GMOペパボ

The secret of programming language development and future

The story of Ruby Ecosystem

SHIBATA Hiroshi / GMO Pepabo, inc.

2017.05.21,23 CodeEurope 2017

GMOペパボ



Hiroshi SHIBATA @hsbt

Executive Officer CPO(Chief Productivity Officer)

Director of Business Process Re-engineering Office

https://www.hsbt.org



最新ニュース　セミナー・イベント情報　IRニュース

2016.09.08　プレスリリース　minne

GMOペパボが運営する国内最大のハンドメイドマーケット「minne（ミンネ）」 Apple Payへの対応開始について

グローバルサイン認証サイト
SSL secured クリックして確認
GlobalSign by GMO

https://pepabo.com

self.introduce

=>
{
  name: “SHIBATA Hiroshi”,
  nickname: “hsbt”,
  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”],
}



asakusarb

foo category:bar/baz comment:foobar

README

Imports 426

meetup 389

hanami 1

misc 1

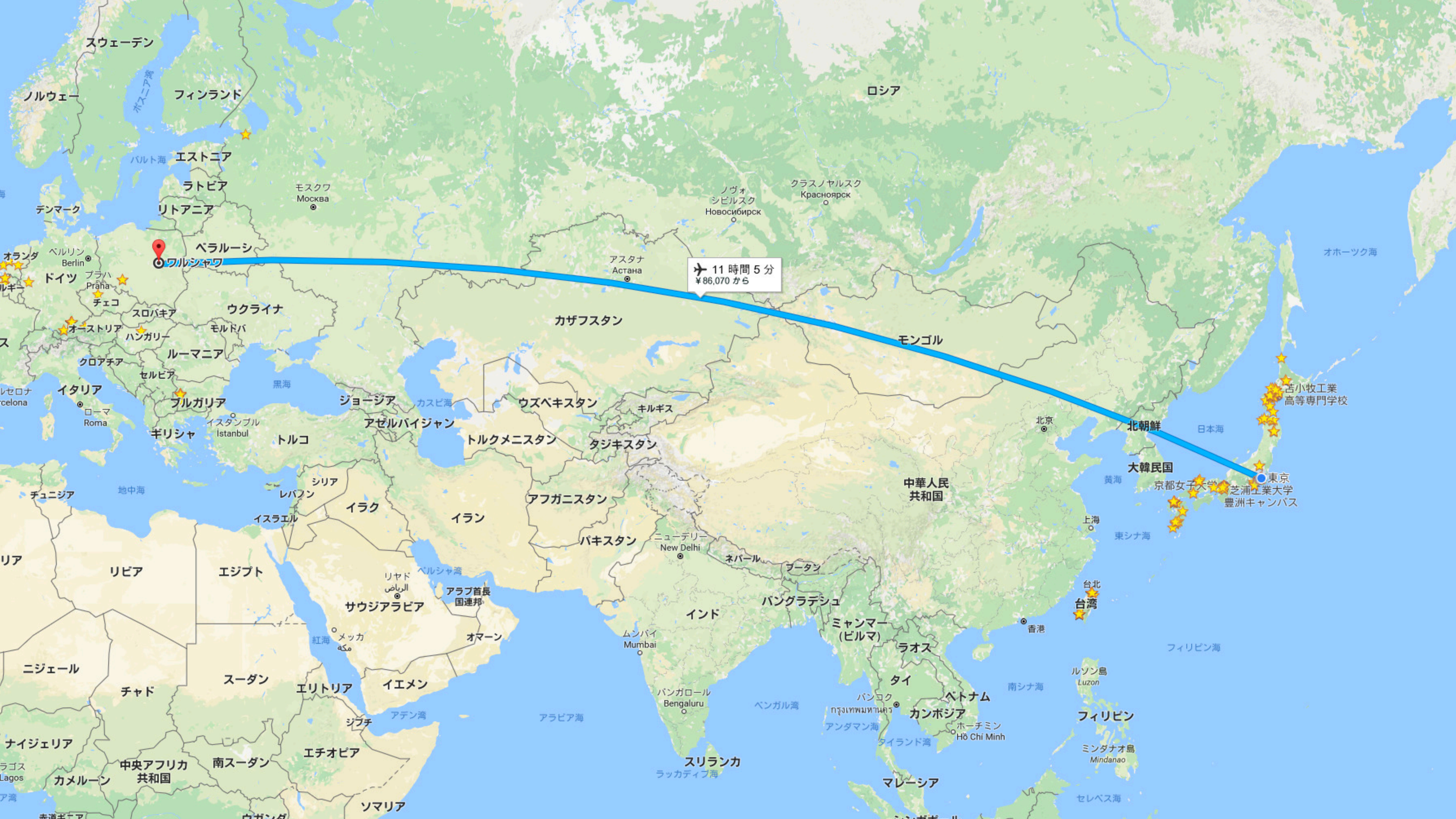
000_Log 2

(no category) 5

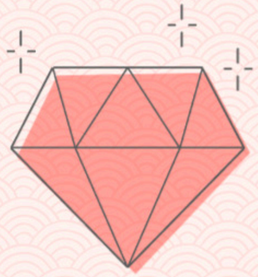
README.md

Edit





スウェーデン
ノルウェー
ボスニア湾
フィンランド
ロシア
バルト海
エストニア
ラトビア
リトアニア
モスクワ
Москва
ノヴォシビルスク
Новосибирск
クラスノヤルスク
Красноярск
デンマーク
ベラルーシ
ワルシャワ
オランダ
ベルリン
Berlin
ドイツ
プラハ
Praha
チェコ
スロバキア
ウクライナ
オーストリア
ハンガリー
モルドバ
ルーマニア
クロアチア
セルビア
イタリア
ローマ
Roma
ブルガリア
イスタンブル
İstanbul
ギリシャ
トルコ
黒海
ジョージア
カスピ海
アゼルバイジャン
カザフスタン
アスタナ
Астана
11 時間 5 分
¥86,070 から
モンゴル
ウズベキスタン
キルギス
トルクメニスタン
タジキスタン
北京
北朝鮮
日本海
オホーツク海
苫小牧工業
高等専門学校
大韓民国
黄海
京都女子大学
芝浦工業大学
豊洲キャンパス
東京
中華人民
共和国
上海
東シナ海
チュニジア
地中海
シリア
レバノン
イスラエル
イラク
イラン
アフガニスタン
パキスタン
ニューデリー
New Delhi
ネパール
ブータン
リビア
エジプト
リヤド
الرياض
ペルシャ湾
アラブ首長
国連邦
サウジアラビア
インド
バングラデシュ
台北
台湾
香港
ミャンマー
(ビルマ)
ラオス
フィリピン海
ムンバイ
Mumbai
オマーン
紅海
メッカ
مكة
ニジェール
チャド
スーダン
エリトリア
イエメン
タイ
ルソン島
Luzon
南シナ海
バンガロール
Bengaluru
ベンガル湾
バンコク
กรุงเทพมหานคร
ベトナム
カンボジア
フィリピン
アデン湾
アラビア海
ジブチ
アンダマン海
ホーチミン
Hồ Chí Minh
タイランド湾
ナイジェリア
エチオピア
ミンダナオ島
Mindanao
中央アフリカ
共和国
南スーダン
スリランカ
ラッカディブ海
カメルーン
マレーシア
セレベス海
ソマリア



RubyKaigi 2017

Sep.18th-20th

International Conference Center Hiroshima,
Hiroshima, Japan

Call for Speakers | Become a Sponsor

Anti Harassment Policy | Team



This work by RubyKaigi 2017 Team is licensed under a Creative Commons Attribution 3.0 Unported License.

1.

What’s (C)Ruby?

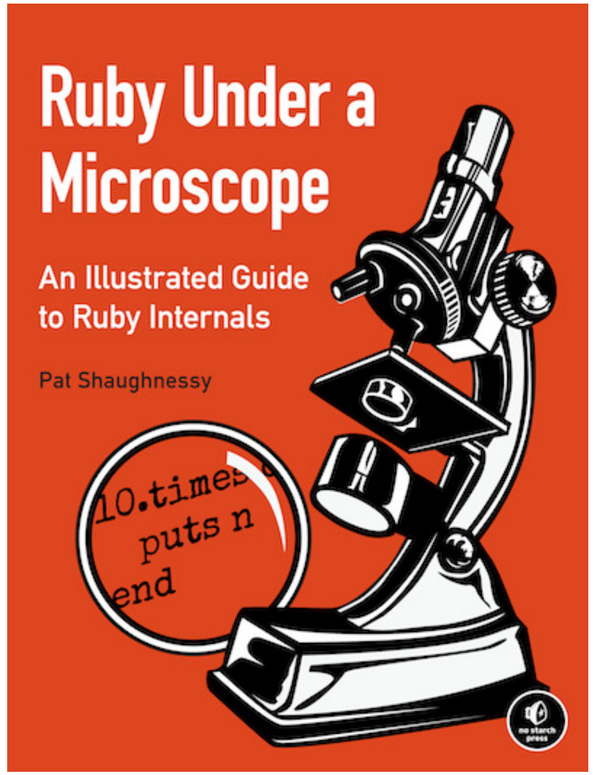
Variation of Ruby Interpreter

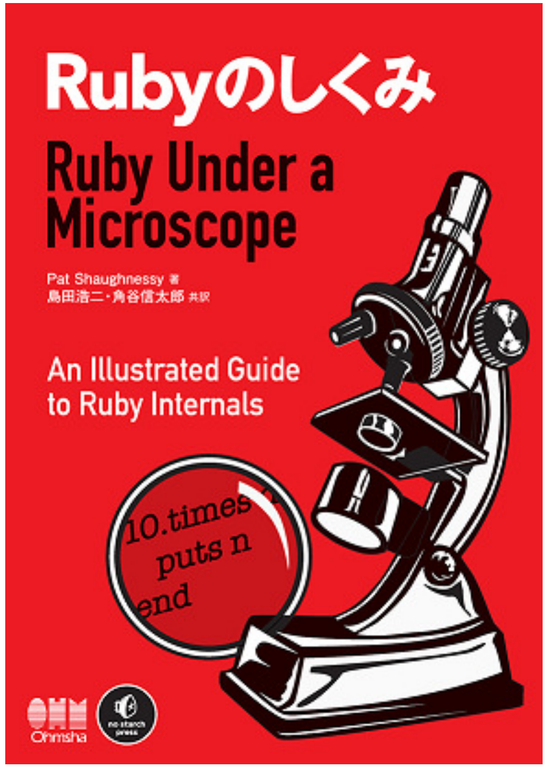
Implementation:

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

ISO/IEC 30170:2012 compatible:

- 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 or later

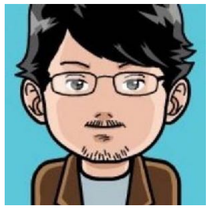
Ruby Under a Microscope, p.33



Ruby Committer

Full-time commiters

Heroku




- matz
- n0kada

Money Forward



- shyouhei

Cookpad

- ko1

Speee



- mrkn

Total 94 people + 2 bot

```
$ cat ~svn/.ssh/authorized_keys | awk '{print $5}' | sort | uniq | wc -l
96
```

Branch maintainers

trunk known as 2.5

nurse: Release manager



2.4

nagachika: Stable branch maintainer



2.3, 2.2

unak: Old stable branch maintainer





Platform maintainers

Linux



- kosaki
- normalperson
- akr
- n0kada

and others

BSD



- nurse

Solaris



- ngoto

macOS



- n0kada

Windows



- unak
- n0kada

Ruby Core Maintenance scope policy

Language core features including security, syntax

Yukihiro Matsumoto (a.k.a Matz)

Ruby VM(YARV), GC, Thread

Koichi Sasada (ko1)

Core classes

Ruby Committers



Ruby Core Maintenance Policy

Standard Libraries

- Each maintainers

Default Gems

- Hiroshi SHIBATA(me), Kazuki Yamaguchi(rhe)
- Nobuyoshi Nakada(nobu)

Bundled Gems

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



*.ruby-lang.org

www.ruby-lang.org

Official Website of Ruby language.

We welcomed to translate contribution. (Polish is also welcome!)

see http://github.com/ruby/www.ruby-lang.org

svn.ruby-lang.org

Main repository of Ruby source. (Not git !!!)

bugs.ruby-lang.org

Official issue tracker build by redmine.



*.ruby-lang.org

ftp.ruby-lang.org

Site of distribute official package

docs.ruby-lang.org

Hosted document generated from RDoc on Ruby source code and rubima project. Rubima project is a Japanese Documentation about ruby languages.



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**



Sponsorship

We have a lot of sponsor offer from company and university.

- **fastly: OSS plan of CDN for *.ruby-lang.org**
- **heroku: Unlimited dyno resources for websites**
- **NaCl: Network and Compute resources for website**
- **Sugaya research laboratory: Network resources for macOS Srv**
- **Internet Initiative Japan(IIJ): Compute resource for website**
- **Ruby Association and Ruby-no-Kai: Grant of development**
- **Microsoft: Provides MSDN Enterprise license**
- **…**



What does mean “official”?

“official” means “Matz controllable”

Un-controllable examples:

- ruby-doc.org
- rubygems.org
- bundler.io
- Ruby version manager(rvm/rbenv/chruby)

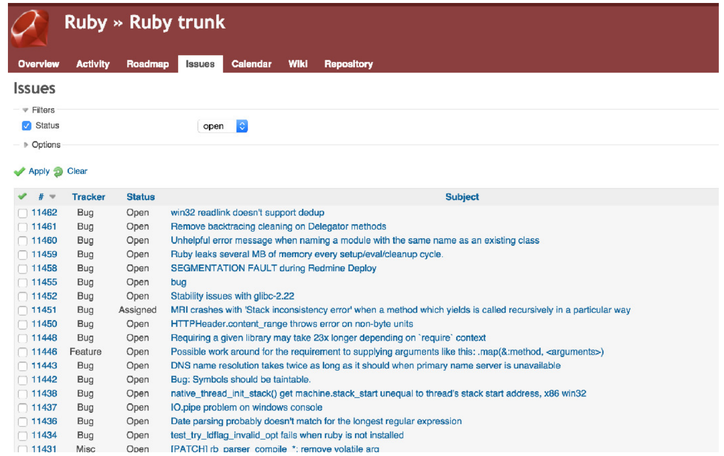


2.

How request Ruby core

Issue tracker

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



Mailing list integration

- ruby-core is official mailing list to develop
- see https://lists.ruby-lang.org
- This integration is same as GitHub’s one.



Redmine vs GitHub

https://github.com/ruby/ruby is acceptable space 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 and svn



Feature request

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



Acceptable requests without usecase

- **Symmetrical:**

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

  `Socket.gethostbyname`
- **[BUG] [SEGV]**



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/DevelopersMeeting20170417Japan



DevelopersMeeting20170417Japan

Date: 2017/04/17 (Mon)
Time: 14:00- 19:00 (JST)
Place: Cookpad Inc. Headquarter
Sign-up: https://ruby.connpass.com/event/53124/
log edit: https://docs.google.com/document/d/1_tWfg7_t-JRXgGlMaOGnpyipGP7P-IKQ18NtciWhKIY/edit#heading=h.b7fl897vl6up
log: https://docs.google.com/document/d/1_tWfg7_t-JRXgGlMaOGnpyipGP7P-IKQ18NtciWhKIY/pub
Attendees:
Language: mostly Japanese (sorry for non native Japanese speakers)

Please add your favorite ticket numbers you want to ask to discuss.

- NOTE
  - Dev meeting *IS NOT* a decision making place. All decisions should be done at the bug tracker.
  - Dev meeting is a place we can ask Matz, nobu, nurse and other developers directly. Matz is a very busy person. Take this opportunity to ask him. If you can not attend, other attendees can ask instead of y
    attendees can understand your issue).
  - We will write a log about discussion to a file or to each ticket in English.
  - All activities are best-effort (keep in mind that most of us are volunteer developers).
  - The date, time and place is scheduled according to when/where we can reserve Matz's time.

Agenda

- NOTE: Write at least "ticket number/title/link" and your name (see example below). Explain details on the ticket. If you cannot attend the meeting, we appreciate a short summary because we can understand it n
  (long discussion is difficult to read, especially in a non-native language). Your motivation is also welcome.

About 2.5 timeframe

- https://bugs.ruby-lang.org/projects/ruby-trunk/wiki/ReleaseEngineering25

Carry-over from previous meeting(s)

- [Bug #13257] Symbol#respond_to?(:singleton_class) should be false (naruse)
- [Misc #13163] Uncaught exceptions may not be reported when Thread#report_on_exception=true and Thread#abort_on_exception=true (naruse)
- [Bug #13181] Unexpected line in rescue backtrace (shyouhei) what's this?
- [Feature #8263] Support discovering yield state of individual Fibers (shyouhei)
- [Feature #8576] Add optimized method type for constant value methods (shyouhei)
- [Bug #13188] Reinitialize Ruby VM. (shyouhei) New C API ...?
- [Feature #2740] Extend const_missing to pass in the nesting (shyouhei)
- [Feature #13211] Hash#delete taking a splat (shyouhei)
- [Bug ~~#11567~~] Segmentation fault CFUNC :gets (shyouhei)
- [Feature #13172] Method that yields object to block and returns result (shyouhei) new name candicate
- [Feature #13224] Add FrozenError as a subclass of RuntimeError (shyouhei)
- [Misc ~~#13230~~] Better Do ... while structure (shyouhei)
- [Feature #13245] [PATCH] reject inter-thread TLS modification (shyouhei)
- [Feature #13252] C API for creating strings without copying (shyouhei)
- [Bug #13249] Access modifiers don't have an effect inside class methods in Ruby >= 2.3 (shyouhei) design issue or bug?


akr
nalsh
n0kada
mrkn
Matz
k0kubun
shyouhei
ko1
amatsuda

3.

How develop ruby language

Licenses of Ruby language

- 2-clause BSDL
- Ruby License

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



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
```



make test

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



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)
```



ruby/spec

ruby/spec

Q. What's ruby/spec?
A. ruby/spec is an spec style test suites for the Ruby programming language.

"ruby/spec" is not a "specification". It's actually a set of "test".

The Ruby specification is only inside of Matz :)



make test-spec

CRuby has `make test-spec` task.

- `make test-spec`
  - invoke mspec with the ruby binary and the latest rubyspecs.
- `make exam`
  - invoke `make check` and `make test-spec`



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



ruby/spec and mspec

We approved new or updated examples at https://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.



4.

How release Ruby core

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**
- **2.5.0: 2017/12/25(TBD)**
- **…**
- **3.0.0: 2020/xx/xx**



Release management

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

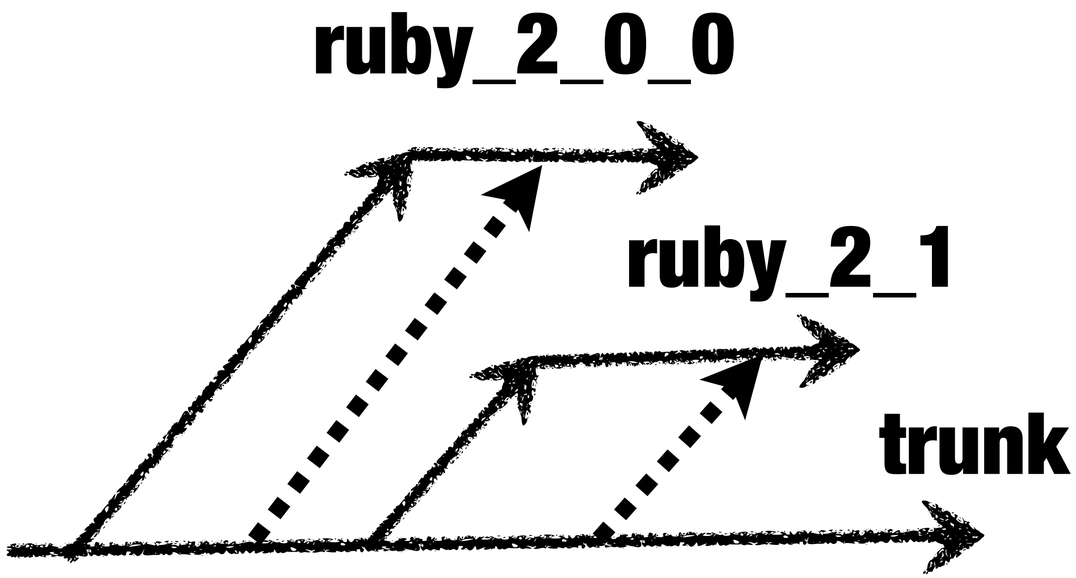
There is no exception to this rule.

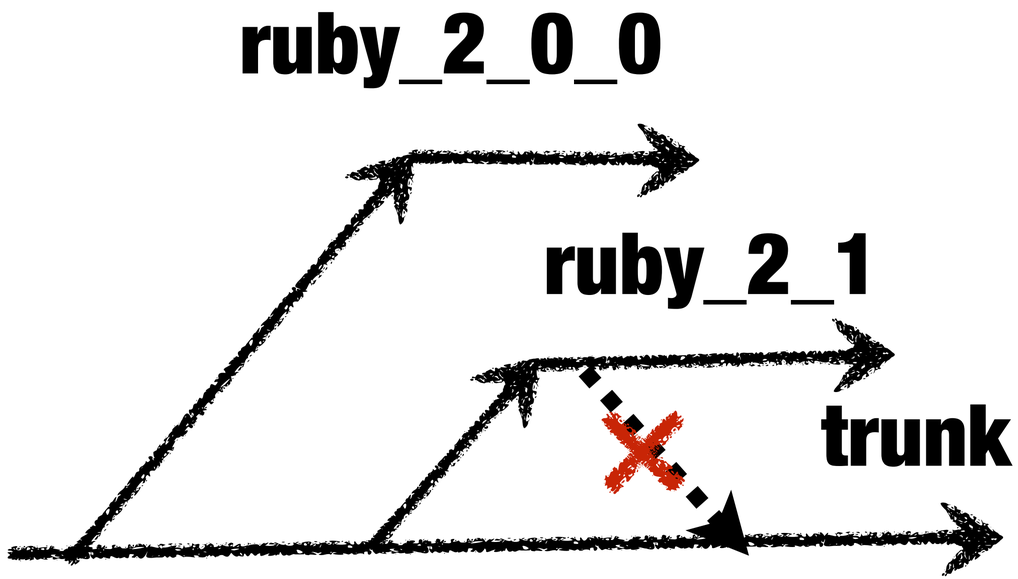
- If we have incompletion issue or feature, we will revert it.
- If we don’t have enough discussion for some issue, we don’t merge or implement it into new version of ruby.
- If we found some regression, we need to fix it or revert to related code or issue.



Ruby core backport model

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





Ruby CI

What's Ruby CI

Ruby CI is a CI results collector for alternative platforms

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

Ruby CI goal is entirely supports all of Ruby platform.



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

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.

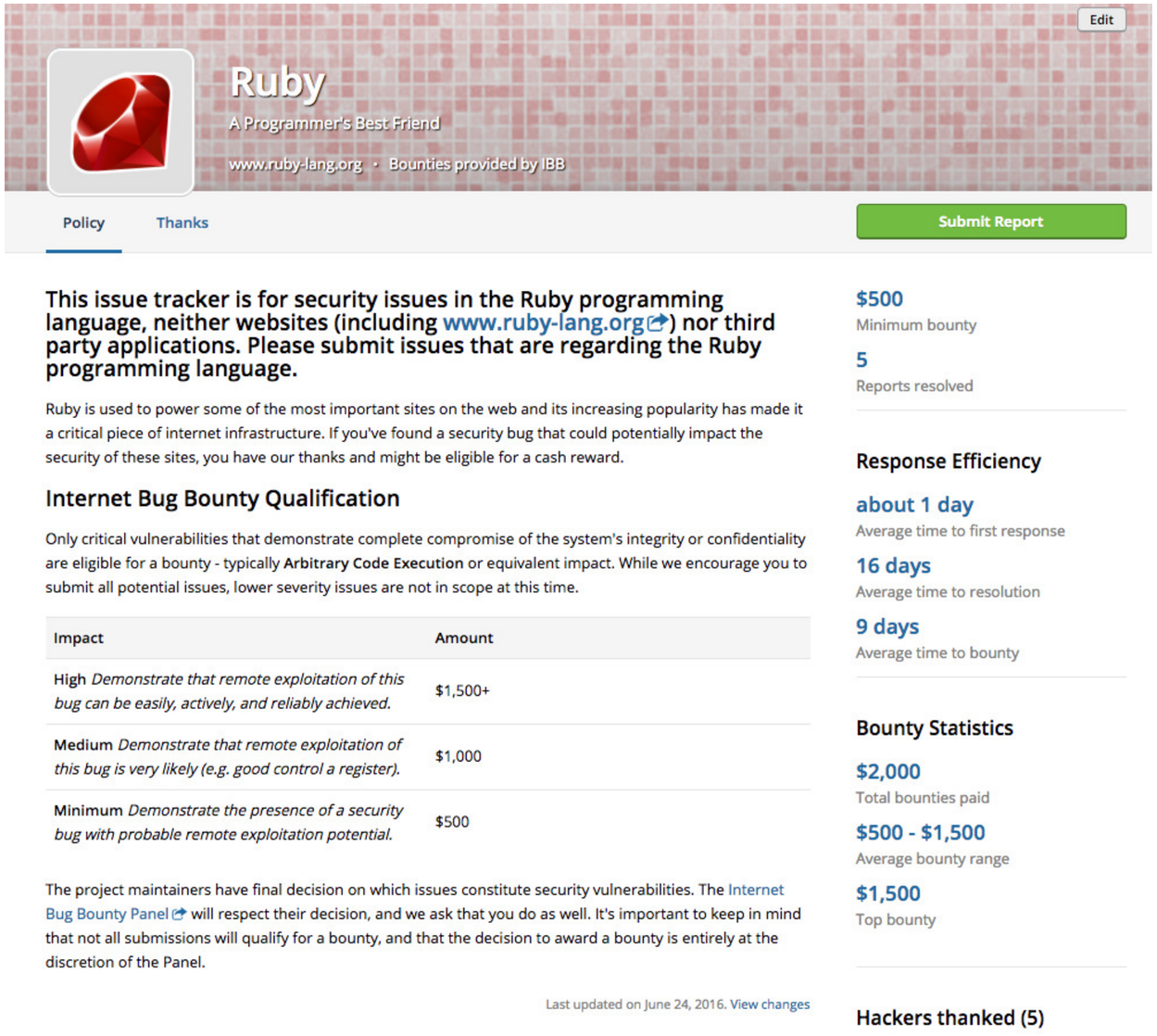


HackerOne

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

It has bounty program provided by IBB.

We accepts to security report via hackerone too.



5.

Language update
Ruby 2.4

Ruby 2.4

Ruby 2.4.0 was released 25, Dec 2016.

https://www.ruby-lang.org/en/news/2016/12/25/ruby-2-4-0-released/

Ruby 2.4 series introduced following improvements.

- Hash optimization
- binding.irb
- Integer Unification
- Unicode case mappings



Hash Optimization

vmakarov (Vladimir Makarov) introduced new implementation of hash tables.

This improvement has been discussed with many people, especially with Yura Sokolov. Matz chose vmakarow's patch finally.

https://bugs.ruby-lang.org/issues/12142



binding.irb

It is same as `binding.pry` without pry gem.

https://github.com/ruby/ruby/commit/493e48897421d176a8faf0f0820323d79ecdf94a

It helps following situation in the world. :)

```
> cat foo.rb
p :foo
binding.pry
p :bar
> ruby foo.rb
:foo
foo.rb:2:in `<main>': undefined method `pry' for #<Binding:0x00007fbc2483ad78> (NoMethodError)
> gem i pry
(waiting for some minutes!!!!!!)
```



Integer Unification

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.

https://bugs.ruby-lang.org/issues/12005

Ruby 2.3.4

```
>> 4611686018427387903.class
=> Fixnum
>> 4611686018427387904.class
=> Bignum
```

Ruby 2.4.1

```
>> 4611686018427387903.class
=> Integer
>> 4611686018427387904.class
=> Integer
```



Unicode case mappings

Add non-ASCII case conversion to String#upcase/downcase/swapcase/capitalize.

https://bugs.ruby-lang.org/issues/10085

Ruby 2.3.4

```
>> "lubię".upcase
=> "LUBIę"
```

Ruby 2.4.1

```
>> "lubię".upcase
=> "LUBIĘ"
```



6.

Language Update
Ruby 2.5

Ruby 2.5

- Gemification of standard libraries
  - https://bugs.ruby-lang.org/issues/5481
  - I’m going to describe this details in this talk :)
- in-tree Rubyspec:
  - https://bugs.ruby-lang.org/issues/13156
  - Ruby 2.5 has in-tree copy of ruby/spec. We aggressively improve for alternative implementation of Ruby interpreter.



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
  - Other gems are also extracted at the future.



Ruby Core Maintenance Policy(again)

Standard Libraries

- Each maintainers

Default Gems

- Hiroshi SHIBATA(me), Kazuki Yamaguchi(rhe)
- Nobuyoshi Nakada(nobu)

Bundled Gems

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



What differences these libraries?

Standard Libraries

- Upstream: Only Ruby core repository(svn.ruby-lang.org)
- Release cycle: 1 year

Default Gems

- Upstream: Ruby core repository and GitHub
- Release cycle: 1year or maintainers matter

Bundled Gems

- Upstream: Only GitHub
- Release cycle: Maintainers matter



Pros/Cons of Gemification

Pros:

- Maintainers can release gem for bugfix, new feature independent with Ruby core.
- Easily back port stable version from develop version. Ruby users can use new feature on stable version.
- If upstream is available on GitHub, Ruby users easily send patch via Pull request.

Cons:

- Abandoned and complex dependency on rubygems and bundler
- Maintainers need to maintain ruby core and GitHub repositories both.

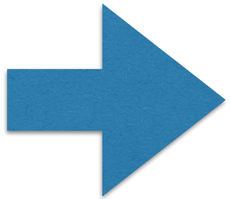


What number of these libraries - 2.4.0 to 2.5.0

In Ruby 2.4
Standard Libraries
- Pure ruby: 69
- Extensions: 23

Default gems
- Pure ruby: 1
- Extensions: 5

Bundled Gems
- Pure ruby: 7
- Extensions: 0



In Ruby 2.5
Standard Libraries
- Pure ruby: 64 (-5)
- Extensions: 14 (-9)

Default gems
- Pure ruby: 6 (+5)
- Extensions: 14 (+9)

Bundled Gems
- Pure ruby: 7
- Extensions: 0



Status of OpenSSL binding

- OpenSSL is already extracted default gems. You can update it separated ruby core releases same as rubygems, rdoc, bigdecimal.
- https://github.com/ruby/openssl
  - It's maintained by @rhe
  - Upstream was changed github repository from svn.ruby-lang.org
  - He aggressively maintains new feature of openssl



Concerns of default gems

Namespaces of standard library was reserved.

- https://rubygems.org/gems/fileutils
- https://rubygems.org/gems/webrick
- etc…

Some of gems like fileutils are completely different implementation from ruby standard library. If users invoke `gem install fileutils`, it broke their environment 1 week ago. (I resolve this)



fiddle

fiddle is standard library for wrapper of libffi. But fiddle was already reserved another implementation rubygems.org

https://github.com/bsm/fiddle

I did coordinate to transfer above namespace and override CRuby implementation now.

https://github.com/ruby/fiddle



gdbm 1.3.1

This library provides a gdbm library compatible with the MRI standard library, but using Ruby-FFI rather than a C extension. This allows gdbm to easily be used from alternative Ruby implementations, such as JRuby. It can also be used with MRI, if there is some kind of need for that.

VERSIONS:

1.3.1 - April 29, 2016 (9 KB)
1.3.0 - December 14, 2014 (9 KB)
1.2 - May 3, 2010 (12 KB)
1.1 - December 1, 2009 (12 KB)
1.0 - October 10, 2009 (12 KB)

Show all versions (6 total) →

AUTHORS:

Justin Collins

OWNERS:

SHA 256 CHECKSUM:

fad4bed21852e402fc37dbd6770c40cd12ccc2bd9cd8fbde07a9603b4925a72f

Star

TOTAL DOWNLOADS
21,259

FOR THIS VERSION
2,052

GEMFILE:

```
gem 'gdbm', '~> 1.3
```

INSTALL:

```
gem install gdbm
```

LICENSES:

N/A

REQUIRED RUBY VERSION:

>= 0

LINKS:

Homepage

Source Code

Documentation

What we will do?

I must offer to transfer ownership of these gems to rubygems.org.
But some of gems are masked status. As first, I need to un-mask it.

My plan for reserved gems:

- Transfer request to owners of reserved gems.
- Removed gemification gems from blacklist on rubygems.
- Override reserved gems by standard libraries.



Concerns of bundled gems(1)

Bundled gems couldn’t support cross compilation. Therefore, some extensions of default gems can’t escalate verification status bundled gems from default gems.

Ex. We need to make Date gem to bundled gem before extracted from ruby core.



Concerns of bundled gems(2)

We need test suite for bundled gem with ruby trunk.

PoC of unak: https://gist.github.com/unak/a80b03d9a33de59bedb52e2711410e0d

No one guarantee work bundled gems and develop version of ruby.



Ruby 3.0

Ruby 3x3

“At RubyConf 2015 in San Antonio, Yukihiro "Matz" Matsumoto announced Ruby 3x3. “

http://engineering.appfolio.com/appfolio-engineering/2015/11/18/ruby-3x3

Matz says requirements of Ruby 3 are following things:

- Performance Improvement(3x3, JIT, Concurrent GC or others)
- Concurrency
- Soft Typing



Current Status in Ruby 3x3

- Performance Improvement
  - JIT proposal named Rujit: https://bugs.ruby-lang.org/issues/12589
  - Deoptimization Engine: https://github.com/ruby/ruby/pull/1419
- Concurrency
  - Guild: http://www.atdot.net/~ko1/activities/2016_rubykaigi.pdf
- Soft Typing
  - It has no concept implementation and proposals.
  - (Because it is most valuable place in Ruby 3!)



Why use Ruby?

“Ruby is designed to make programmers happy.”