GMOペパボ

RubyGems 3 & 4

The Package Manager of the Ruby Language

Hiroshi SHIBATA / GMO Pepabo, Inc.

2018.06.29 Paris.rb Conf 2018

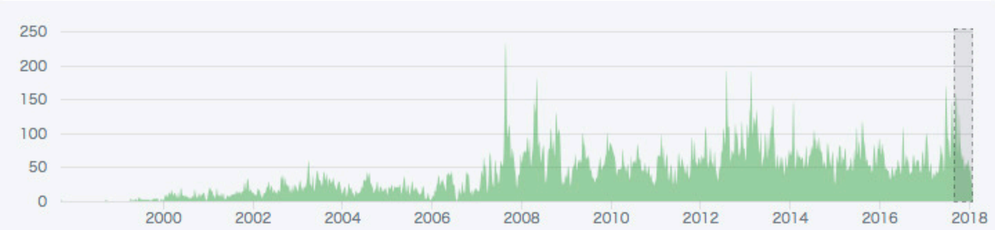
self.introduce

```
=> {
  name: “SHIBATA Hiroshi”,
  nickname: “hsbt”,
  organizations: [“ruby”, “rubygems”, “asakusarb”, “pepabo”, …],
  commit_bits: [“ruby”, “rake”, “rubygems”, “bundler”, “rdoc”,
“psych”,  “ruby-build”, “railsgirls”, “railsgirls-jp”, …],
  sites: [“hsbt.org”, “ruby-lang.org”, “rubyci.org”, “railsgirls.com”,
“railsgirls.jp”],
}
```

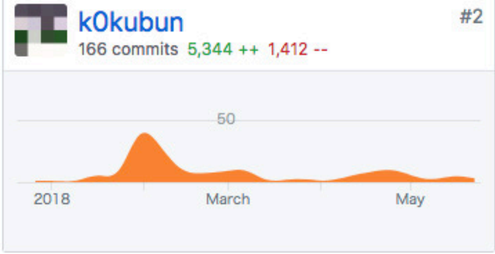


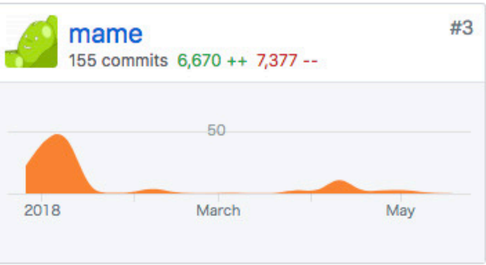
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network
- Forks

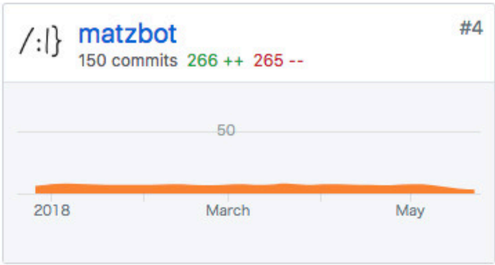
Contributions to trunk, excluding merge commits





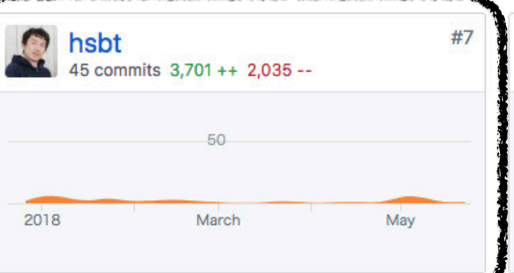


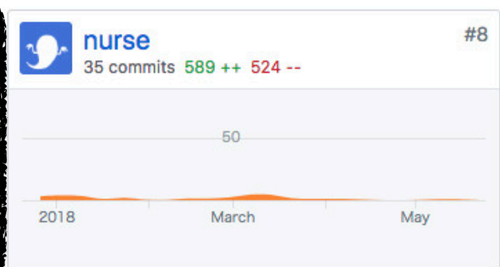












README
Imports 426
meetup 389
hanami 1
misc 1
000_Log 2
(no category) 5

README.md

Edit



RubyGems 2.7

Current status of RubyGems 2.7

- The maintenance policy of RubyGems(RG) is different as Ruby's maintenance policy. So, RubyGems team is different from Ruby core team.
- RG 2.5 and 2.6 are only security maintenance and no longer release separate from Ruby.
- RG 2.7 has bug and security fixes.



Surprisedly, RG 2.7 still supports Ruby 1.8.

```
~/D/g/r/rubygems (2.7) > rg respond_to
test/rubygems/test_gem_request_set_gem_dependency_api.rb
630:    tf.close! if tf.respond_to? :close!

test/rubygems/test_gem_source.rb
60:    response.uri = URI('http://example') if response.respond_to? :uri

test/rubygems/test_gem_package.rb
755:    tf.close! if tf.respond_to? :close!

test/rubygems/test_gem_util.rb
45:    if File.respond_to?(:realpath)

test/rubygems/test_gem_installer.rb
58:  str = str.dup.force_encoding("BINARY") if str.respond_to? :force_encoding
65:if Gem.respond_to?(:activate_bin_path)
893:    skip unless "".respond_to?(:force_encoding)

(snip)
```



Bundler Integration(rubygems.rb)

- Bundler was located rubygems repository as git submodule
- You can enabled it with `gem update —system`

```
if USE_BUNDLER_FOR_GEMDEPS

  ENV["BUNDLE_GEMFILE"] ||= File.expand_path(path)
  require 'rubygems/user_interaction'
  Gem::DefaultUserInteraction.use_ui(ui) do
    require "bundler"
    @gemdeps = Bundler.setup
    Bundler.ui = nil
    @gemdeps.requested_specs.map(&:to_spec).sort_by(&:name)
  end

else

  rs = Gem::RequestSet.new
  @gemdeps = rs.load_gemdeps path

  rs.resolve_current.map do |s|
    s.full_spec.tap(&:activate)
  end

end
```



Installer of Rubygems

- It’s provided by update_rubygems, setup.rb, setup_command.rb
- Installer of RubyGems promote Bundler to default gems.

```
  def install_default_bundler_gem
    return unless Gem::USE_BUNDLER_FOR_GEMDEPS

    specs_dir = Gem::Specification.default_specifications_dir
    specs_dir = File.join(options[:destdir], specs_dir) unless Gem.win_platform?
    mkdir_p specs_dir

(snip…)

    bundler_bin_dir = bundler_spec.bin_dir
    bundler_bin_dir = File.join(options[:destdir], bundler_bin_dir) unless Gem.win_platform?
    mkdir_p bundler_bin_dir
    bundler_spec.executables.each do |e|
      cp File.join("bundler", bundler_spec.bindir, e), File.join(bundler_bin_dir, e)
    end
```



The policy of RubyGems merging

- Merge latest stable version into Ruby Core
- Ruby 2.6.0 will bundle RubyGems 3.0(TBD)
- Ruby 2.7 or 3.0 will bundle RubyGems 4.0(TBD)



RubyGems 3.0

What’s new in RubyGems3?

- Removed deprecated methods.
- Removed to support for < Ruby 2.2.
- Added warnings of deprecated methods.
- To use release toolchain directly.



Introduction of Gem::Deprecate

- How to deprecate method in your code.

```
require 'rubygems/deprecate'

module Gem
(snip)
  def self.gunzip(data)
    Gem::Util.gunzip data
  end

  class << self
    extend Gem::Deprecate
    deprecate :gunzip, "Gem::Util.gunzip", 2018, 12
  end
(snip)
```

- You can show deprecated warning to end-users.

```
NOTE: Gem.gunzip is deprecated; use Gem::Util.gunzip instead. It will be removed on or after 2018-12-01.
Gem.gunzip called from /Users/hsbt/.rbenv/versions/2.6.0-dev/lib/ruby/gems/2.6.0/gems/rubygems-mirror-1.2.0/
lib/rubygems/mirror.rb:37.
```



Deprecate methods of platform

- We should care compatibility for new version of libraries.
- But rubygems is critical infrastructure in the Ruby ecosystem.
- I use akr/gem-codesearch and akr/all-ruby for investigating compatibility.



gem-codesearch

- akr/gem-codesearch and rubygems/rubygems-mirror
- We can search all of ruby code on rubygems.org with 80GB+ storage.

```
~/D/g/a/gem-codesearch (master) > gem i rubygems-mirror
~/D/g/a/gem-codesearch (master) > ulimit -n 8192; and env
RUBYGEMS_MIRROR_ONLY_LATEST=true rake mirror unpack; and find latest-gem -size +5M
-not -name "*.rb" -delete; and rake index
```

```
function gemsearch
  csearch $argv | sed 's/^\/Users\/hsbt\/Documents\/github\.com\/akr\/gem-
codesearch\/latest-gem\///g';
end
```



Demo

```
~ > gemsearch extension_build_error
rb2exe-0.3.0/bin/traveling-ruby-2.2.2/l32/lib/ruby/2.2.0/rubygems/installer.rb:
  def extension_build_error(build_dir, output, backtrace = nil) # :nodoc:
rb2exe-0.3.0/bin/traveling-ruby-2.2.2/l64/lib/ruby/2.2.0/rubygems/installer.rb:
  def extension_build_error(build_dir, output, backtrace = nil) # :nodoc:
rb2exe-0.3.0/bin/traveling-ruby-2.2.2/osx/lib/ruby/2.2.0/rubygems/installer.rb:
  def extension_build_error(build_dir, output, backtrace = nil) # :nodoc:
rb2exe-0.3.0/bin/traveling-ruby-2.2.2/win/lib/ruby/2.2.0/rubygems/installer.rb:
  def extension_build_error(build_dir, output, backtrace = nil) # :nodoc:
ruby-compiler-0.1.1/vendor/ruby/lib/rubygems/installer.rb:
  def extension_build_error(build_dir, output, backtrace = nil) # :nodoc:
rubygems-update-2.7.7/lib/rubygems/installer.rb:
  def extension_build_error(build_dir, output, backtrace = nil) # :nodoc:
rubygems-update-2.7.7/lib/rubygems/installer.rb:
  deprecate :extension_build_error, :none, 2018, 12
```



akr/all-ruby

- all-ruby provides the environment for all release versions of Ruby includes 0.59 to 2.6.0-preview2.
- It was Dockerized by shyouhei and hsbt.

https://hub.docker.com/r/rubylang/all-ruby/

```
alias all-ruby "docker run --rm -t rubylang/all-ruby /all-ruby/all-ruby"
```

- We can try to code block on all of versions of Ruby



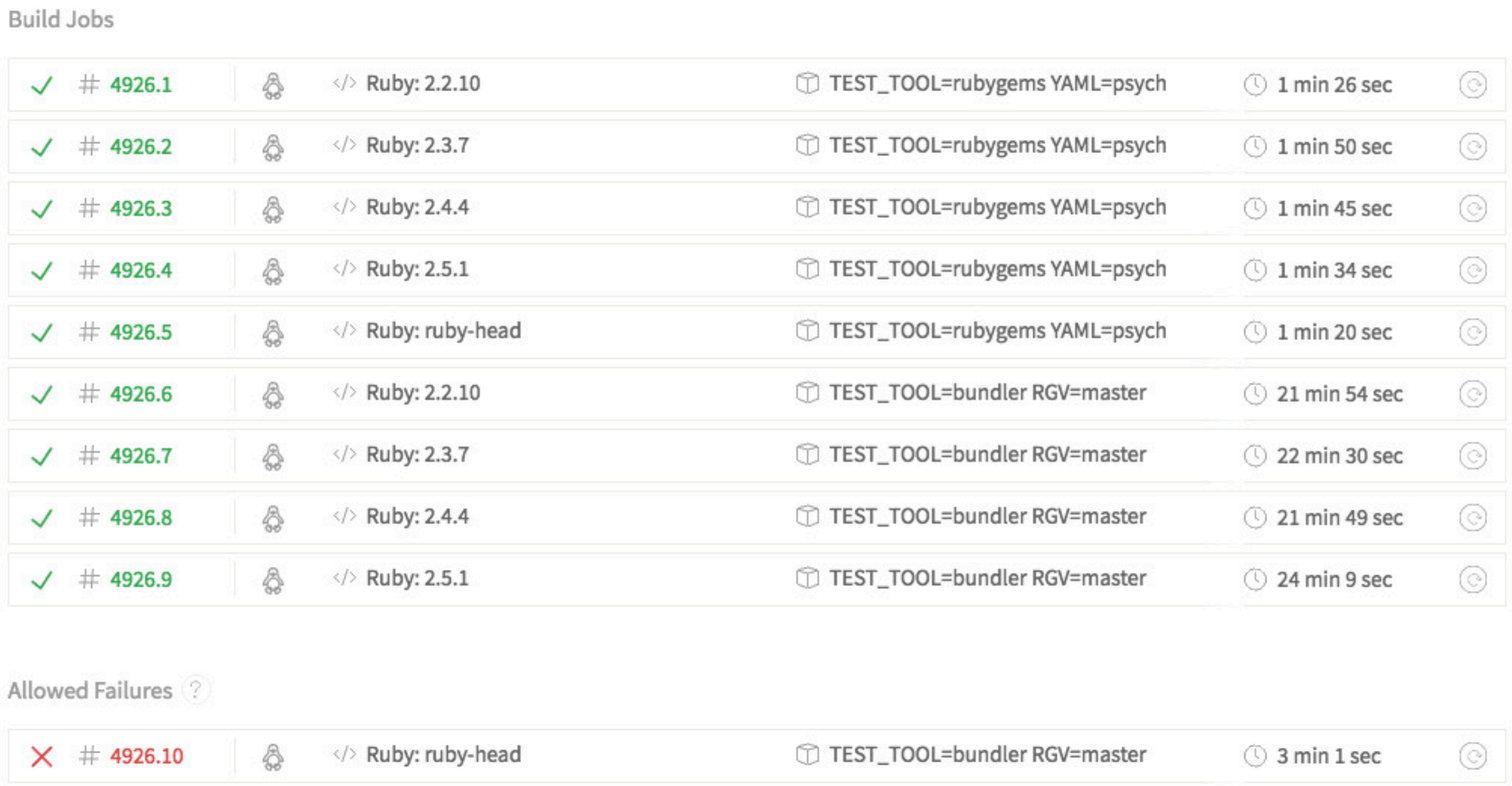
Demo

```
all-ruby -e "puts ‘foo'&.match?(/foo/)"
all-ruby -e "require 'tempfile'; p Tempfile.new.close!”
all-ruby -e "require 'io/console'; p IO.method_defined?(:noecho)”
all-ruby -e "require 'uri'; puts URI.const_defined?(:DEFAULT_PARSER)”
all-ruby -e "p [].respond_to? :flat_map"
```



Only support Ruby 2.2+

- We can use Keywords argument, Refinement, Other cool features in RubyGems now.
- Simple build matrix



Remove deprecated code

- RG have a lot of workarounds for old Ruby. They are branches like RUBY_VERSION, respond_to?, defined?

```
-  if [].respond_to? :flat_map
-    def pinned_requirement name # :nodoc:
-      requirement = Gem::Dependency.new name
-      specification = @set.sets.flat_map { |set|
-        set.find_all(requirement)
-      }.compact.first
+  def pinned_requirement name # :nodoc:
+    requirement = Gem::Dependency.new name
```



Code ceverage

- Generate test coverage with simplecov
- I found the needless code by coverage tool like rdoc workaround for old Ruby and delete it.



All Files (89.4% covered at 809.28 hits/line)

136 files in total. 9037 relevant lines. 8079 lines covered and 958 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	
lib/rubygems/resolver/local_specification.rb	25.0 %	42	20	5	15
lib/rubygems/commands/rdoc_command.rb	40.91 %	97	44	18	26
lib/rubygems/resolver/installed_specification.rb	45.45 %	59	22	10	12
lib/rubygems/commands/generate_index_command.rb	45.83 %	85	24	11	13
lib/rubygems/resolver/git_specification.rb	53.57 %	59	28	15	13
lib/rubygems/safe_yaml.rb	60.0 %	51	15	9	6
lib/rubygems/resolver/activation_request.rb	60.56 %	186	71	43	28
lib/rubygems/resolver/stats.rb	62.96 %	45	27	17	10
lib/rubygems/resolver/index_set.rb	64.86 %	81	37	24	13
lib/rubygems/syck_hack.rb	65.0 %	77	20	13	7
lib/rubygems/resolver/conflict.rb	66.04 %	160	53	35	18
lib/rubygems/gem_runner.rb	66.67 %	86	33	22	11
lib/rubygems/resolver/index_specification.rb	66.67 %	70	27	18	9
lib/rubygems/resolver/api_specification.rb	67.5 %	91	40	27	13
lib/rubygems/resolver/requirement_list.rb	68.97 %	82	29	20	9
lib/rubygems/resolver/lock_specification.rb	69.05 %	88	42	29	13
lib/rubygems/commands/check_command.rb	69.57 %	94	46	32	14
lib/rubygems/resolver/lock_set.rb	70.97 %	83	31	22	9
lib/rubygems/dependency_installer.rb	71.89 %	495	217	156	61
lib/rubygems/commands/search_command.rb	73.33 %	41	15	11	4
lib/rubygems/resolver/dependency_request.rb	73.68 %	120	38	28	10
lib/rubygems/server.rb	73.97 %	878	219	162	57
lib/rubygems/commands/setup_command.rb	75.32 %	629	316	238	78
lib/rubygems/resolver/vendor_set.rb	75.86 %	88	29	22	7
lib/rubygems/commands/signin_command.rb	78.57 %	33	14	11	3
lib/rubygems/request.rb	79.25 %	294	159	126	33
lib/rubygems/commands/server_command.rb	81.25 %	87	32	26	6
lib/rubygems/commands/signout_command.rb	81.25 %	33	16	13	3
lib/rubygems/source/specific_file.rb	81.48 %	73	27	22	5
lib/rubygems/commands/open_command.rb	81.58 %	81	38	31	7
lib/rubygems/available_set.rb	81.82 %	165	77	63	14
lib/rubygems/request_set.rb	82.16 %	452	213	175	38
lib/rubygems/commands/uninstall_command.rb	82.89 %	175	76	63	13
lib/rubygems/resolver/best_set.rb	83.87 %	79	31	26	5
lib/rubygems/resolver/source_set.rb	84.21 %	48	19	16	3
lib/rubygems/resolver/git_set.rb	84.78 %	123	46	39	7
lib/rubygems/test_utilities.rb	84.85 %	384	165	140	25
lib/rubygems/validator.rb	84.85 %	166	66	56	10
lib/rubygems/commands/fetch_command.rb	85.0 %	78	40	34	6
lib/rubygems/commands/specification_command.rb	85.29 %	146	68	58	10
lib/rubygems/dependency_list.rb	85.56 %	244	90	77	13
lib/rubygems/install_update_options.rb	85.9 %	216	78	67	11
lib/rubygems/source.rb	86.27 %	238	102	88	14
lib/rubygems/psych_tree.rb	86.67 %	32	15	13	2
lib/rubygems/commands/sources_command.rb	87.21 %	211	86	75	11
lib/rubygems/resolver/installer_set.rb	87.91 %	227	91	80	11
lib/rubygems/resolver/api_set.rb	88.46 %	126	52	46	6
lib/rubygems/commands/stale_command.rb	89.47 %	39	19	17	2
lib/rubygems/commands/help_command.rb	89.83 %	375	59	53	6
lib/rubygems/indexer.rb	90.0 %	435	200	180	20
lib/rubygems/commands/which_command.rb	90.24 %	91	41	37	4

Upgrade toolchain

- Removed hoe, This is same as rake, rdoc, psych
- Added preparation task without Bundler

```
desc "Setup Rubygems dev environment"
task :setup => ["bundler:checkout"] do
  gemspec = Gem::Specification.load(File.expand_path("../rubygems-update.gemspec", __FILE__))

  gemspec.dependencies.each do |dep|
    sh "gem install '#{dep.name}:#{dep.requirement.to_s}'"
  end
end
```



rubygems-update.gemspec

- How works with `gem update —system`?
- lib/rubygems/commands/update_command.rb

```
def execute
  if options[:system] then
    update_rubygems
    return
  end
(snip)
```

```
def update_rubygems
  check_update_arguments

  version, requirement = rubygems_target_version

  check_latest_rubygems version

  update_gem 'rubygems-update', version

(snip)

  install_rubygems version
end
```



pre-release rubygems-update.gem

- Usually, We should test gem with prerelease version before production release.
- But rubygems couldn’t do it.

```
diff --git lib/rubygems/specification.rb lib/rubygems/specification.rb
index 897ee85e97..29088372a6 100644
--- lib/rubygems/specification.rb
+++ lib/rubygems/specification.rb
@@ -2733,7 +2733,11 @@ class Gem::Specification < Gem::BasicSpecification

   def version= version
     @version = Gem::Version.create(version)
-    self.required_rubygems_version = '> 1.3.1' if @version.prerelease?
+    # skip to set required_ruby_version when pre-released rubygems.
+    # It caused to raise CircularDependencyError
+    if @version.prerelease? && @name.strip != "rubygems"
+      self.required_rubygems_version = '> 1.3.1'
+    end
     invalidate_memoized_attributes

     return @version
```



RubyGems 4.0

RubyGems 4

- It has imcompatible features.
  - Upgrade dependency resolver.
  - Make enable as default for conservative option.
  - Behaviour changes with default gems installer.
  - Executables in bin folder conflict with their gem versions.
  - Make ruby gem install to user-install by default.



Dependency Resolver incompatible

- RubyGems 2.x, 3.x uses Molinillo-0.5.7

```
~/D/g/r/rubygems (master) > ls lib/rubygems/resolver/molinillo/lib/molinillo
delegates           dependency_graph.rb gem_metadata.rb     resolution.rb       state.rb
dependency_graph    errors.rb           modules             resolver.rb
```

- Bundler 1.16.x also uses Molinillo-0.6.4

```
~/D/g/b/bundler (master) > ls lib/bundler/vendor/molinillo/lib/molinillo
compatibility.rb    dependency_graph    errors.rb           modules             resolver.rb
delegates           dependency_graph.rb gem_metadata.rb     resolution.rb       state.rb
```

- These are different versions and behavior of dependency resolver.



RubyGems 3 or 4

- I try to upgrade Molinillo-0.6.x on RubyGems.

https://github.com/rubygems/rubygems/pull/2026

```
~/D/g/r/rubygems (molinillo-0-6-3) > env GEMSRC_SKIP=true rake test
Gem::Indexer tests are being skipped.  Install builder gem.
Run options: --seed 2600

# Running:

.................................................................................................
...............................E.................................E.......................
......E.................E....E.....................F.........FF.F.FFFF....F...F.....F....................
.................................................................................................
.................................................................................E.........
.......................................S....................S.....................................
.................................................................................................
....
```

Make conservative option as default

- We got the installation time when already installed gems.

```
~ > gem i rails
      clone http://rubyonrails.org -> /Users/hsbt/Documents/rubyonrails.org
        git ls-remote http://rubyonrails.org
         hg identify http://rubyonrails.org
        svn info http://rubyonrails.org
      error Could not find version control system: http://rubyonrails.org
     exists /Users/hsbt/Documents/github.com/rails/rails
Successfully installed rails-5.2.0
1 gem installed
```

- To use conservative is ignore re-install action.

```
~ > gem i rails -conservative
~ >
```



Change behavior of default option

- `gem install --default` put gemspec into default gem directory.
- But it is not put and build the library files like .rb and native extensions.

```
~ > gem i csv --default
Fetching: csv-1.0.2.gem (100%)
Successfully installed csv-1.0.2 as a default gem
1 gem installed
~ > ls ~/.rbenv/versions/2.3.7/lib/ruby/gems/2.3.0/gems/csv-1.0.2/
~ >
```

- I hope to install completely library like csv-1.0.2 into the old Ruby versions.



Make `--user-install` as default

- Rubygems 4 will install the all gems to `~/.gem`
- Pros: Ruby in linux distribution has many of FAQ for gem installation for using `sudo`. This change resolve this issues.
- Cons: Ruby version manager like rbenv is not support it. And This is big incompatible feature.



Known Issue of rbenv with `gem I bundler`

```
~ > gem i bundler --conservative --user-install
Fetching: bundler-1.16.2.gem (100%)
WARNING:  You don't have /Users/hsbt/.gem/ruby/2.3.0/
bin in your PATH,
    gem executables will not run.
Successfully installed bundler-1.16.2
1 gem installed
~ > which -a bundle
/Users/hsbt/.rbenv/shims/bundle
~ > bundle -v
rbenv: bundle: command not found

The `bundle' command exists in these Ruby versions:
  2.5.1
  2.6.0-dev
  jruby-9.1.17.0

~ > .gem/ruby/2.3.0/bin/bundle -v
Bundler version 1.16.2
```



The location of execution wrapper

- Ruby core put executable script directly under the bin directory.
- We often faced conflict error when upgrading rdoc.

```
~ > gem update rdoc
Updating installed gems
Updating rdoc
Fetching: rdoc-6.0.4.gem (100%)
rdoc's executable "rdoc" conflicts with /Users/hsbt/.rbenv/versions/2.3.7/bin/rdoc
Overwrite the executable? [yN]  y
rdoc's executable "ri" conflicts with /Users/hsbt/.rbenv/versions/2.3.7/bin/ri
Overwrite the executable? [yN]  y
Successfully installed rdoc-6.0.4
Gems updated: rdoc
```

- You completely lost original exe wrapper.



What's happened?

- RubyGems generate wrapper script for executable script of gem

```
#!/Users/hsbt/.rbenv/versions/2.3.7/bin/ruby
#
# This file was generated by RubyGems.
#
# The application 'rdoc' is installed as part of a
# this file is here to facilitate running it.
#

require 'rubygems'

version = ">= 0.a"

if ARGV.first
  str = ARGV.first
  str = str.dup.force_encoding("BINARY") if str.re
  if str =~ /\A_(.*)_\z/ and Gem::Version.correct?
    version = $1
    ARGV.shift
  end
end

load Gem.bin_path('rdoc', 'rdoc', version)
```

```
#!/Users/hsbt/.rbenv/versions/2.6.0-dev/bin/ruby
#
#  RDoc: Documentation tool for source code
#        (see lib/rdoc/rdoc.rb for more information
#
#  Copyright (c) 2003 Dave Thomas
#  Released under the same terms as Ruby

begin
  gem 'rdoc'
rescue NameError => e # --disable-gems
  raise unless e.name == :gem
rescue Gem::LoadError
end

require 'rdoc/rdoc'

begin
  r = RDoc::RDoc.new
  r.document ARGV
rescue Errno::ENOSPC
```



GMOペパボ



Hiroshi SHIBATA @hsbt

Executive Officer CPO(Chief Productivity Officer)
Director of Business Process Re-engineering Office
Director of Technical Division(New!!1)
at GMO Pepabo, Inc. @pepabo

https://www.hsbt.org

Ruby is designed to make programmers happy.

Yukihiro Matz Matsumoto