

# THE PERL REVIEW'S COMMENTARY

by Nadim Khemir  
Guest  
Commentator



## You are Perl's best advertising

The road is long. Developing with Perl feels like beating the tracks of a remote region of Sulawesi. You're mostly going to meet nice people but you're still going to curse and cry and mosquitoes will suck you dry after they've made you crazy.

"Why Perl?". I think I'll lose my sanity if I hear that question one more time. Every time the word "Perl" pops up, the body's core temperatures go up, fingers point everywhere, some critical mass is closed to being reached and overload is imminent. I can avoid all this by keeping clear from language zealots, incompetent people and just meet Perl geeks like yourself; with the risk of reducing your social life to IRC chatting and mongers meetings.

I think most Perl developers know at least one or two other programming languages. That makes Perl developers a threat because they are knowledgeable. Taking in consideration experience, the average age for a Perl developer must be around 40; we simply qualify for the `nerd-knows_what_a_linker_is-has_no_life-runs_a_cluster_at_home-owns_two_amigas-geek`.

Those are dangerous people because they can not only make things work they can even make them work efficiently, quickly, and cheaper than other without any micro-management overhead or latest smoke-and-mirrors methodology.

Have you noticed that all the hyped development methodologies are more about management than development? This may sounds surprising coming from a manager, but not all managers are alike and some even know what they are talking about (which they achieve by listening).

"Why Perl?" is a question I had to seriously answer at least five times in the last 6 months. That would be okay if it wasn't for exactly the same project that is replacing another, successful, Perl project.

The world runs on Perl but no one knows it except when things go bad. And then it's never the sub-optimal conditions, the non-existent budget, the 50% time spent in meetings or the lack of interest that is the source of the failure. It's Perl and all those line-noise-writing weirdos (you can't be normal and use Perl it seems). It doesn't matter if you put up a solution for a third of the price and didn't get a decent raise for two years.

When did you last hear that a java project went bad? How often do you hear a Perl project goes well? At YAPC:EU 2007 I tried to explain what I believe every Perl developer should do to make Perl mainstream (and mainstream is not when you use it everywhere to save the world again and nobody notices, so 'no', Perl is not mainstream. Perl's just invaluable, which is bad enough in itself).

I started asking myself "Why people don't know how good Perl is?". Is it the fault of its gentle-mannered creator? Is it the fault of his low profile followers? Is it the fault of The Perl Foundation? Is it my fault (arff, that can't be I'm just a nobody,

we're all just simple people right?)?

I don't know, but things are getting seriously difficult in the real world, which is just outside the cave where you retire to do line noise incantations. The real world is the one paying your salary. I just want to be able to develop in Perl in five years without having to be compared to a COBOL developer.

Sure, I want to have a rich development ecosystem, I welcomed java even though I thought it crippled C++ and lobotomized developers, but after consideration it did fill a need: application development by almost anyone who couldn't handle a pointer properly. But hold on, we could already do that.

I also welcomed Python. It did even do something for me that I couldn't do easily with Perl: embedding easily in an application. Right, that can be done with Perl without cursing too much these days.

I still haven't understand the use of PHP but that doesn't matter because other fellow programmers think it's good and it does the work they expect it to do. So I'm happy for them although I recommend that they get a grip. Perl 5 is not bad, not at all.

The Perl community has a great plan to fix all the flaws that the current Perl has. It's called Perl 6. I'm sure Perl 6 will be great because our community is a great one but I very seriously doubt it's going to be any good if it is used in caves.

I still haven't found the answer to "why?" I probably should go fix the problems I have in my modules instead for writing this philosophical article or open a window (Sweden is a country where caves have windows) and profit of the last days of summer.

I've got it!

Like all other Perl developers I like development so much that I never take time to advertise Perl. Perl is the duct tape of the Internet, I read. That's not completely right: it's the duct tape of the world. My company is asking me to explain why we should use Perl (which is an excellent question) instead for another language for which good developers are easier to find (and I thought good developers have nothing to do with the language they use). Maybe I should ask the company why 40% of the developers know a little bit of Perl and use it in their daily work (some write horribly crappy Perl though) and they are never send them to any Perl education or even a cool Perl conference.

The reason why Perl does not have the consideration it deserves is extremely simple. We're cheap, cheap like duct tape. Perl costs nothing. Educating Perl developers costs nothing (they do it for fun), The license doesn't oblige anyone to pay astronomic fees because they simply download the SDK and happened to put in in you product. Nobody sells companies expensive libraries or modules because they are all free with

a great distribution environment. No big corporation stands behind Perl (Microsoft did pay for porting Perl to Win32, right?), no flashy web sites, no expensive conference, no certification programs, no "our lawyer will contact you". Yeah, we're definitely cheap.

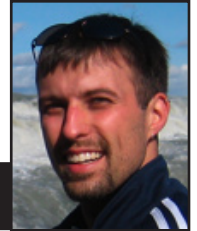
Take a product, change the box, multiply the price by five and it's suddenly a much better product. Ever bought stainless steel screws in a boat shop?.

The real world is, well, strange. It wants packaging, commercial support, certification, white papers, a list great project success (and excellent hiding mechanism of the bad ones), methodologies and reassurance that they will find developers five years in the

future. A language that is taught at university and schools not something you rapidly pick up to fix the flaws of the languages that you learned.

If you believe we have all that, then brother write it down and give the rest of the community the URL so we can finish the work together. Now I'll go and heal my end-of-summer blues and prepare for a great time at YAPC with a nice song. The Hollies - The Road is Long, he's my brother - 1969.

by Joshua McAdams,  
TPF Public Relations



## The Perl Foundation Summer Report

The Perl Foundation has been busy in the past few months. We received a record amount of contributions, handed out thousands of dollars in grants, aided numerous conferences, workshops, and hack-a-thons, and honored hard-working members of the Perl community.

In May, The Perl Foundation received a philanthropic donation of \$200,000 from Ian Hague. Mr. Hague is a co-founder of Firebird Management LLC, a financial fund management company based in New York City. This donation was the result of extensive discussions between Mr. Hague, The Perl Foundation, and a Perl community member who wishes to remain anonymous.

Of this extremely generous donation, we'll use roughly half of the money to fund Perl 6 related projects. These projects will advance the release date of a Perl 6 implementation. We'll use the other half to develop the capabilities of The Perl Foundation and to help us in our fundraising efforts.

These earmarked funds have already been put to use with the granting of \$2,500 to Daniel Ruoso for his "Simple Meta Object Programming" project to deliver an alternative implementation of Perl 6.

Other grants that we've recently funded include "Test::Builder 2" and "Make localtime() and gmtime() Work Past 2038" by Michael Schwern, "Fixing Bugs in the Archive::Zip Perl Module" by Alan Haggai Alavi, and the exciting "Perl on a Stick" project by Adam Kennedy. We've been posting all grant proposals, as well as biweekly grant updates, to our news blog (<http://news.perlfoundation.org>).

"Perl on a Stick" is a Windows compilation of Perl packaged with crucial modules and intended to be executed from removable media on any modern Microsoft Windows system. We featured at O'Reilly's Open Source Conference and at YAPC::EU 2008 by handing out thumb drives with an alpha version of Portable Strawberry Perl.

The Perl Foundation is also wrapping up its involvement with Google's Summer of Code. Five of our projects accepted by Google are complete and their results have already made an effect in the Perl world.

"Flesh out the Perl 6 Test Suite" was one of the successful projects. This project was developed by Adrian Kreher and mentored by Moritz Lenz and added 700 new tests to the Perl 6 test suite.

Another successful project was "wxCPANPLUS" developed by Samuel Tyler and mentored by Herbert Breunung and Jos Boumans. This project aimed to create a graphical user interface for the CPANPLUS Perl module, written in wxPerl, including many of the functions available on the CPAN web site and Randy Kobe's Search (<http://kobesearch.cpan.org/>).

The third successful project was "Native Call Interface Signatures and Stubs Generation for Parrot" developed by Kevin Tew and mentored by Jerry Gay. This project created NCI signatures and stubs generation using Parrot's c99 parser.

The fourth project was the "Incremental Tricolor Garbage Collector" developed by Andrew Whitworth and mentored by chromatic. This project replaces the mark-and-sweep garbage collector in Parrot with a new and more efficient one.

The final project was the Math::GSL project developed by Thierry Moisan and mentored by Jonathan Leto that created a Perl interface to the GNU Scientific Library. It's on CPAN (<http://search.cpan.org/dist/Math-GSL>).

Congratulations to all of the students and thanks to all of the mentors and to Google for making the Summer of Code a success once again. The Perl Foundation is already preparing for next year in order to make the program easier for students and even more beneficial to the Perl community.

Perl had a huge presence at O'Reilly's OSCON 2008. As usual, O'Reilly were great hosts and provided the Perl community with ample speaking slots and with additional time and space for Larry Wall to deliver the State of the Onion address and for The Perl Foundation to award the White Camel awards.

The White Camel Awards, now in their tenth year, honors those who make noteworthy and ongoing non-technical contributions in benefit of the Perl community. This year's winners are Jacinta Richardson for her tireless care and feeding of the Australian Perl community as well as organizing OSDC::AU, Linux.conf.au, and SAGE AU; Tatsuhiko Miyagawa for his work organizing several Perl events in Asia (including three YAPCs); and Gabor Szabo for organizing several events, including the Hungarian Perl Workshop, YAPC::Israel, and OSDC::Israel. Congratulations to all of the White Camel winners. Read about previous winners at [http://www.perl.org/advocacy/white\\_camel/](http://www.perl.org/advocacy/white_camel/).

The next few months look to be an exciting time for The Perl Foundation. Keep up with all that is happening at <http://news.perlfoundation.org>.