

# Ruby 1.9.1 RC Released

**Finnbarr P. Murphy**

([fpm@fpmurphy.com](mailto:fpm@fpmurphy.com))

In his usual understated way, [Yuhi Sonoda](#) announced the first Ruby 1.9.1 release candidate earlier today on the [ruby-talk](#) mailing list.

Hi, folks

Ruby 1.9.1 has been just released.

This is a release candidate of Ruby 1.9.1, which will be the first stable version of Ruby 1.9 series. Try it early and have a experience of modern, faster, with clearer syntax, multilingualized and much improved Ruby world.

We have fixed 72 bugs and implemented some features:  
[http://redmine.ruby-lang.org/projects/ruby-19/issues?query\\_id=9](http://redmine.ruby-lang.org/projects/ruby-19/issues?query_id=9)

If you encounter a bug or a problem, please let us know it via the official issue tracking system (<http://redmine.ruby-lang.org> ).

I am a fan of Ruby and look forward to every release. My first introduction to Ruby was way back in early days of 1997 just after Ruby 1.0 was released by [Yukihiro Matsumoto](#) (aka Matz) whose blog, or as the Japanese prefer to term it *web diary* is [here](#). Since then I have used it from time to time for various small scripts in the same why that I also use Perl. I recommend it to people when asked what programming language they should learn first. My only reservations are that traditionally Ruby tends to be somewhat slow compared to other popular scripting languages and that there now any number of Ruby implementations which leads to some portability issues. Performance has improved with every release and there is no reason nowadays not to use Ruby in production environments.

This release is significant in that it is the first release candidate of Ruby 1.9.1 which is targeted to be the first stable production-grade version of the Ruby 1.9 (aka Yaru) series. Ruby 1.9 is intended to be a major evolution of the Ruby language akin to Python 3.0 or Perl 6. Old crud and language features that were, in hindsight, poorly designed have been removed. A number of new language features are implemented. If you want more information about the changes in Ruby 1.9, a good reference to start with is [Mauricio Fernandez's Changes in Ruby 1.9](#). With this release, the language features are frozen except for multilingual support is some of the standard libraries which come with Ruby. I have played with beta releases of Ruby 1.9 for some time now and am pleased with the language changes and especially the improved threading model.

As I said previously, there is not one single implementation of the Ruby language. The implementation released today is regarded as the official Ruby language implementation in that it is maintained by Matz, Yuhi and a number of trusted lieutenants. It is sometimes referred to as *CRuby* since the underlying language is C.

If you are unfamiliar with the different Ruby implementations other than *CRuby*, the leading contenders, in no particular order, are [Ruby Enterprise Edition](#), [JRuby](#), [Rubinius](#), [MagLev](#) and finally [Microsoft's IronRuby](#) which interestingly they are developing in a semi-open model. You can actually contribute code and examine the source code! For a comparison of the performance of the various implementations, have a look at [Antonio Cangiano's Great Ruby Shootout 2008](#). To

interpret these results correctly, you need to read his followup post [Reflections on the Ruby Shootout](#).

I still have some reservations about the speed of code execution in Ruby. While Ruby 1.9.1 is around twice as fast as Ruby 1.8.7, Ruby is still quite slow compared to other scripting languages such as Python or Perl even if it is somewhat faster than PHP. Work still needs to be done to improve performance.

If you want to compare the performance of Ruby to other languages for various benchmarks, one place you can do so is at Debian's [Alioth](#) which claims to be the successor to Doug Bagley's *The Great Computer Language Shootout* amongst other things. Other benchmarks worth looking at are [here](#) and [here](#). However, keep in mind the golden rule of benchmarking, to wit - all benchmarks are flawed benchmarks.

You can download the source code for this release at [ruby-lang.org](#). If you find the [ruby-lang.org](#) FTP server is too slow, Peter Cooper has made the tarball available on [Amazon S3](#) which should result in a much faster download.

For some reason, new releases of CRuby always seem to occur in December. Anybody know why?