GMOペパボ

How to Begin to Develop Ruby Core

The secrets of CRuby ecosystem

Hiroshi SHIBATA / GMO PEPABO inc.

2016.09.24 Euruko 2016

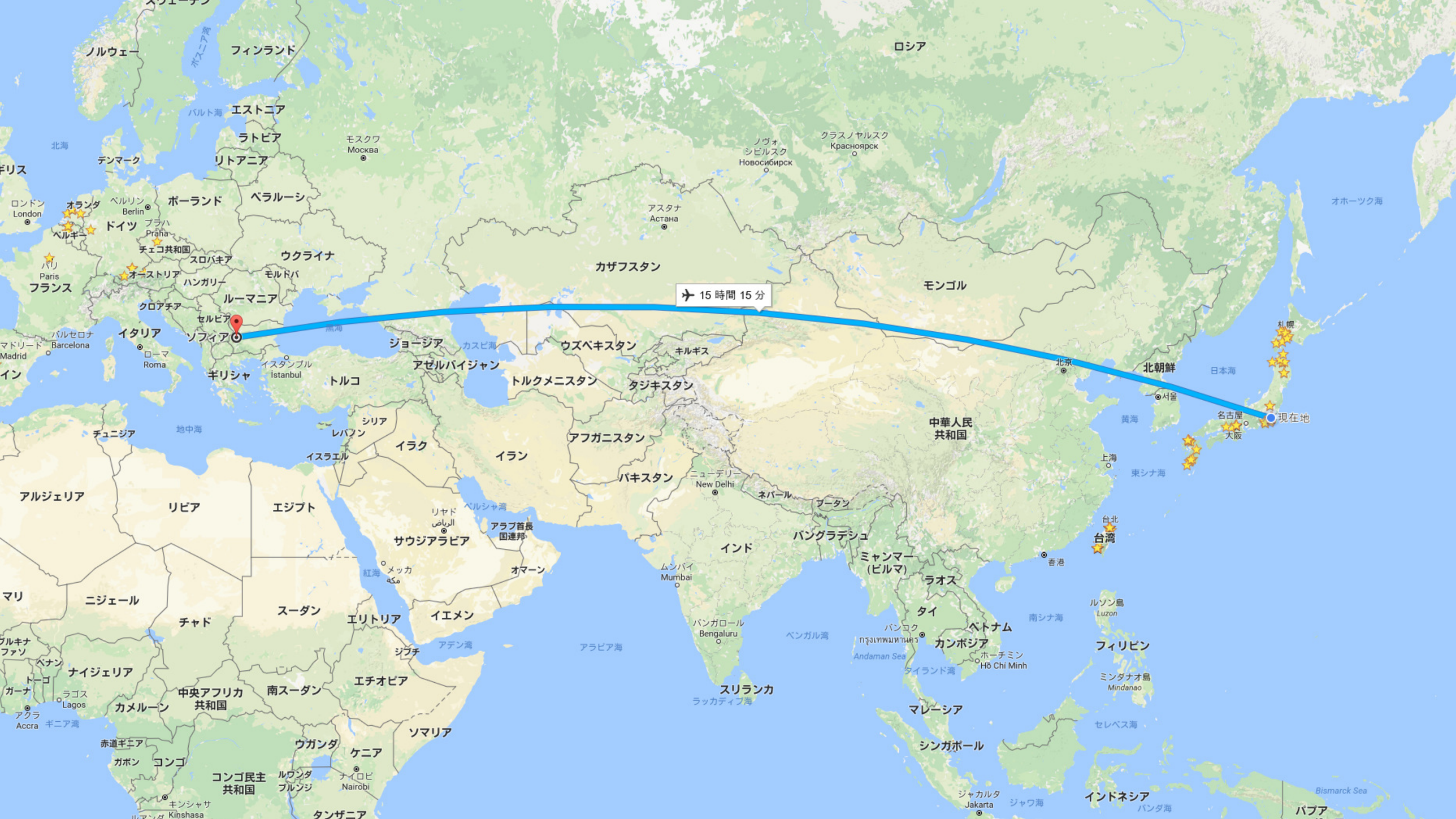
GMOペパボ



Hiroshi SHIBATA @hsbt

Chief Engineer

https://www.hsbt.org



15 時間 15 分
ソフィア
ルーマニア
ブルガリア
ギリシャ
イタリア
フランス
ドイツ
ポーランド
ベラルーシ
ウクライナ
モルドバ
ハンガリー
スロバキア
チェコ共和国
オーストリア
クロアチア
セルビア
ロンドン
London
オランダ
ベルギー
デンマーク
ベルリン
Berlin
プラハ
Praha
バリ
Paris
バルセロナ
Barcelona
マドリード
Madrid
ローマ
Roma
イスタンブル
İstanbul
トルコ
ジョージア
アゼルバイジャン
カスピ海
黒海
ウズベキスタン
トルクメニスタン
キルギス
タジキスタン
アフガニスタン
パキスタン
イラン
イラク
シリア
レバノン
イスラエル
チュニジア
地中海
アルジェリア
リビア
エジプト
サウジアラビア
リヤド
الرياض
ペルシャ湾
アラブ首長
国連邦
オマーン
紅海
メッカ
مكة
イエメン
エリトリア
スーダン
チャド
ニジェール
マリ
ブルキナファソ
ベナン
トーゴ
ガーナ
ナイジェリア
ラゴス
Lagos
アクラ
Accra
ギニア湾
カメルーン
中央アフリカ
共和国
南スーダン
エチオピア
ジブチ
アデン湾
ソマリア
ウガンダ
ケニア
ナイロビ
Nairobi
ルワンダ
ブルンジ
コンゴ民主
共和国
コンゴ
赤道ギニア
ガボン
キンシャサ
Kinshasa
タンザニア
アラビア海
スリランカ
ラッカディブ海
インド
ムンバイ
Mumbai
バンガロール
Bengaluru
ベンガル湾
ニューデリー
New Delhi
ネパール
ブータン
バングラデシュ
ミャンマー
(ビルマ)
ラオス
タイ
バンコク
กรุงเทพมหานคร
ベトナム
カンボジア
ホーチミン
Hồ Chí Minh
タイランド湾
Andaman Sea
マレーシア
シンガポール
ジャカルタ
Jakarta
ジャワ海
インドネシア
バンダ海
パプア
Bismarck Sea
ルソン島
Luzon
フィリピン
ミンダナオ島
Mindanao
セレベス海
南シナ海
香港
台北
台湾
東シナ海
上海
黄海
北朝鮮
北京
서울
日本海
札幌
名古屋
大阪
現在地
オホーツク海
モンゴル
カザフスタン
アスタナ
Астана
ロシア
ノヴォシビルスク
Новосибирск
クラスノヤルスク
Красноярск
モスクワ
Москва
ノルウェー
フィンランド
ボスニア湾
バルト海
エストニア
ラトビア
リトアニア
北海
ギリス



RubyKaigi2016

at Kyoto International Conference Center, Kyoto, Japan
Sep. 8th - 10th

› Registration › Call for Sponsors

› Anti Harassment Policy › Team

For news and updates, follow @rubykaigi on Twitter

self.introduce

```
=>
{
  name: “SHIBATA Hiroshi”,
  nickname: “hsbt”,
  title: “Chief engineer at GMO Pepabo, Inc.”,
  commit_bits: [“ruby”, “rake”, “rubygems”, “rdoc”, “psych”, “syck”, “ruby-build”, “railsgirls”, “railsgirls-jp”, ...],
  sites: [“hsbt.org”, “ruby-lang.org”, “rubyci.org”, “railsgirls.com”, “railsgirls.jp”],
}
```





CEO(@kentarow) and CTO(@kentaro) support me

https://pepabo.com

1.

What’s (C)Ruby?

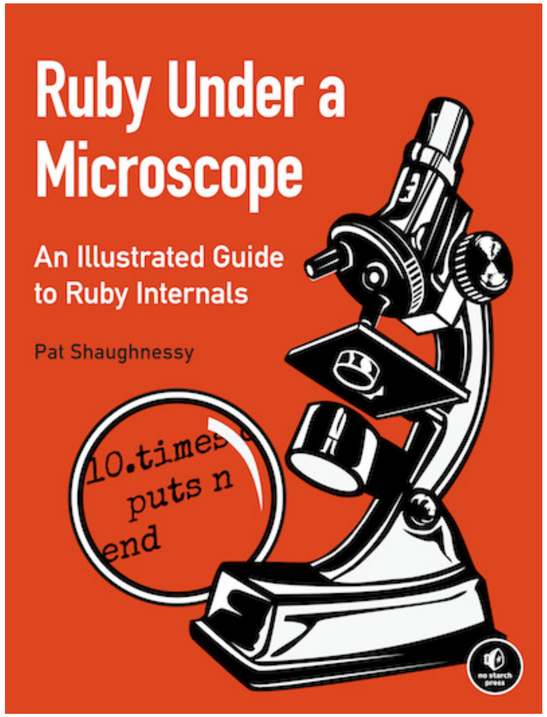
Variation of Ruby Interpreter

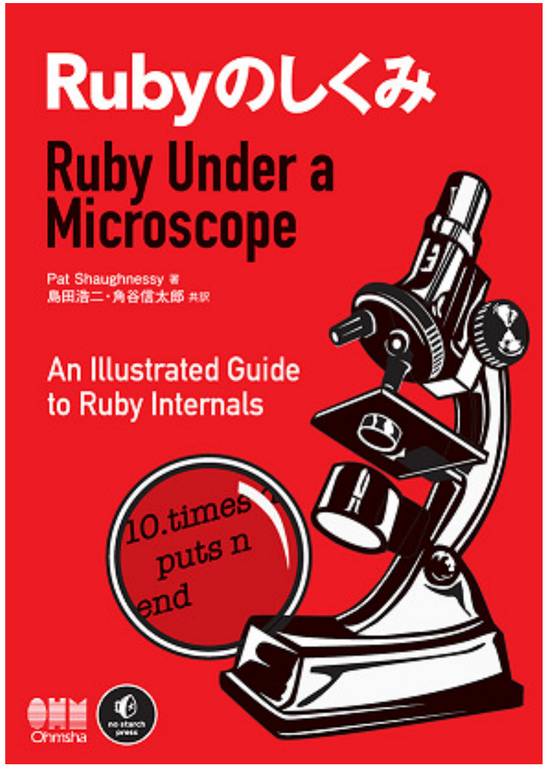
Implementation:

- Ruby(MRI, CRuby)
- JRuby
- Rubinius
- Opal

ISO/IEC 30170:2012:

- CRuby
- mruby





Basis of CRuby and YARV

“ Throughout most of this book we’ll learn about the original, standard implementation of Ruby, known as Matz’s Ruby Interpreter (MRI) after Yukihiro Matsumoto, who invented Ruby in 1993.”

Ruby 1.8

Ruby Under a Microscope, p.4

“ With Ruby 1.9, Koichi Sasada and the Ruby core team introduced Yet Another Ruby Virtual Machine (YARV), which actually executes your Ruby code.”

Ruby 1.9

Ruby Under a Microscope, p.33

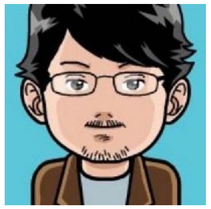


Ruby Committer

Full-time commiters

Heroku

- Matz
- ko1
- nOkada





MoneyForward

- shyouhei



Total 91 people + 2 bot

```
$ cat ~svn/.ssh/authorized_keys | awk '{print $5}' | sort | uniq | wc -l
93
```



Branch maintainers

trunk known as 2.4

nurse: Release manager



2.3

nagachika: Stable branch maintainer



2.2, 2.1

unak: Old stable branch maintainer





Platform maintainers

Linux



- akr
- normalperson
- n0kada

and others

Windows



- unak
- n0kada

BSD



- nurse

Solaris



- ngoto

OS X



- n0kada
- mrkn

Ruby Core Maintenance scope policy

Language core features including security
  Yukihiro Matsumoto (matz)

Ruby VM(YARV)
  Koichi Sasada (ko1)

Core classes
  Ruby Commiters



Ruby Core Maintenance Policy

Standard Libraries

- Each maintainers

Bundled Libraries(default/bundled gems)

- Hiroshi SHIBATA(hsbt), Nobuyoshi Nakada(nobu)

Documentation

- Zachary Scott(zzak)



*.ruby-lang.org

www.ruby-lang.org

Official Website of Ruby language. It's hosted by Heroku.
We welcomed to translate contribution.
see http://github.com/ruby/www.ruby-lang.org

svn.ruby-lang.org

Main repository of Ruby source.

bugs.ruby-lang.org

Official issue tracker of Ruby. It's hosted by Heroku

*.ruby-lang.org

ftp.ruby-lang.org

Site of distribute official package

docs.ruby-lang.org

Hosted document generated from RDoc and rubima project.



CDN

Our site and package distribution were supported by fastly.

- www.ruby-lang.org
- cache.ruby-lang.org

Statistics of our CDN:

- Access ratio: USA: 37.9%, EU: 17.8%, Asia: 39.0%, Others: 5.3%
- Bandwidth: 6181 GB/month
- Requests: 12,296,848 req/month = 4 req/sec



What does mean “official”?

“official” means “Matz controllable”

Un-controllable examples:

- ruby-doc.org
- rubygems.org
- bundler.io
- rvm/rbenv/chruby



Licenses of Ruby language

- 2-clause BSDL
- Ruby License

see details of `COPYING` file in ruby source code.



Ruby is OSS

2.

Do submit your patch to OSS

Contributing of OSS

People say:
“Contributing to OSS is easy! Please write some documentation and submit a patch!”

You say:
“Okay! I will contribute new documentation for Ruby!”



Documentation is hard

- No-one knows the true behavior of OSS
  - Only author knows that.

- Documentation is tedious work :bow:

Because documentation is valuable work.



Testing and Running are easy

I always start code reading with the following commands

```
$ git clone https://github.com/some/gems
$ cd gems
$ less .travis.yml
```

I pick out `before_script` and `script` code from .travis.yml and invoke it. For example:

```
$ bundle install
$ rake spec
$ rake spec:plugins
```



In the case of ruby

```
$ git clone https://github.com/ruby/ruby
$ cd ruby
$ less .travis.yml
```

You will get…



.travis.yml in ruby/ruby

```
before_script:
  - "uname -a"
  - "uname -r"
  - "rm -fr .ext autom4te.cache"
  - "echo $TERM"
  - "make -f common.mk BASERUBY=ruby MAKEDIRS='mkdir -p' srcdir=. update-config_files"
  - "autoconf"
  - "mkdir config_1st config_2nd"
  - "./configure -C --disable-install-doc --with-gcc=$CC $CONFIG_FLAG"
  - "cp -pr config.status .ext/include config_1st"
  - "make reconfig"
  - "cp -pr config.status .ext/include config_2nd"
  - "diff -ru config_1st config_2nd"
  - "make after-update BASERUBY=ruby"
  - "make -s $JOBS"
  - "make update-rubyspec"
script:
  - "make test TESTOPTS=--color=never"
  - "make test-all TESTOPTS='-q -j3 --color=never --job-status=normal'"
  - "make test-rubyspec MSPECOPT=-fm"
```



How to contribute to ~~DSS~~ Ruby

Do submit your patch to ruby core

1. Write code :)
2. Run tests
3. Open bugs.ruby-lang.org and create new account
4. Open https://bugs.ruby-lang.org/projects/ruby-trunk/issues/new
5. Attach your code and write description of your proposal
6. Press “submit”



Do submit your patch to ruby core

1. Write code :)
2. Run tests
3. Open bugs.ruby-lang.org and create new account
4. Open https://bugs.ruby-lang.org/projects/ruby-trunk/issues/new
5. Attach your code and write description of your proposal
6. Press “submit”



3.

Test ruby language

Start to test Ruby language

You can invoke language tests with the following instructions:

```
$ git clone https://github.com/ruby/ruby
$ cd ruby
$ autoconf
$ ./configure —disable-install-doc
$ make -j
$ make check
```



```
% make check TESTS='-j4'
(snip)

PASS all 1010 tests
exec ./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext  -- --disable-gems "./bootstraptest/runner.rb" --ruby="ruby --disable-gems"
./KNOWNBUGS.rb
2016-06-18 15:42:02 +0900
Driver is ruby 2.4.0dev (2016-06-18 trunk 55440) [x86_64-darwin15]
Target is ruby 2.4.0dev (2016-06-18 trunk 55440) [x86_64-darwin15]
last_commit=* test/rubygems/test_gem_installer.rb: Fixed broken test with extension   build. https://github.com/rubygems/rubygems/pull/1645

KNOWNBUGS.rb  PASS 0
No tests, no problem

test succeeded
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext  -- --disable-gems"

# Running tests:

Finished tests in 2.513254s, 87.9338 tests/s, 177.4592 assertions/s.
221 tests, 446 assertions, 0 failures, 0 errors, 0 skips

ruby -v: ruby 2.4.0dev (2016-06-18 trunk 55440) [x86_64-darwin15]
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext  -- --disable-gems" --excludes-dir=./test/excludes --name=!/
memory_leak/ --exclude=/testunit/ --exclude=/minitest/ -j4

# Running tests:

[ 89/803] 4625=test_bigdecimal 4626=test_table 4627=test_encodings 4628=test_bug_reporter
```



common.mk

- check/check-ruby
- btest/btest-ruby
- test-basic/test-knownbug
- test-testframework
- test
- test-all
- test-almost
- test-ruby
- test-rubyspec



make test

- btest-ruby
  - snip for next slides
- test-basic
  - invoke `basictest/runner.rb` with target ruby
- test-knownbug
  - invoke `KNOWNBUGS.rb`
  - It’s empty a lot of the time.



make btest-ruby

```
% ls bootstraptest
pending.rb  runner.rb*  test_attr.rb  test_autoload.rb  test_block.rb  test_class.rb  test_eval.rb  test_exception.rb
test_finalizer.rb  test_flip.rb  test_flow.rb  test_fork.rb  test_gc.rb  test_io.rb  test_jump.rb  test_literal.rb  test_literal_suffix.rb
test_load.rb  test_marshal.rb  test_massign.rb  test_method.rb  test_objectspace.rb  test_proc.rb  test_string.rb  test_struct.rb
test_syntax.rb  test_thread.rb
```

What's `bootstraptest/runner.rb` ?

- load test files and invoke them
- define assertion methods like `assert_equal` etc.



`cat bootstraptest/test_class.rb`

```
assert_equal 'true',    %q( class C; end
                        Object.const_defined?(:C) )
assert_equal 'Class',   %q( class C; end
                        C.class )
(snip)

assert_equal 'Class',   %q( class A; end
                        class C < A; end
                        C.class )
(snip)

assert_equal 'M',       %q( module M; end
                        M.name )
(snip)

assert_equal 'A::B',    %q( class A; end
                        class A::B; end
                        A::B )
```



test-basic

```
a, = nil; test_ok(a == nil)
a, = 1; test_ok(a == 1)
a, = []; test_ok(a == nil)

(snip)

def r; return *[]; end; a = r(); test_ok(a == [])
def r; return *[1]; end; a = r(); test_ok(a == [1])
def r; return *[nil]; end; a = r(); test_ok(a == [nil])

(snip)

f = lambda { |a, b=42, *c| [a,b,c] }
test_ok(f.call(1      ) == [1,42,[  ]] )
test_ok(f.call(1,43   ) == [1,43,[  ]] )
test_ok(f.call(1,43,44) == [1,43,[44]] )

(snip)
```



These are for Developer of VM/GC/Syntax/etc

make test-all

test-all invokes test files under the `test` directory. test-all has some options for testing:

- make test-all TESTS=“logger”
  - test only files under `test/logger`

- make test-all TESTS=“-j4”
  - it make parallel execution with 4 processes.



cat `test/ruby/test_array.rb`

```
% cat test/ruby/test_array.rb
# coding: US-ASCII
require 'test/unit'

class TestArray < Test::Unit::TestCase

(snip)

  def test_percent_i
    assert_equal([:foo, :bar], %i[foo bar])
    assert_equal([:"\"foo"], %i["foo])
  end

  def test_0_literal
    assert_equal([1, 2, 3, 4], [1, 2] + [3, 4])
    assert_equal([1, 2, 1, 2], [1, 2] * 2)
    assert_equal("1:2", [1, 2] * ":")

(snip)
```



cat `test/logger/test_logger.rb`

```
% cat test/logger/test_logger.rb
# coding: US-ASCII
require 'test/unit'
require 'logger'
require 'tempfile'

class TestLogger < Test::Unit::TestCase

(snip)

 def test_add
    logger = Logger.new(nil)
    logger.progname = "my_progname"
    assert(logger.add(INFO))
    log = log_add(logger, nil, "msg")
    assert_equal("ANY", log.severity)
    assert_equal("my_progname", log.progname)
(snip)
```



cat `test/-ext-/array/test_resize.rb`

```
% cat ext/-test-/array/resize/resize.c
#include "ruby/ruby.h"

static VALUE
ary_resize(VALUE ary, VALUE len)
{
    rb_ary_resize(ary, NUM2LONG(len));
    return ary;
}

void
Init_resize(void)
{
    rb_define_method(rb_cArray, "__resize__", ary_resize, 1);
}
```

```
require 'test/unit'
require '-test-/array/resize'

class TestArray < Test::Unit::TestCase
  class TestResize < Test::Unit::TestCase
    def test_expand
      feature = '[ruby-dev:42912]'
      ary = [*1..10]
      ary.__resize__(10)
      assert_equal(10, ary.size, feature)
      assert_equal([*1..10], ary, feature)
      ary.__resize__(100)
      assert_equal(100, ary.size, feature)
(snip)
```



These are for “you”

make check

make check depends on the following definitions:

- main
  - build encodings and extensions.
- test
- test-testframework
  - run tests for `testunit` and `minitest`
- test-almost
  - run tests under `test` excluding `testunit` and `minitest`



test-unit/minitest

Why separated the test framework?

The following libraries uses minitest directly in Ruby 2.3:

- **rubygems**
- **rdoc**

Other libraries uses test-unit. rubygems and rdoc are developed at github.com/rubygems/rubygems and github.com/rdoc/rdoc. We need to support these libraries and their tests.



How to merge upstream from others

I merged upstream into ruby/ruby periodically using following instructions.

```
$ git clone https://github.com/ruby/ruby
$ git clone https://github.com/rubygems/rubygems
$ cd ruby
$ rm -rf lib/rubygems test/rubygtems lib/rubygems.rb
$ cp -rf ../../rubygems/rubygems/lib/rubygems ./lib
$ cp -rf ../../rubygems/rubygems/lib/rubygems.rb ./lib
$ cp -rf ../../rubygems/rubygems/test/rubygems ./test
$ git checkout lib/rubygems/LICENSE.txt
```

ruby and rubygems guarantee to work to test and code each other.
it’s the same situation for ruby and rdoc



backport is hard

rubygems still supports Ruby 1.8.

```
% g show a34fb569e41cd87866e644d92a9df4be89b3cad2 test/rubygems/test_gem_package.rb
commit a34fb569e41cd87866e644d92a9df4be89b3cad2
Author: Eric Hodel <drbrain@segment7.net>
(snip)
--- test/rubygems/test_gem_package.rb
+++ test/rubygems/test_gem_package.rb
@@ -638,7 +638,7 @@ class TestGemPackage < Gem::Package::TarTestCase
                    e.message
       io
     end
-    tf.close!
+    tf.close! if tf.respond_to? :close!
   end

   def test_verify_empty
```



ruby/spec

RubySpec

Q. What's rubyspec?
A. RubySpec is an executable specification for the Ruby programming language.

"Matz's Ruby Developers Don't Use RubySpec and It's Hurting Ruby"
http://rubini.us/2014/12/31/matz-s-ruby-developers-don-t-use-rubyspec/

"ruby/spec" is not a "specification". It's actually a set of "test". The only real ruby specification is inside of Matz :)



make test-rubyspec

CRuby has `make update-rubyspec` and `make test-rubyspec` tasks.

- `make update-rubyspec`
  - pulls ruby/rubyspec and ruby/mspec into the spec directory.

- `make test-rubyspec`
  - invokes mspec with the ruby binary and the latest rubyspecs.



cat spec/rubyspec/core/string/append_spec.rb

```
% cat spec/rubyspec/core/string/concat_spec.rb
require File.expand_path('../../../spec_helper', __FILE__)
require File.expand_path('../fixtures/classes', __FILE__)
require File.expand_path('../shared/concat', __FILE__)

describe "String#<<" do
  it_behaves_like :string_concat, :<<
  it_behaves_like :string_concat_encoding, :<<
end
```

```
% cat spec/rubyspec/core/string/shared/concat.rb
describe :string_concat, shared: true do
  it "concatenates the given argument to self and returns self" do
    str = 'hello '
    str.send(@method, 'world').should equal(str)
    str.should == "hello world"
  end

(snip)
```

Please check details: https://github.com/ruby/spec



rubyspec and mspec

We approved new or updated examples at github.com/ruby/spec.

@headius wrote: “So nice to see RubySpec getting a steady stream of Ruby 2.3 specs.”

https://twitter.com/headius/status/667793518098673664

A lot of contributors submitted new specs for Ruby 2.3 and 2.4 features.



test coverage

I added a coverage task using simplecov

```
% make update-coverage
updating simplecov ...
remote: Counting objects: 90, done.
(snip)
updating simplecov-html ...
updating doclie …
% COVERAGE=1 make test-all TESTS=webrick
```

You can get coverage results for `webrick` under the coverage directory.



```ruby
# The ERB file is evaluated with the local variables +servlet_request+ and
# +servlet_response+ which are a WEBrick::HTTPRequest and
# WEBrick::HTTPResponse respectively.
#
# Example .rhtml file:
#
#   Request to <%= servlet_request.request_uri %>
#
#   Query params <%= servlet_request.query.inspect %>

class ERBHandler < AbstractServlet

  ##
  # Creates a new ERBHandler on +server+ that will evaluate and serve the
  # ERB file +name+

  def initialize(server, name)
    super(server, name)
    @script_filename = name
  end

  ##
  # Handles GET requests

  def do_GET(req, res)
    unless defined?(ERB)
      @logger.warn "#{self.class}: ERB not defined."
      raise HTTPStatus::Forbidden, "ERBHandler cannot work."
    end
    begin
      data = open(@script_filename){|io| io.read }
      res.body = evaluate(ERB.new(data), req, res)
      res['content-type'] ||=
        HTTPUtils::mime_type(@script_filename, @config[:MimeTypes])
    rescue StandardError
      raise
    rescue Exception => ex
      @logger.error(ex)
      raise HTTPStatus::InternalServerError, ex.message
    end
  end

  ##
```

Please contribute tests to ruby

You can invoke focused tests with coverage:

```
$ COVERAGE=1 make test-all TESTS=“logger”
$ open coverage/index.html
```

You can code new tests or update existing tests. and submit it.



Ruby (language) testing is so easy

4.

How request Ruby core

Feature request

- We need to focus “Use case” than “function”.
- We need to attach patch to feature request.
- We need Matz approval.



Acceptable requests without usecase

- Symmetrical:

  `Array#shift`, `Array#unshift`
- POSIX

  Socket.gethostbyname
- [BUG] [SEGV]



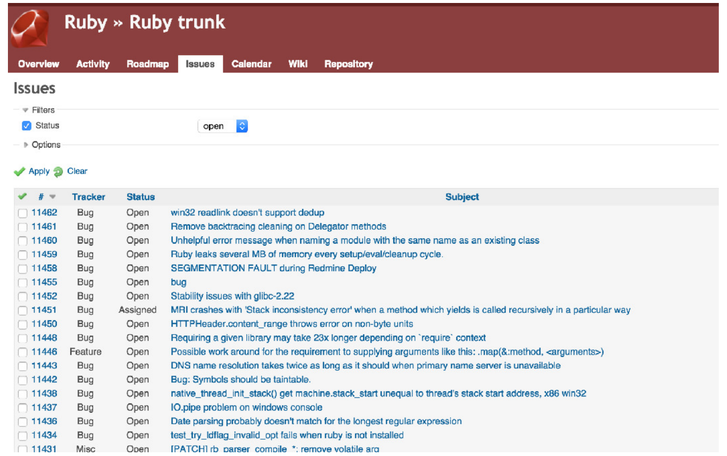
Bug report

- We need minimum reproduction code
- We need to describe expected/unexpected behavior
- We need to attach crash log
- We need to try different version of ruby



Issue tracker

Our official tracker is “bugs.ruby-lang.org”



Mailing list integration

- https://github.com/ruby/redmine_*
- This behavior is same as github

Continuous Upgrade Ruby and Rails to latest version.



Redmine vs GitHub

github.com/ruby/ruby is ok for ruby core team

[CAUTION] If you hope to ask new feature to Matz, You need to submit bugs.ruby-lang.org . Matz is only available on redmine.

Why Ruby does not use GitHub?

- GitHub is proprietary service
- ruby committers do not have problem with redmine
- We start to consider issue of “Why Ruby does not use Git”



Monthly Developer Meeting

We hope to increase to transparency for Ruby development process.

One of our challenges is “Developer Meeting”. It’s open discussion time for feature and issue of Ruby every months.

[ruby-core:69550]: https://bugs.ruby-lang.org/projects/ruby/wiki/DevelopersMeeting20150728Japan



Gemification for stdlib

- We extract old or un-maintain status stdlibs like net-telnet, xmlrpc, tk to bundled gems.
  - These are extracted under the https://github.com/ruby/ . And shipped on rubygems.org
  - Date, CGI, Readline are also extracted at the future(It's my ideas).
- OpenSSL is also extracted default gems. You can update it separated ruby core releases same as rubygems, rdoc, bigdecimal



5.

How develop Ruby core

Ruby CI

What's Ruby CI

Ruby CI is a CI results collector for alternative platforms

- **http://rubyci.org**
- **https://github.com/nurse/rubyci**
- **https://github.com/ruby/chkbuild**

Ruby CI goal is entirely supports all of Ruby platform. We can detect a lot of build fails using Ruby CI. It has 2 or 3 versions every linux distribution and BSD, Windows, OS X, Solaris Environments.



Server	Datetime	Branch	Option	Revision	test	test-all	rubyspec	Summary	Diff
Debian 7.5 i686	08-12 14:05	trunk		55879				480W success	diff:configure,t...
Debian 7.9 x86_64	08-12 12:00	trunk		55865				156W success	diff:test-all
Debian 8.2 x86_64	08-12 12:30	trunk		55877				151W success	diff:src,test-all,...
Debian 9.0(testing) x86_64	08-12 12:30	trunk		55877				149W success	diff:src,proces...
Ubuntu armv7l eabihf	08-12 11:17	trunk		55865				167W success	diff:miniruby,m...
Ubuntu 15.10 x86_64	08-12 12:30	trunk		55877				162W success	diff:src,proces...
Ubuntu 16.04 x86_64	08-12 12:30	trunk		55877				149W success	diff:src,proces...
Gentoo	08-12 12:30	trunk		55865				152W success	diff:test-all
CentOS 5 i386	08-12 13:03	trunk		55879				245W success	diff:src,test-all
CentOS 5 x86_64	08-12 12:33	trunk		55877				229W success	diff:src,test-all
CentOS 6 64bit	08-12 14:02	trunk		55879				152W success	diff:src,test-all
CentOS 7.1 x86_64	08-12 13:00	trunk		55879				163W success	diff:src,test-all
Fedora 22 x86_64	08-12 13:15	trunk		55879				167W success	diff:src,proces...
Fedora 23 x86_64	08-12 09:30	trunk		55865				167W success	diff:process-st...
Amazon Linux 2016.03 x64	08-12 09:30	trunk		55865				159W success	diff:process-st...
Arch Linux	08-12 09:30	trunk		55865	BF			173W failed(btest)	diff:process-st...
openSUSE 13.2	08-12 12:30	trunk		55877				157W success	diff:src,make,t...
AIX 7.1 ppc	08-11 19:33	trunk		55863		1F0E	4F38E	273W 1F0E rubyspec:4F38E	no diff
RHEL 7.1 s390x	08-12 06:33	trunk		55865				173W success	diff:src,test-all
FreeBSD 10.1 x86	08-12 14:03	trunk		55879				147W success	diff:src,configu...
FreeBSD 10.1 x64	08-12 14:03	trunk		55879				147W success	diff:src,configu...
OS X Yosemite	08-12 08:45	trunk		55865				157W success	diff:configure,t...
OS X El Capitan	08-11 06:45	trunk		55857		1F0E		156W 1F0E	diff:configure,t...
icc16-x64	08-12 13:00	trunk		55879		1F0E		152W 1F0E	diff:src,make,t...
Solaris 10x	08-12 14:18	trunk		55879				143W success	diff:pflags,make
Solaris 11x	08-12 13:24	trunk		55879				216W success	diff:src,pflags,t...
Solaris 10s	08-12 10:19	trunk		55865		0F1F		207W 0F1F	diff:pflags,test

How to add a new server

You can add your server to rubyci.org

Requirements:

- **not yet supported platforms.**
  - **ex. linux with ARM, *BSD, icc with OSX, Windows**
- **periodically running every day**
- **It must be possible to access to AWS S3**

You should check the following commands on your server

```
$ git clone https://github.com/akr/chkbuild
$ cd chkbuild
$ ruby start-build
```



Version number and release cycle

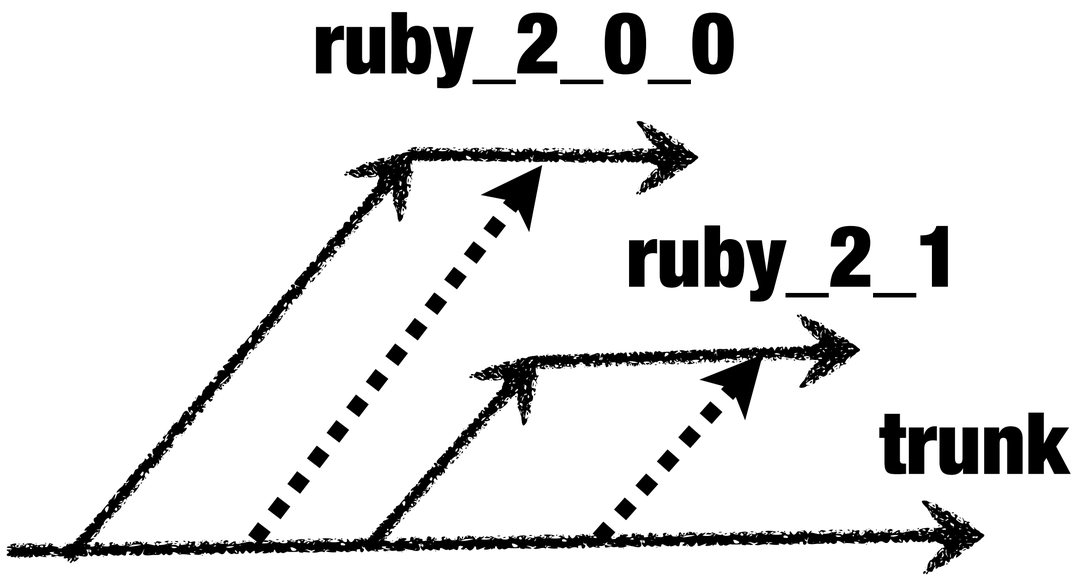
We plan to release every christmas.

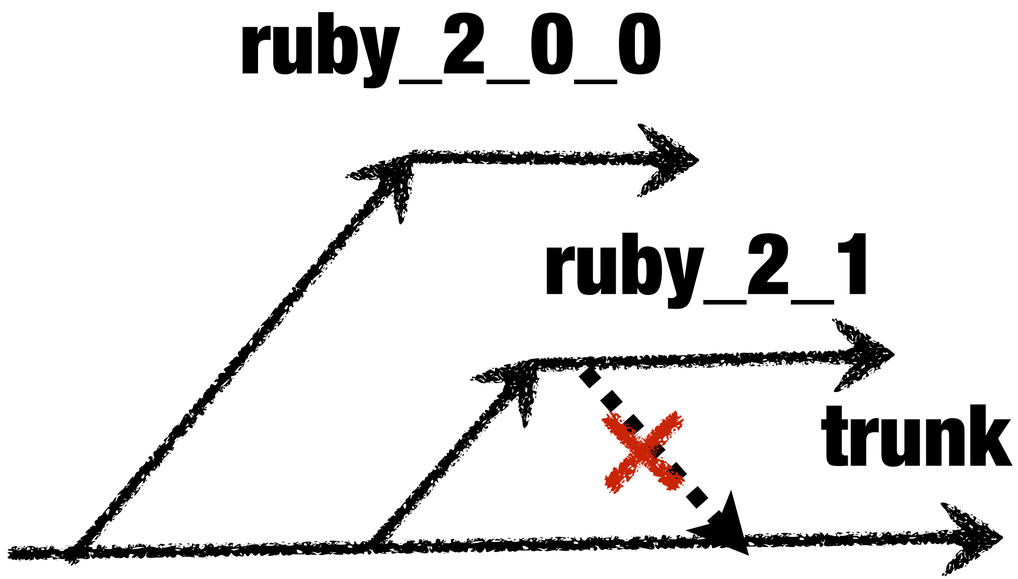
- **2.1.0: 2013/12/25**
- **2.2.0: 2014/12/25**
- **2.3.0: 2015/12/25**
- **2.4.0: 2016/12/25(TBD)**



Ruby core backport model

We backport fixes to stable branch from trunk.
We do not merge fixes to trunk from stable branch





Release management

We will release new version of Ruby at “Release Day” by @narse

There is no exception to this rule.

- If we have incompletion issue or feature, we will revert it.
- If we don’t have enough discussion for some issue, we don’t merge or implement it into new version of ruby.
- If we found some regression, we need to fix it or revert to related code or issue.



Security release

We have “security@ruby-lang.org” for security report. We received buffer overflow, memory leak, escape string etc etc…

We hard to fix and release these security issue. so all of release maintainer are volunteer work.

Our release delayed by preparing new releases of stable and old stable version.

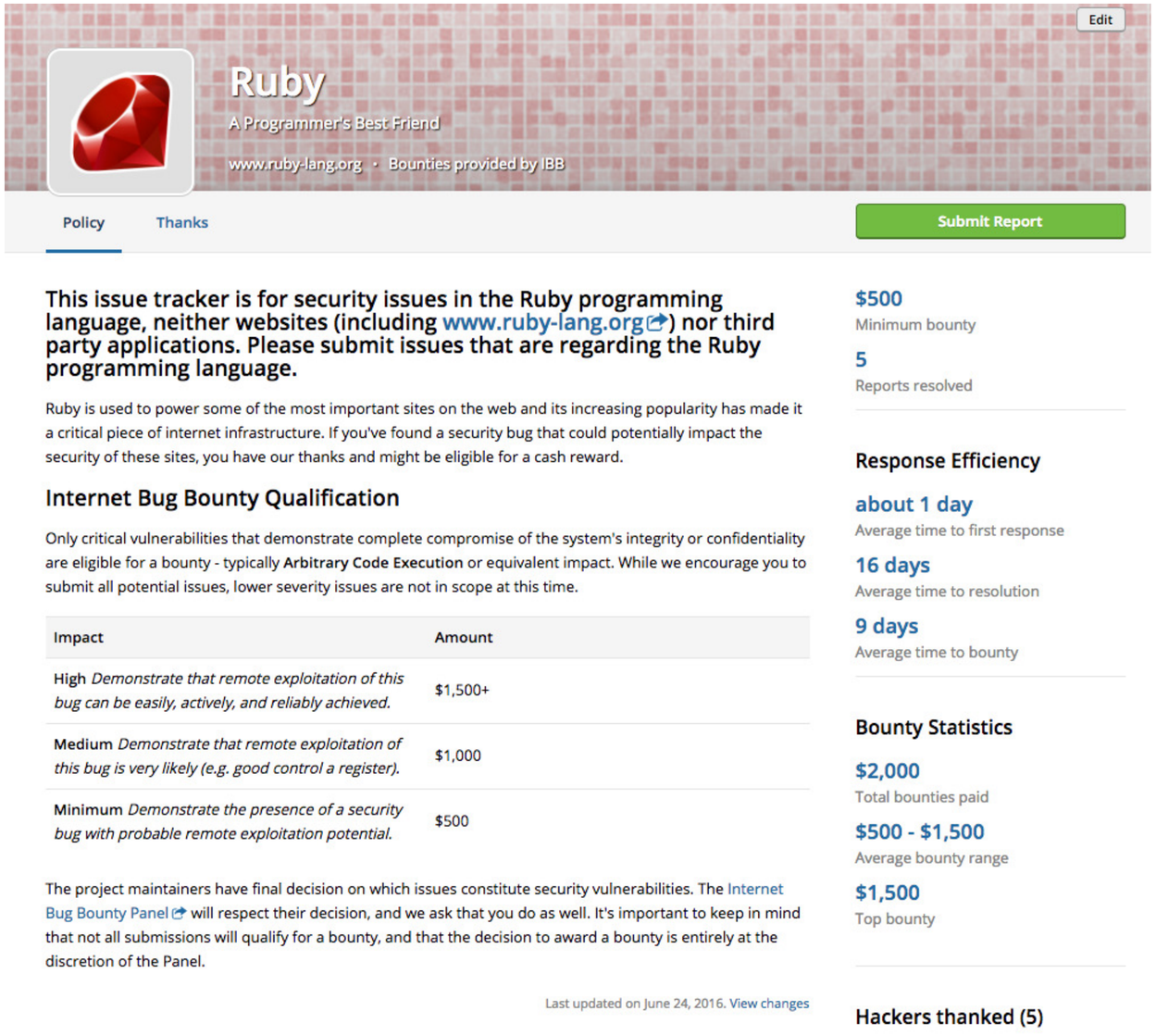


HackerOne

We've been use https://hackerone.com/ruby

It has bounty program provided by IBB.

We accepts to security report via hackerone too.





Ruby

A PROGRAMMER'S BEST FRIEND

Search

Downloads Documentation Libraries Community **News** Security About Ruby

Ruby 2.4.0-preview2 Released

Posted by naruse on 8 Sep 2016

We are pleased to announce the release of Ruby 2.4.0-preview2.

Ruby 2.4.0-preview2 is the second preview of Ruby 2.4.0. This preview2 is released to get feedback from the community. Feel free to send feedback since you can still influence the features.

Unify Fixnum and Bignum into Integer

Though ISO/IEC 30170:2012 doesn't specify details of the Integer class, Ruby had two visible Integer classes: Fixnum and Bignum. Ruby 2.4 unifies them into Integer. All C extensions which touch the Fixnum or Bignum class need to be fixed.

See also the ticket and akr's slides.

Recent News

Ruby 2.4.0-preview2 Released

ConFoo Montreal 2017 is looking for Ruby speakers

Ruby Prize 2016 nominations are now open

Ruby 2.4.0-preview1 Released

ConFoo Vancouver 2016 is looking for Ruby speakers

Syndicate

Recent News (RSS)

Conclusion

CRuby secrets

- Who develop CRuby
- Where is official resource of CRuby
- How to develop CRuby
- What is important of feature request in CRuby
- When release new version of CRuby

and…



Why use Ruby?

“Ruby is designed to make programmers happy.”