

# Google's Summer of Code

by Eric Wilhelm  
ewilhelm@cpan.org



Each summer, Google sponsors a “Summer of Code” where it pays \$5,000 per project to support students to work on open source projects over their summer break. Each student is mentored by a project representative.

The Google Summer of Code 2008 enters the “And now we write code” phase on May 26, with six students working under The Perl Foundation banner this year. I’m very pleased with the quality of proposals and we’re all looking forward to a great summer.

This year, I organized the mentors and applications for Perl projects through The Perl Foundation. I’ll tell you about the accepted projects and give you some background on the organization and application processes.

## ■ Projects for 2008 -----

The Perl project abstracts are listed online with all of the other SoC projects (<http://code.google.com/soc/2008/perl/about.html>). We’ll put all of the projects into an aggregator for the students to post updates as they work, but we haven’t done that yet. Here are short summaries of this summer’s Perl projects—the students wrote excellent proposals and I think we might post the full texts soon because the abstracts don’t do them justice:

### Flesh out the Perl 6 Test Suite

Student: Adrian Kreher  
Mentors: Moritz Lenz, Jerry Gay, and Will Coleda

Adrian Kreher will be transforming the Perl 6 Synopses into running code. Currently, the Perl 6 test suite for the language specification is in the Pugs repository and the test results are automatically merged into the Synopses (<http://perlcabal.org/syn/>). Adrian will identify key areas not already checked, especially edge cases, and write tests for them. He plans to make it through one synopsis per week.

### wxCpanPLUS

Student: Samuel Tyler  
Mentors: Herbert Breunung, Jos Boumans, and Steffen Müller

Samuel Tyler is going to create a wxPerl GUI for CPANPLUS, the tool that interacts with the Comprehensive Perl Archive Network, mostly for inspecting and installing Perl modules. Currently CPANPLUS either operates from the command line through the `cpanp` tool or through other automatic, non-

interactive tools. The project will have a YUM Extender-like interface (<http://www.yum-extender.org>), the popular GUI installer for yum package manager.

### Native Call Interface Signatures and Stubs Generation for Parrot

Student: Kevin Tew  
Mentors: Jerry Gay, chromatic, and Will Coleda

Kevin Tew will be making it easier to use NCI (e.g. at runtime) for all parrot-hosted languages. This includes building a tool to parse C header files and then generate the `dlfunc` signatures and stubs to interface with the external C libraries. With this tool, Parrot-hosted languages would easily be able to use important libraries such as GSL, graphviz, and Expat.

### Incremental Tricolor Garbage Collector

Student: Andrew Whitworth  
Mentors: chromatic, Jerry Gay, and Will Coleda

Andrew Whitworth will be tackling the task of creating a Parrot garbage collector which doesn’t need to stop the world to take out the trash. The current Parrot garbage collector uses “mark and sweep”, so Parrot must inspect all objects when it decides to clean-up. While inspecting all objects, a Parrot program has to temporarily pause the useful work. A tri-color garbage collector works incrementally. “White” objects have not been checked, “grey” objects have been checked at the container level but their contents have not been checked, and “black” objects have been checked completely. In each pass, white objects turn to grey objects, grey to black objects, and black objects can be garbage collected. Breaking it into smaller tasks rather than a single pass should spread out the GC time and allow Parrot to keep more threads engaged in useful work.

### Create Math::GSL

Student: Thierry Moisan  
Mentors: Jonathan Leto, Alberto Simões, and Steffen Müller

Thierry Moisan will be writing, documenting, and testing a SWIG-based binding to the GNU Scientific Library (<http://www.gnu.org/software/gsl/>), which has over 1,000 functions to handle advanced and applied mathematics such as linear algebra, root-finding, and fast Fourier transforms. Thierry will create