GMOペパボ

The Future of Bundled Bundler

The first step of package management integration

Hiroshi SHIBATA / GMO Pepabo, Inc.

2019.04.20 RubyKaigi 2019

self.introduce

GMOペパボ



Hiroshi SHIBATA @hsbt

Executive Officer VP of Engineering
Technical Director
at GMO Pepabo, Inc. @pepabo

https://www.hsbt.org

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”],
}
```



```
Pepabo.engineers.select do |u|
  u.include?(RubyKaigi.attendees)
end
```

self.introduce

=> ["hsbt", "udzura", "pyama", "kunit", "daiki", "sangun", "yamoto", "akatsuura", "tokky", "hayapi", "kenken", "azuki", "tosite", "tsumichan", "ryoma", "kymmt", "amacou", "kenchan", "nishihata", "tanaken", "kurotaky", "shimoju", "tascript", “ogidow”, “kaya”]

We supported to 25 people for attending the RubyKaigi in this year.



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



About (English)

ruby / ruby

Unwatch 1,249 | Unstar 15,525 | Fork 4,129

Code | Pull requests 191 | Actions | Insights | Settings

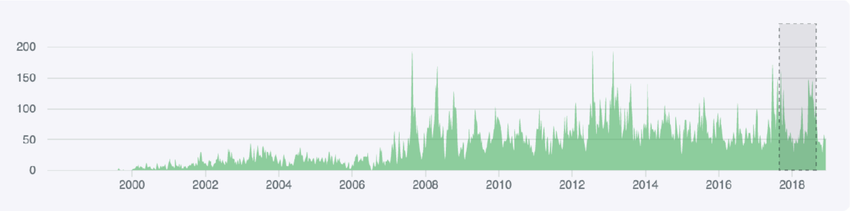
- Pulse
- Contributors
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Alerts
- Network
- Forks

- People

Dec 25, 2017 – Dec 25, 2018

Contributions: Commits

Contributions to trunk, excluding merge commits



















No.7

rubygems / rubygems

Unwatch 148 | Unstar 2,129 | Fork 1,046

Code | Issues 81 | Pull requests 29 | Actions | Insights | Settings

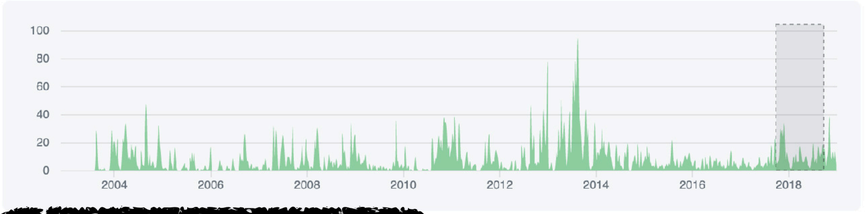
- Pulse
- **Contributors**
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Alerts
- Network
- Forks

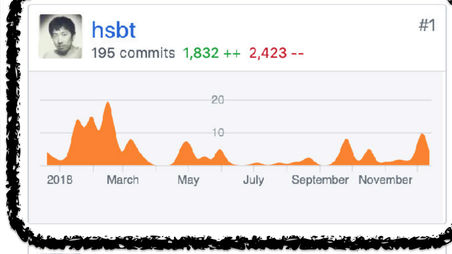
- People

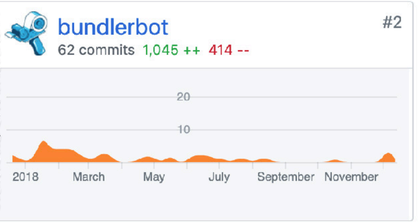
Dec 25, 2017 – Dec 25, 2018

Contributions: Commits

Contributions to master, excluding merge commits





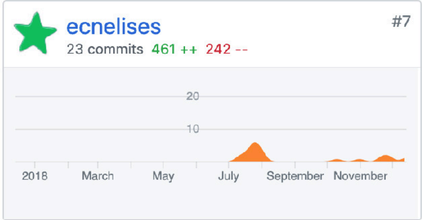














No.1

bundler / bundler

Unwatch 177 | Unstar 4,515 | Fork 2,028

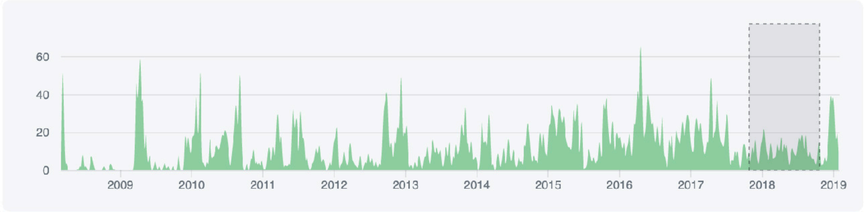
Code | Issues 175 | Pull requests 75 | Actions | Insights | Settings

- Pulse
- **Contributors**
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Alerts
- Network
- Forks

Dec 25, 2017 – Dec 25, 2018

Contributions: Commits

Contributions to master, excluding merge commits



bundlerbot #1
122 commits 3,998 ++ 727 --



agrim123 #2
71 commits 2,076 ++ 949 --



deivid-rodriguez #3
70 commits 1,125 ++ 674 --



colby-swandale #4
67 commits 613 ++ 213 --



hsbt #5
56 commits 656 ++ 704 --



segiddins #6
44 commits 1,103 ++ 387 --



indirect #7
34 commits 510 ++ 279 --



saveriomiroddi #8
15 commits 182 ++ 113 --



No.5

Agenda

- The Introduction of The Bundled Bundler
- What's happened in Ruby 2.6?
- BugMash after releasing Ruby 2.6
- The Challenge for Bundler Integration
- RubyGems 4.0
- The Roadmap for Ruby 3.0



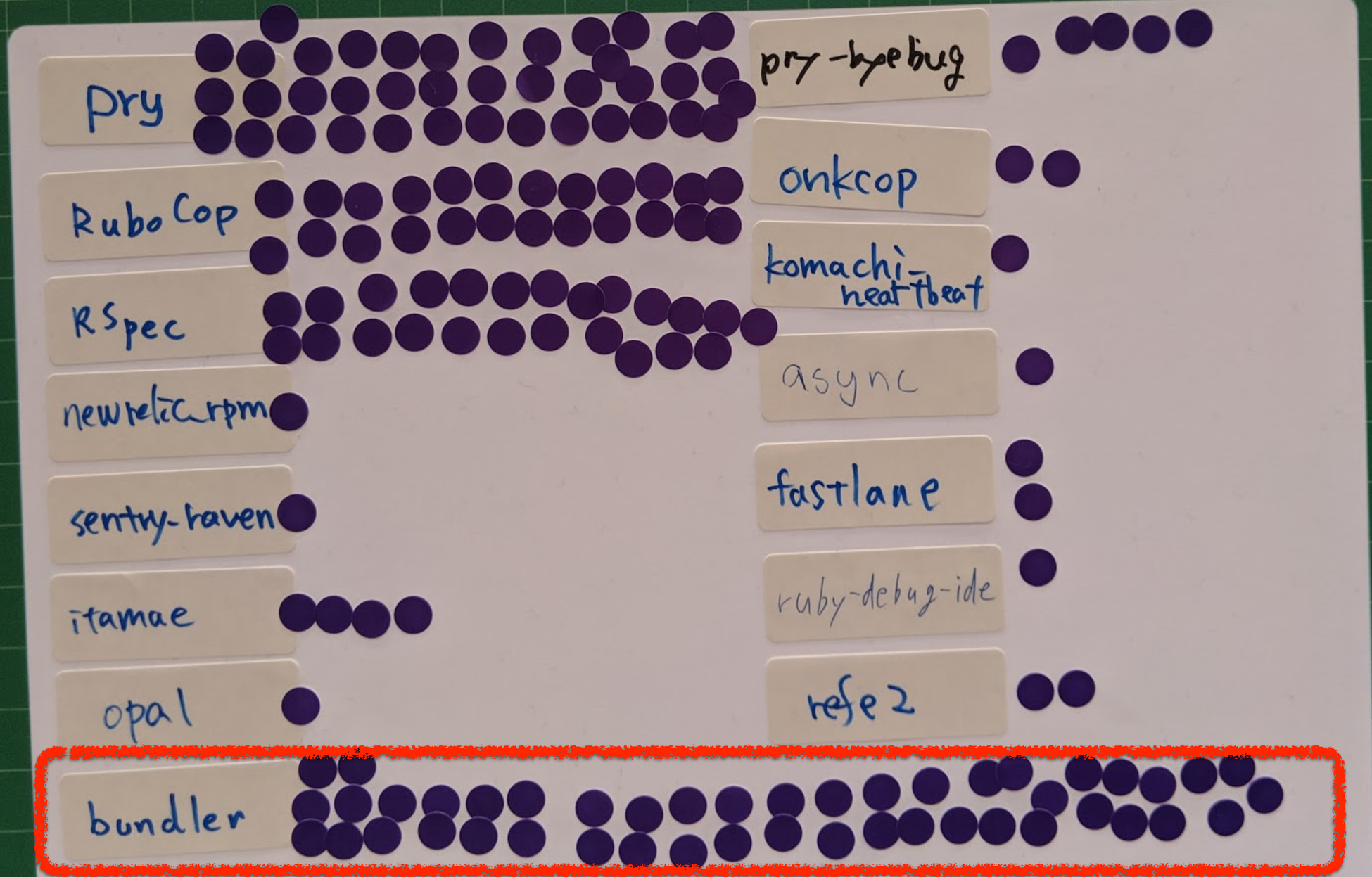
1.

The Introduction of The Bundled Bundler

Why should we integrate bundler to rubygems?

最初に入れるgemは？※railsを除く3つまで

What is the gem to install first? ※3 other than rails



RubyGems/Bundler integration in 2018

- We are working to integrate RubyGems and Bundler. But it’s no progress in the last year.
- RubyGems 3 drop to support under the Ruby 2.2.
- I’m working merging bundler into ruby core because Bundler 2 was released. Because Bundler 1.x still supports Ruby 1.8 and 1.9.



Bundler Integration on RubyGems 2.7

- It disabled in Ruby 2.5 because bundler is not part of standard library.
- You can enabled it with only `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(not Ruby)

- 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 current behavior of the bundled bundler

- It integrates with default gems like json, psych.
- The upstream is https://github.com/bundler/bundler. I backport the released/developed version to ruby repository.
- Ruby 2.6 always enabled Bundler gem_deps now(New!)

```
~ > gem list | rg default:
bigdecimal (1.4.3, default:
1.4.2)
bundler (default: 2.0.1, 1.17.3)
cmath (default: 1.0.0)
csv (3.0.6, default: 3.0.4)
(snip)
thwait (default: 0.1.0)
tracer (default: 0.1.0)
webrick (default: 1.4.2)
zlib (default: 1.0.0)
```



2.

What’s happened in Ruby 2.6?

What's new in RubyGems 3

- I released RubyGems 3 at 19 Dec 2018
  - https://blog.rubygems.org/2018/12/19/3.0.0-released.html
- This version dropped to support the old Ruby versions like 1.8 and 1.9
- RubyGems 3 have a lot of features and bugfixes.



The Important Notification

Do protect your account of RubyGems.org

Remove deprecated code

- RubyGems 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
```



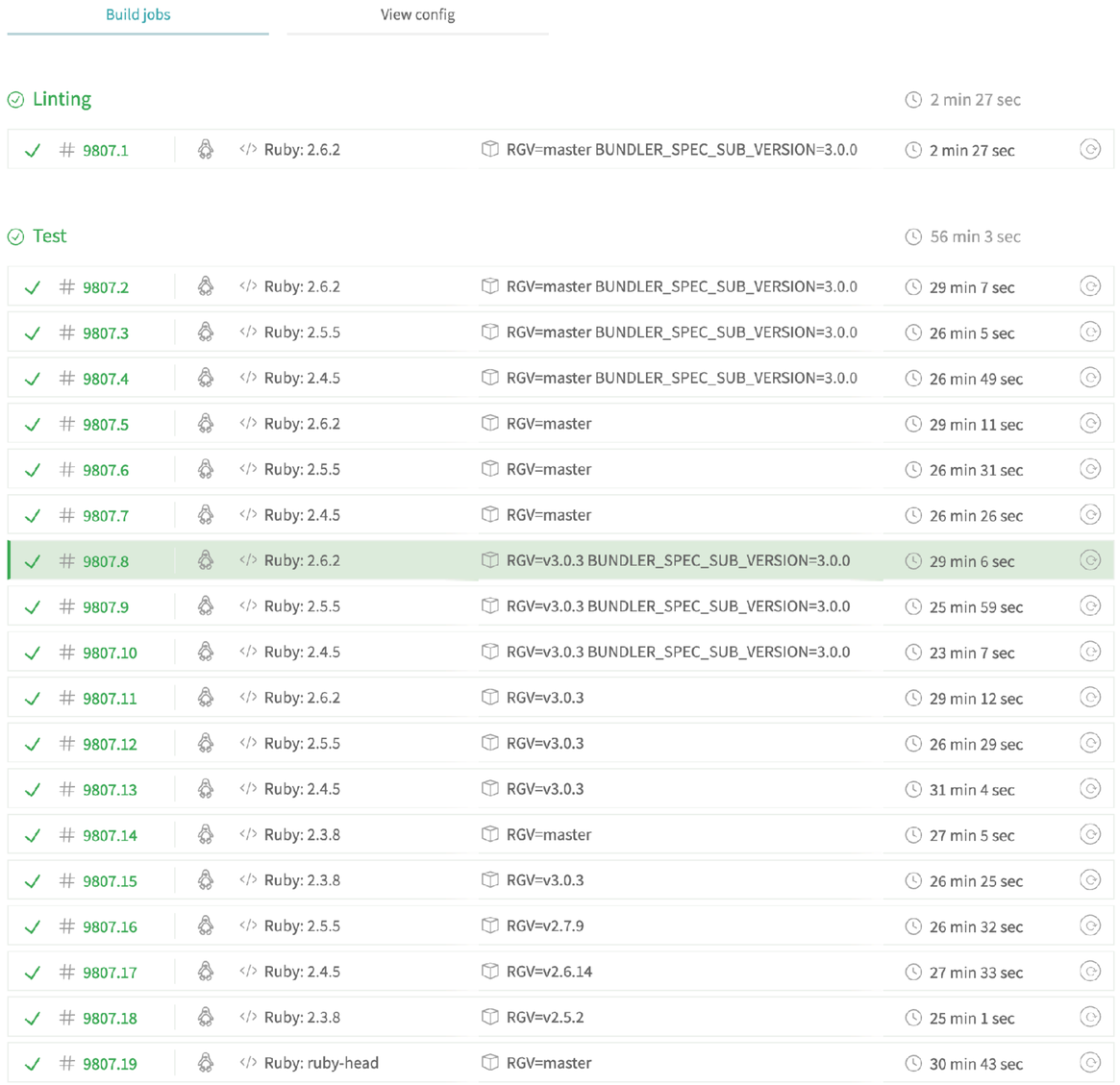
What’s new in Bundler 2?

- We disabled the incompatible features like renaming `gems.rb` from `Gemfile`
- They no longer support under the Ruby 2.2.
- There is no incompatible feature from Bundler 1.17.x.



Only support Ruby 2.2+

- We can use Keywords argument, Refinement, Other cool features in RubyGems and Bundler now.
- Finally, We got the simple build matrix



Introduce `make test-bundler`

```
when "bundler"
  `rm -rf lib/bundler* libexec/bundler libexec/bundle libexec/bundle_ruby \
   spec/bundler man/bundle* man/gemfile*`
  `cp -r ../../bundler/bundler/lib/bundler* ./lib`
  `cp -r ../../bundler/bundler/exe/bundle* ./libexec`
  `cp ../../bundler/bundler/bundler.gemspec ./lib/bundler`
  `cp -r ../../bundler/bundler/spec spec/bundler`
  `cp -r ../../bundler/bundler/man/*.{1,5,1\.txt,5\.txt,ronn} ./man`
  `rm -rf spec/bundler/support/artifice/vcr_cassettes`
```

- I added `test-prepare-bundler` for preparing to invoke rspec on ruby core repository. I put them into `.bundle` directory under the ruby repo and set it to `GEM_HOME` when running `make test-bundler`
- Now, We can invoke bundler examples with miniruby each commits.



The issues of bundler test suite.

- The Bundler examples is hard way.
  - The most of Bundler examples are integration test. Example for invoking to `bundle exec` command and assert standard output.

```
Finished in 52 minutes 54 seconds (files took 1.7 seconds to load)
2626 examples, 0 failures, 8 pending
```

- Finally, I added `ruby_core` filter into bundler examples. Because some of examples expect that installed ruby interpreter like `/usr/local/bin/ruby`



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
```

- When You put ‘y’, You completely lost original executable.



What's happened?

- RubyGems generate wrapper script for executable script of gem

```
#!/Users/hsbt/.rbenv/versions/2.6.0-dev/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)
```

Gem wrapper

```
#!/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
```

Original executable



Update BundlerVersionFinder

- BundlerVersionFinder was introduced at RubyGems 2.7
- It ability is the version detection by RubyGems strictly. Ex. 1.17.3 matches only 1.17.3.
- We update the filter condition. Now, 1.17.3 matches 1.x.y, 2.0.3 also matches 2.x.y.

```
def self.bundler_version_with_reason
  if v = ENV["BUNDLER_VERSION"]
    return [v, "`$BUNDLER_VERSION`"]
  end
  if v = bundle_update_bundler_version
    return if v == true
    return [v, "`bundle update --bundler`"]
  end
  v, lockfile = lockfile_version
  if v
    return [v, "your #{lockfile}"]
  end
end
```



Merge Bundler into ruby src

- I fixed the all of failures of Bundler examples with ruby core head.
- At first, I did merge Bundler 2.0 to Ruby core because Bundler 2 dropped to support the old versions of Ruby. But Bundler 2.0 causes the issues on Heroku platform.
- Heroku only support Bundler 1 at that time.
- I backport the patches to Bundler 1.17 from 2.0 and merge Bundler 1.17 into ruby core.



3.

BugMash
after releasing Ruby 2.6

20/40min

The issues after releasing Ruby 2.6

- Path Injection problem about rubylibdir to LOAD_PATH.
- The installer generate invalid gemspec of Bundler.
- The version switcher on Heroku.



The path injection for LOAD_PATH issue

- https://bugs.ruby-lang.org/issues/15469

```
- “/Users/user-name/.rbenv/versions/2.5.3/lib/ruby/gems/2.5.0/gems/bundler-1.17.2/lib"
- “/Users/user-name/.rbenv/rbenv.d/exec/gem-rehash”
- "/Users/user-name/temp/aiueo/vendor/bundle/ruby/2.5.0/gems/json-1.8.6/lib"
- (snip)
```

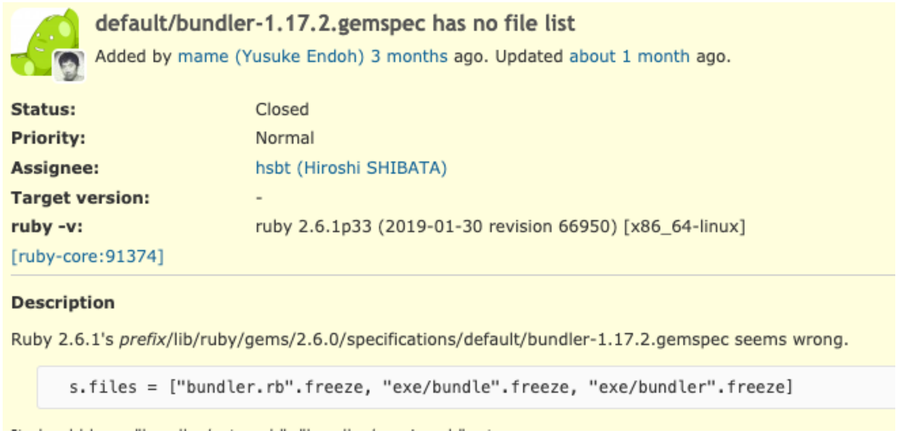
```
- "/Users/user-name/.rbenv/versions/2.6.0/lib/ruby/2.6.0"
- "/Users/user-name/.rbenv/rbenv.d/exec/gem-rehash"
- "/Users/user-name/temp/aiueo/vendor/bundle/ruby/2.6.0/gems/json-1.8.6/lib"
- (snip)
```

- After that, You can’t use the specified version of gems like json or psych. It activates the versions of default gems provided by ruby core.



The invalid gemspec generation issue

- The installer of ruby core causes to generate invalid gemspec when you install ruby core.
- https://bugs.ruby-lang.org/issues/15582



default/bundler-1.17.2.gemspec has no file list

Added by mame (Yusuke Endoh) 3 months ago. Updated about 1 month ago.

Status:	Closed
Priority:	Normal
Assignee:	hsbt (Hiroshi SHIBATA)
Target version:	-
ruby -v:	ruby 2.6.1p33 (2019-01-30 revision 66950) [x86_64-linux]

[ruby-core:91374]

Description

Ruby 2.6.1's *prefix*/lib/ruby/gems/2.6.0/specifications/default/bundler-1.17.2.gemspec seems wrong.

```
s.files = ["bundler.rb".freeze, "exe/bundle".freeze, "exe/bundler".freeze]
```

- Its file-list is the incomplete files by the bundled bundler. It activate the unexpected version of bundler with loading library.

The bundler switcher issue of Heroku

- https://github.com/heroku/heroku-buildpack-ruby/pull/850
- Heroku platform only uses version 1 of Bundler like 1.17.x. But Bundler version finder of RubyGems detects Bundler 1 or 2 from your Gemfile.lock. @schneems fixes this issue on heroku.

```
BLESSED_BUNDLER_VERSIONS = {}
BLESSED_BUNDLER_VERSIONS["1"] = "1.15.2"
BLESSED_BUNDLER_VERSIONS["2"] = "2.0.1"
```

- When You use Gemfile.lock updated by Bundler 2 with `bundle update --bundler`, Heroku reject your app. Now you can use Ruby 2.6 and Bundler 2 on heroku.



Ruby 2.6.3 is the best version ever

4.

The Challenge for Bundler Integration

Dependency Resolver incompatible

- RubyGems 2.x and 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.x and 2.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 3.1 with Molinillo-0.6.x

- 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.................................
.................................................................................................
...........
```

Duplicates the certificates

- **RubyGems and Bundler stored the duplicated certificates in your box. I fixed this at r67539**

```
~/D/g/r/rubygems (master) > fd . lib/rubygems/ssl_certs/
lib/rubygems/ssl_certs/index.rubygems.org
lib/rubygems/ssl_certs/index.rubygems.org/GlobalSignRootCA.pem
lib/rubygems/ssl_certs/rubygems.global.ssl.fastly.net
lib/rubygems/ssl_certs/rubygems.global.ssl.fastly.net/DigiCertHighAssuranceEVRootCA.pem
lib/rubygems/ssl_certs/rubygems.org
lib/rubygems/ssl_certs/rubygems.org/AddTrustExternalCARoot.pem
~/D/g/r/rubygems (master) > fd . bundler/lib/bundler/ssl_certs/
bundler/lib/bundler/ssl_certs/index.rubygems.org
bundler/lib/bundler/ssl_certs/index.rubygems.org/GlobalSignRootCA.pem
bundler/lib/bundler/ssl_certs/rubygems.global.ssl.fastly.net
bundler/lib/bundler/ssl_certs/rubygems.global.ssl.fastly.net/DigiCertHighAssuranceEVRootCA.pem
bundler/lib/bundler/ssl_certs/rubygems.org
bundler/lib/bundler/ssl_certs/rubygems.org/AddTrustExternalCARoot.pem
```



Fixed

Activation issues about default gems

- You couldn’t use the specified version of default gems like json when RubyGems/Bundler activated them.
- When rubygems uses json-2.1.0, You couldn’t use json 1.8.x. Because ruby gems and rubygems.org never uses JSON format.
- We can resolve it with `vendoring` approach. But json, psych, and openssl is C extension library.



Namespace?

- It may resolve with namespace feature with Ruby core.
  - https://bugs.ruby-lang.org/issues/10320
  - https://bugs.ruby-lang.org/issues/13847

```
require 'libfile', into: :Lib
```

- When rubygems used psych with shared library like libyaml-x.y.z, Ruby can’t resolve this problem.
- Please pick me with discuss about this if you have a nice idea.



5.

RubyGems 4.0

30/40min

RubyGems 4

- It has non-compatible features.
  - Make enable as default for conservative option: https://github.com/rubygems/rubygems/pull/2233
  - Behaviour changes with default gems installer: https://github.com/rubygems/rubygems/pull/2166
  - Make ruby gem install to user-install by default: https://github.com/rubygems/rubygems/issues/1394



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` maybe.
- 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 RubyGems have a lot of issues related this.



RubyGems still have a lot of issues

- When you share GEM_HOME in your box, You faced…
  - RubyGems always show the warnings for missing extension with platform mismatch. You always get the warnings with `jruby-lanucher`. (I fixed this in upstream)
  - RubyGems will activate the different platform with same version like nokogiri-1.10.1 and 1.10.1-java.
  - RubyGems will remove gem that was failed to `gem pristine`



We always welcome your patch.

6.

Roadmap for Ruby 3.0

35/40min

The policy of RubyGems merging

- The RubyGems accepts SemVer like versioning Policy.
- Merge latest stable version into Ruby Core
  - Ruby 2.6.0 bundled RubyGems 3.0
  - Ruby 2.7.0 will bundle RubyGems 3.1 or 3.2(TBD)
  - Ruby 3.0 will bundle RubyGems 4.0 or ?



Security release of RubyGems

- RubyGems have HackerOne.
  - 3 people handle vulnerability issues and will release RubyGems by SemVer like policy like “2.7.7” from “2.7.6”
- On the other hand, The Ruby core team will back port only vulnerability fixes by independent version like “2.6.5.1”, not “2.7.7”



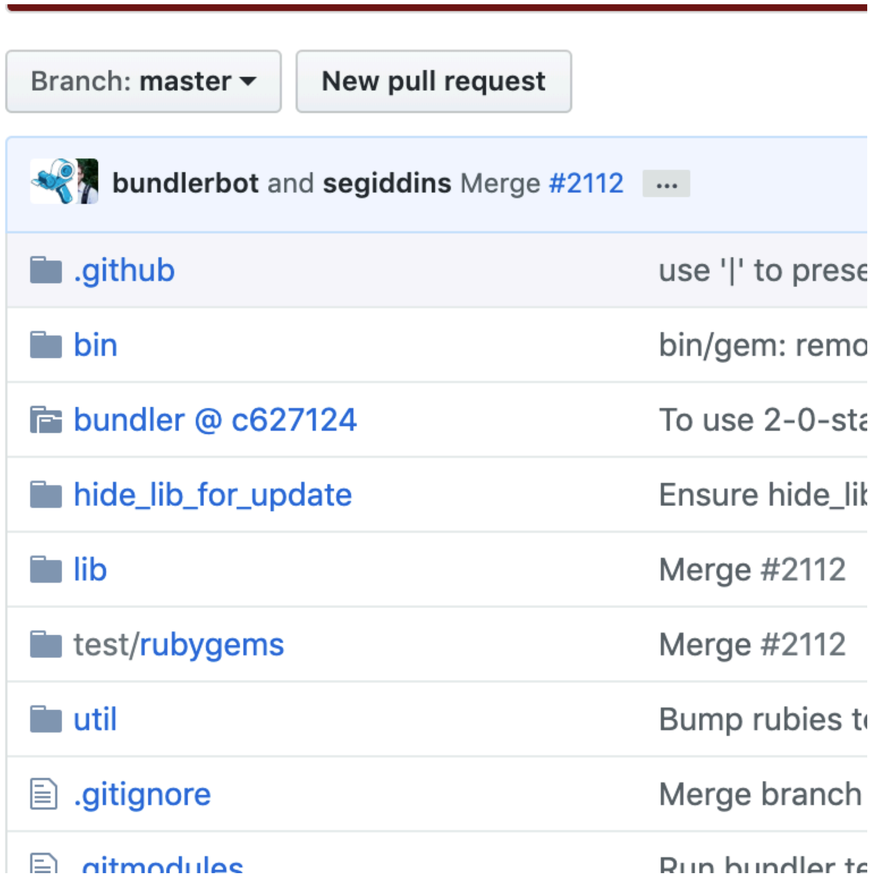
Support JRuby and TruffleRuby

- Surprisedly, RubyGems and Bundler never test JRuby and TruffleRuby in CI.
- We try to add JRuby and TruffleRuby to Travis or other CI environments.
- To JRuby and TruffleRuby tam: Please join us for this support.



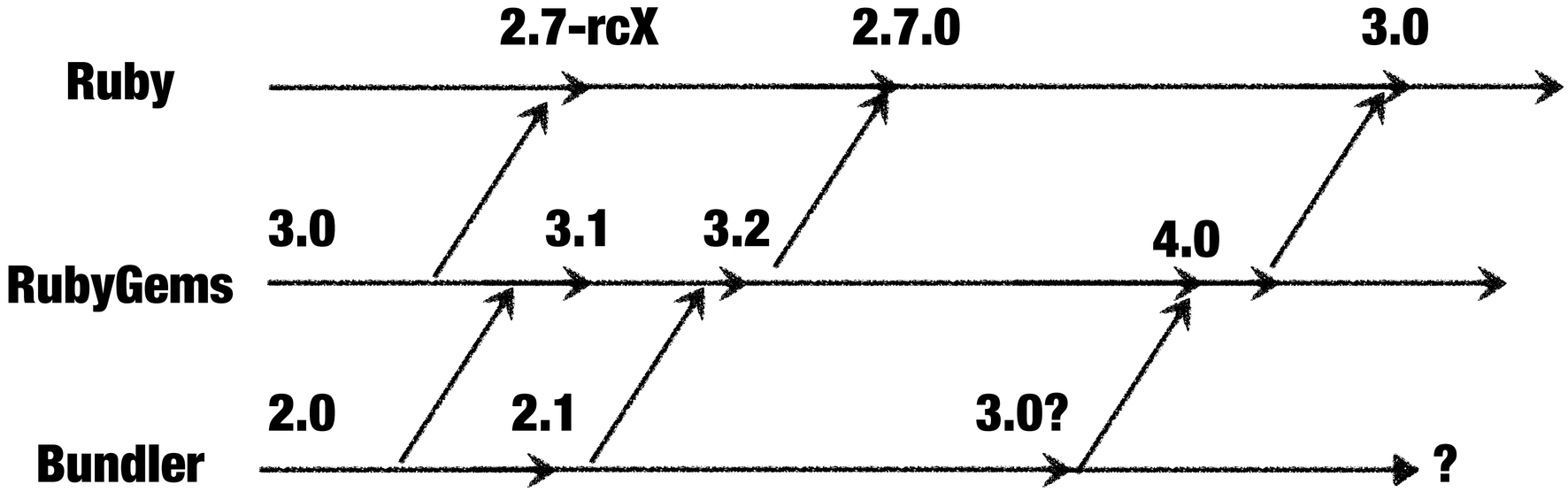
RubyGems/Bundler integration(1)

- Now, We put the bundler as submodule in rubygems repository.
- We will move the canonical repository of bundler to rubygems org or rubygems/rubygems.



RubyGems/Bundler integration(2)

- We will merge into RubyGems 3.2 and Bundler 2.1 into Ruby 2.7.0. After that, RubyGems 4.0 will be merge Ruby 3.



RubyGems/Bundler integration(3)

- Unify the duplicated code and configuration like the certificates.
- We have a plan to separate bundler-runtime and bundler-cli. After that, We will merge bundler-runtime into rubygems.
- I need to learn cargo and npm/yarn for the feature UI.



Rap-Up

- The Introduction of The Bundled Bundler
- What's happened in Ruby 2.6?
- BugMash after releasing Ruby 2.6
- The Challenge for Bundler Integration
- RubyGems 4.0
- Roadmap for Ruby 3.0



















