways, but it also blinds people to some very important things about Open Source.

Despite the excitement about [Red Hat](#), their success at putting software in a box and selling it only helps to perpetuate the myth that most software is written for sale. In fact, as most of us know, a great deal of software is written for use in our businesses. It's a tool, like any other, with a set of build-versus-buy tradeoffs.

I like to point to the Internet rather than Linux as the greatest triumph of Open Source for several reasons: the economic impact of Linux is still in the future, while the economic impact of the Internet is already obvious to everyone. But like anyone ignorant of history, most consumers don't give credit where credit is due. The absolutely mission-critical aspect of programs like Bind and sendmail and the ubiquity of Apache and Perl also undermine many of the common arguments used against Open Source.

I also like people to realize the absolutely seminal impact of Berkeley UNIX. While Richard Stallman certainly

makes a credible [claim](#) that the [GNU](#) project is one of the chief roots of Linux, and he was the first person to articulate the vision of a completely free operating system, I think that the university community, especially Berkeley UNIX and the MIT X Window system, with their more permissive licenses, have actually contributed more groundbreaking technologies. The Internet as we know it wouldn't exist without the BSD effort.

I also like to point out the parallels between the [IETF](#), the Internet standards effort, and the Open Source movement. Here is a long-running experiment in decentralized, collaborative decisionmaking, with technical merit the chief barrier to entry - much like Linux and other Open Source projects. Giving more visibility to other Open Source and Open Source-like projects expands our understanding of how collaborative development works in the age of the Internet. And as Larry Wall likes to say, "There's more than one way to do it."

You've talked about the battles beyond the Linux-Windows competition - the battle to keep Internet software open and non-proprietary, the danger of a new proprietary layer of "infoware" built on top of the Web, and the need to plan for the next generation of applications. But so far (with certain exceptions) the Open Source model seems to deal better with working models than with long-term roadmaps. Given the decentralization of the Open Source milieu, how can we develop a technology (not marketing) strategy that's less tied to short-term tactical needs? Of course, this may be the previous question in another guise.

Yes, in some ways it is, but the answer is forward-looking rather than historical and backward-looking. For a long time, I celebrated the fact that many of the world's leading Web sites were building new kinds of applications on an Open Source foundation. I coined the term "infoware" to capture the kinds of applications that are typically built with html+scripting rather than with traditional higher-level languages.

But then it occurred to me that even though some of these

companies have participated actively in Open Source projects, they didn't necessarily think of themselves as Open Source companies with an *imperative* to contribute back. What's more, because the applications they build are hosted applications, not even a viral license like the [GPL](#) actually requires them to contribute, because their software need not actually be redistributed. It's for this reason that I've started urging Web companies that have benefited from Apache, Perl, and other Open Source tools to provide more explicit support back to the communities that have given them such a leg up. This might be in the form of technical contributions, but I'd also love to see some of these huge-market-cap companies fund some kind of high-profile awards or grants (much like the MacArthur fellowships) to recognize and spark original work in the field.

- I have to say, though, that the fact that the GPL doesn't really compel these companies to contribute may actually be a good thing, because it puts the shoe where it belongs. I want people to contribute because they see the benefits, not because someone tries to compel them to do so.

What are those benefits? Some of them are direct, in the form of increased recognition by the technical community (which may make it easier to hire qualified people), not to mention helping to train more qualified people! (Would you rather hire a paper MCSE or someone who's demonstrated their expertise by making a notable contribution to an Open Source project?) Or a company might find a community of interest with other non-competitive companies that use similar software, and spread the costs of improving the software by collaborating rather than competing. Or the benefits might be indirect, in that giving back simply enriches the computer industry ecosystem.

I like to say that for many Open Source projects, the return on investment is the solution to your own problem. If you give that solution away to others, you may reap unexpected dividends, as they improve on your work, or give you solutions to other problems you might not yet have solved yourself.

Eric Allman has remarked that working on the bat book with Bryan Costales helped push him to make some key improvements to sendmail. This sort of active (you might say dialectical) exchange between author and coder goes well beyond the usual idea of documentation. Does this sort of thing happen a lot at O'Reilly?

Well, that was probably the clearest case, where Eric literally started working actively on sendmail again after a bit of a hiatus, because he said it was easier to fix some of the inconsistencies that Bryan turned up than to explain them!

But actually O'Reilly has had an influence on a number of technology projects. For example, we discovered the Web very early on (through an effort to put our X books online, no less), and helped put it on the radar far earlier than might have happened otherwise. When we first covered it in *The Whole Internet User's Guide and Catalog* in 1992, there were only a few hundred Web sites - hardly enough to get the attention of other publishers, who tended to cover technologies only once they were well established and hyped by the industry. In an attempt to put the catalog portion of the book online, we created the world's first Web portal, GNN, the Global Network Navigator. (Incidentally, GNN was the first Web site to carry advertising, in early 1993, setting the stage for an entire industry.) In fact, one of the Mosaic developers claimed in a *Spy* magazine interview that the NCSA team first heard of the Web from an O'Reilly direct mail piece. (Whether it was for *The Whole Internet User's Guide and Catalog* or for GNN, I don't know.) And we sponsored the first-ever World Wide Web Wizard's Workshop. Tim Berners-Lee talks about some of our involvement in his new book. Dale Dougherty, who has led most of our online efforts since the late '80s, was a key player in the early Web.

Dale also started the Davenport Group, which created the Docbook DTD, which has since been adopted by the Linux Documentation Project. It was a seminal effort that is just starting to bear fruit today now that XML is supplanting SGML as an extensible markup language.

In addition, through our work with Larry Wall on Perl, I've had a small influence on some new features. While we generally just tell Larry to do whatever interests him, occasionally I see something that seems important. So for example, I asked him to put XML support into Perl, an effort that resulted in the `xml::parser` module now maintained by Clark Cooper. And of course, through our investment in ActiveState and active project management by our software director Gina Blaber, we instigated the OnePerl effort that re-unified the UNIX and Windows ports of Perl.

I also like to think that many of our books on UNIX utilities helped make those utilities more accessible, and thus played an enormous role in the spread of UNIX (and ultimately Linux). After all, where would we be if the only documentation on `sed` remained the original paper showing how it was used to perform transforms on Coleridge's "Kubla Khan"? We've had countless customers who've told us that they've gotten jobs, or developed software programs that eventually came to be part of the UNIX/Linux oeuvre, because of the start they got with O'Reilly books. When we started doing these books, it wasn't clear that they were going to be big moneymakers! In fact, our first print run for many of our early books was only a few hundred copies. We wrote them because we thought there was a need for them, not from some grand plan to build the business they've turned into. We were a consulting business and wrote many of our early books in between consulting jobs, as a way of doing some good while our people didn't have paying work.

But I'm getting off the point. I do think that there's an incredible dialectic between authors of books and authors of software, and it's great when they can work together. That's one of the reasons we like working on Open Source related books: we *can* work with the original developers and knowledgeable users of the product, both to make the software better and to create documentation that really extends the power of the software.

You've spoken about how Open Source software gives

weight to open standards (Apache and HTTP, sendmail and SMTP) and keeps them from becoming pawns in proprietary skirmishes. Do you see standards that are in danger of being "taken private" in that way?

Obviously, HTML became a pawn between Microsoft and Netscape. XML is likely to be a real battleground, with companies claiming "openness" but trying to gain the high ground from which they can crush their competitors. But frankly, there are battlegrounds everywhere. I'd say to companies or independent developers that for any protocol or data interchange standard whose openness and standardization is important to you, you should work to support an Open Source implementation. It's a great way to keep the big companies honest.

Software engineering doesn't come cheap, but funding rigorous Open Source development can be tricky. What funding models do you see that successfully combine engineering methodology with a commitment to Open Source? Are instances like the [Apache Foundation](#) and [sourceXchange](#) signs that Open Source projects will find more sustainable models, in terms of both revenue and methodology?

I love the Apache model, where a group of users got together to combine their efforts to improve the NCSA server. "A patchy server" became Apache because it mattered to enough users that they found ways to collaborate rather than compete. They had individual revenue streams that rewarded their participation; Apache was a means, not an end.

I do think that we're going to see a lot more projects funded using that model. There are an awful lot of software areas where the benefits to large end users of cooperation trump the benefits to vendors of competition. What Apache teaches us is that you don't need a single visionary leader, you need a well managed process for collaboration. And really, that's what we're trying to do with Collab.Net, the company that produces sourceXchange. SourceXchange is really just the first

"product" of the company - the long-term vision is to provide the infrastructure that will allow companies - not just leading-edge hacker communities - to experience the benefits of collaborative internet-enabled development.

What's new from O'Reilly - books, conferences, whatever - that you'd especially like people to pay attention to?

Well, I'd like everyone to know about the new hardcover edition of Eric Raymond's essays, *The Cathedral and the Bazaar*, complete with cover quotes from Guy Kawasaki, Tom Peters, Clayton Christenson, and other names familiar to Fortune 500 executives. While many of you will have read the essays online, I'd love the help of the hacker community in putting this book on the *New York Times* bestseller list, since if we can get it there, we'll be spreading the message to a whole new group. The essays in the book have been updated, and it's a really handsome edition, if I do say so myself, well worth the \$19.95 cover price.

Another book I've recently published that I feel very strongly about is Jon Udell's *Practical Internet Groupware*. This isn't an obvious buzzword with a built-in customer base to rush out and buy it (unlike the upcoming third edition of *sendmail*, for instance!), but it's a really important book. So check it out!

Our upcoming books on Samba will also be of interest to Open Source readers.

The Comdex show in Las Vegas in November will be offering a Linux Business Expo and the first ever [O'Reilly Open Source Software Development Forum](#), including three two-day intensive tutorials on Perl, Apache, and Linux.

And besides all that, we'll be back again with our Open Source Convention next July 17-20 in Monterey. This is really becoming *the* event where the Open Source community comes together to evangelize and collaborate. Next year we'll structure the event to feature an even

broader range of Open Source technologies and more opportunities for cross-community interaction.

One last question: At what point did you realize that you weren't just writing great documentation, that you were actually making (and writing) history as well?

Gosh, that's kind of a strong statement. I guess when other people started saying it :-). But truly, you can't believe all the press. Just as you have to discount the negative stuff, you have to discount the positive stuff as well. Clearly the work we did with GNN was a significant piece of computer industry history, and I suppose the Open Source summit and some of the following advocacy has been as well. But I remember Bob Schiefler, the onetime head of the X Consortium and the architect of the X protocol, remarking on how fleeting fame was. "Most of the people who use X don't even know who I am, and that's as it should be," he said. While I think that anyone *in the computer industry* (and especially in the Open Source community) who doesn't know Bob and his accomplishments really ought to, I have to agree that in the long term most of us will be forgotten, even as the creations we've set free have changed the world.

Somehow that reminds me of a Rilke quote that my friends now make fun of because they hear it from me so much. In his poem "The Man Watching," he describes Jacob's Biblical wrestling match with an angel, and concludes with something like this: "What we fight with is so small, and when we win, it makes us small. What we want is to be defeated, decisively, by successively greater things."

[home](#)[search](#)[about us](#)[site map](#)[policies](#)