GMOペパボ

Dependency Resolution with Standard Libraries

The roadmap of RubyGems for Ruby 3.0

Hiroshi SHIBATA / GMO Pepabo, Inc.

2020.09.05 RubyKaigi takeout 2020

GMOペパボ



Hiroshi SHIBATA @hsbt

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

https://www.hsbt.org

Agenda

- The introduction for RubyGems 3.1 and Bundler 2.1
- Recent Update for RubyGems 3.2 and Bundler 2.2
- The feature plan and Roadmap for Ruby 3.0



The introduction for RubyGems 3.1 and Bundler 2.1

What's new in RubyGems 3.1?

RubyGems 3.1 has been released at 16 Dec, 2019

https://blog.rubygems.org/2019/12/16/3.1.0-released.html

The key features:

- **Add the multi-factor authentication for `gem yank` command**
- **Cleanup the compatibility code of Ruby 1.8 and 1.9**
- **Bundle Bundler 2.1**

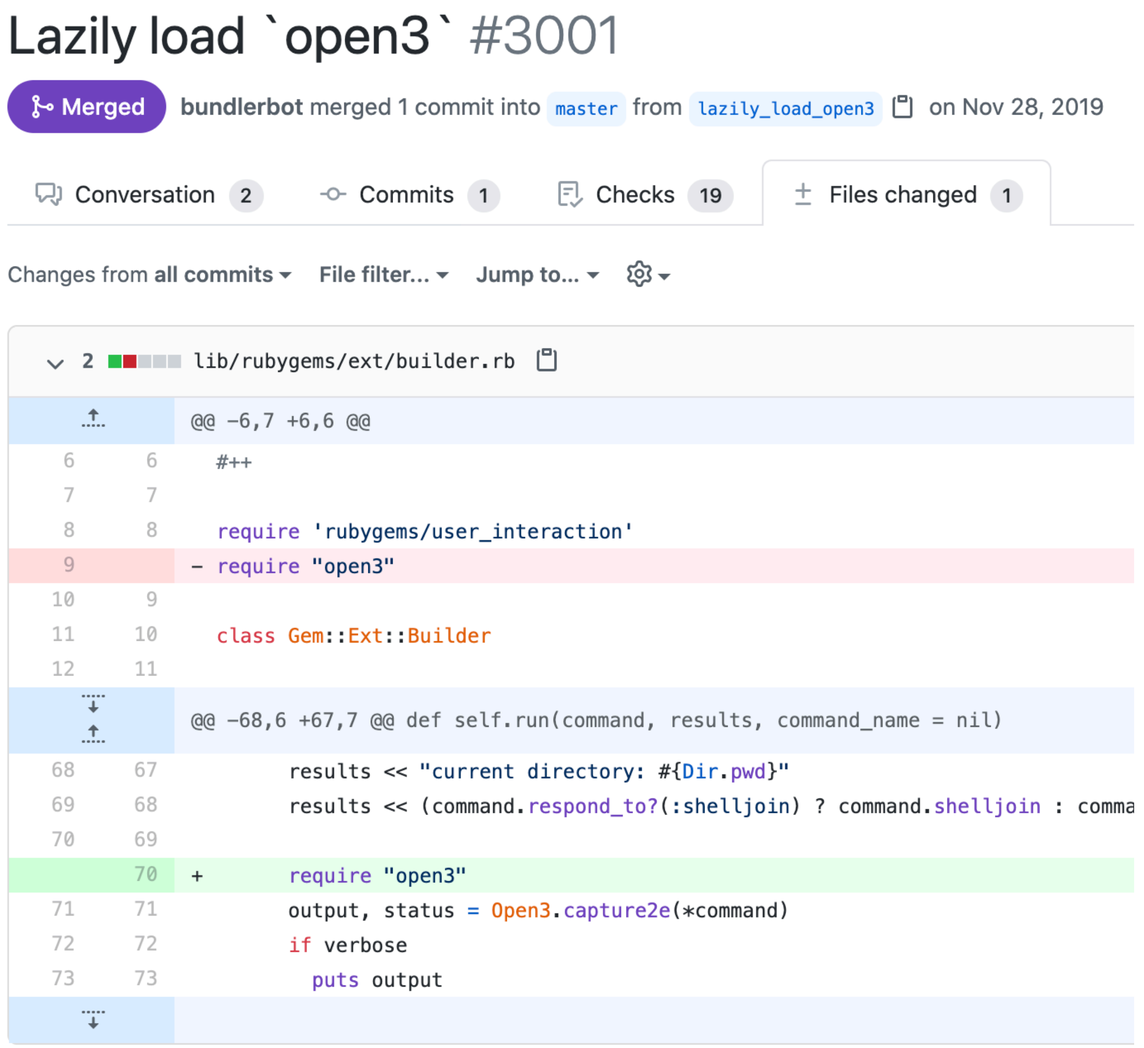


Lazily load the default gems

We make these library to lazily load in RubyGems.

- open3
- uri

It did prevent that you specify the several versions of the default gems.



What's new in Bundler 2.1?

Bundler 2.1 has been released at 15 Dec, 2019.

https://github.com/bundler/bundler/releases/tag/v2.1.4

The key features:

- **There is no incompatible feature from Bundler 2.0.x(maybe…)**
- **Cleanup the compatibility code of Ruby 1.8 and 1.9**
- **Tuned on the deprecation warnings for Bundler 3**



Tuned on the deprecation warnings for v3

```
Bundler::SharedHelpers.major_deprecation 2
```

- `bundle console`
- `bundle viz`
- `bundle inject`

- Bundler.environment → Bundler.load
- Bundler.clean_{env,exec} → Bundler.unbundled_{env,exec}
- Bundler.with_clean_env → Bundler.with_unbundled_env

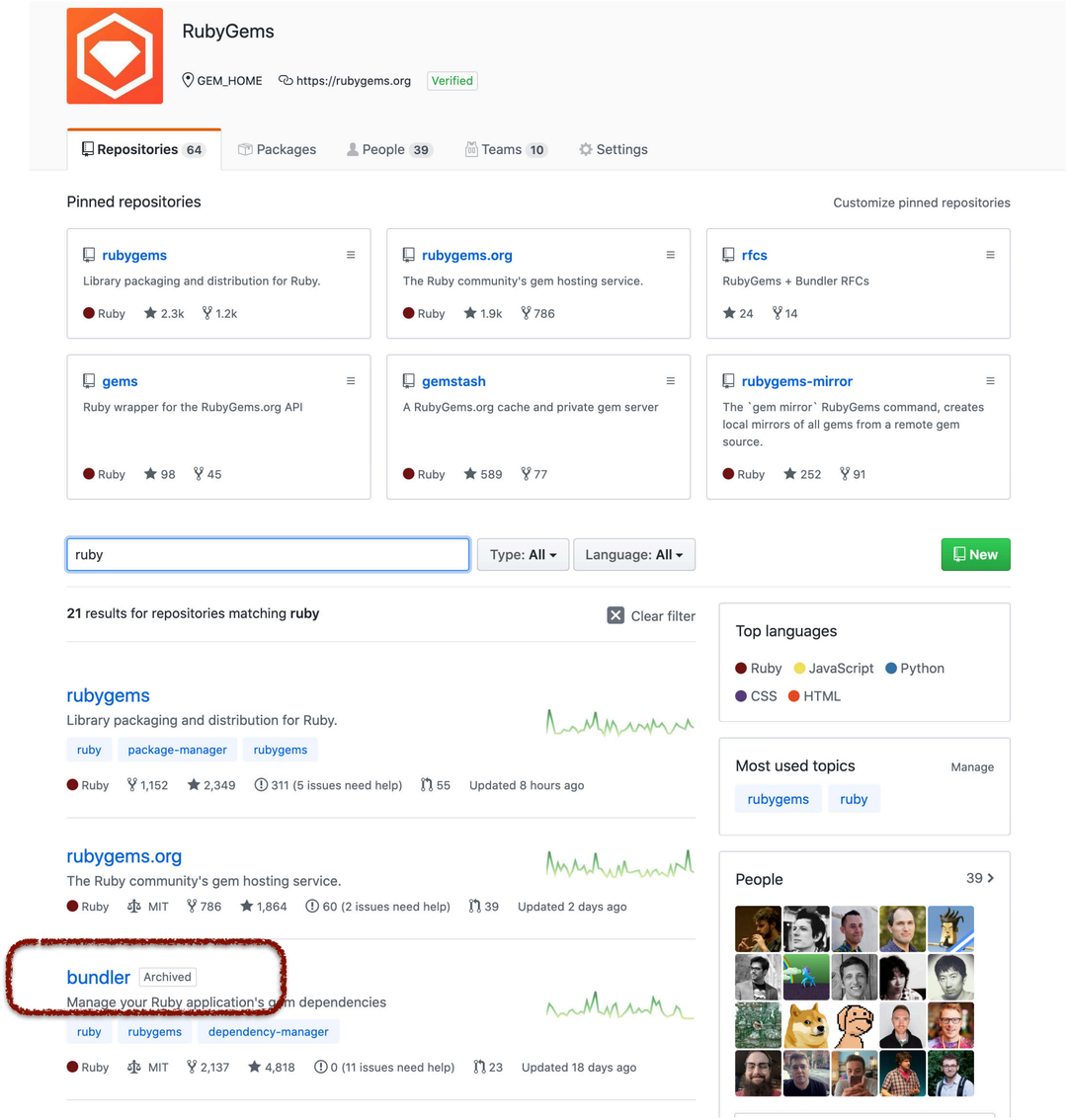


Recent Update for RubyGems 3.2 and Bundler 2.2

RubyGems/Bundler integration

We merged bundler org into rubygems org.

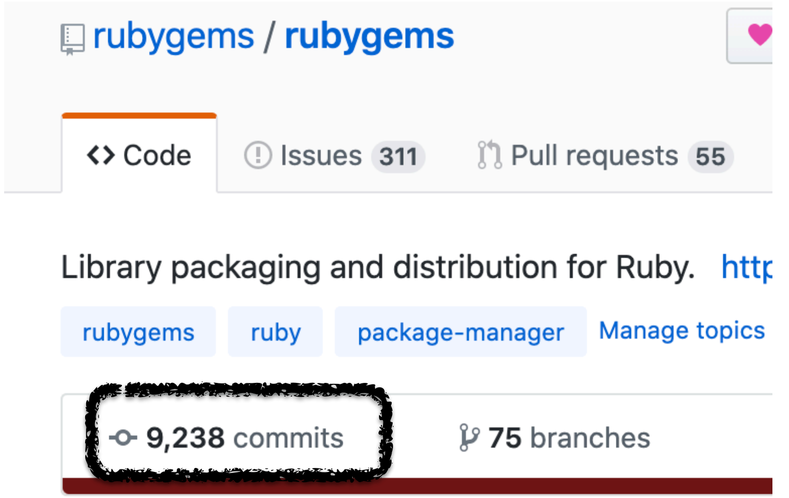
The repository, team, maintainers was merged.



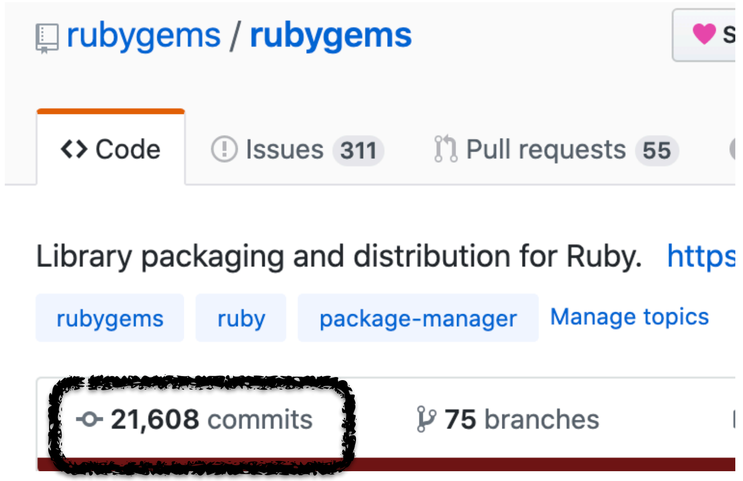
RubyGems/Bundler integration

- We also merged bundler repo into rubygems repo as monorepo.
- The commits, CI, mainainers documents was merged

Before



After



Recent Works

We still use separated release workflow about RubyGems and Bundler. We try to merge these toolchains at first.

- **Rubocop Rule** 🤖
- **Changelog(or History.txt)** 📚

Unify changelog handling for rubygems & bundler #3848

Open **deivid-rodriguez** wants to merge 17 commits into `master` from `rubygems_changelog`

Conversation 0 | Commits 17 | Checks 62 | Files changed 16

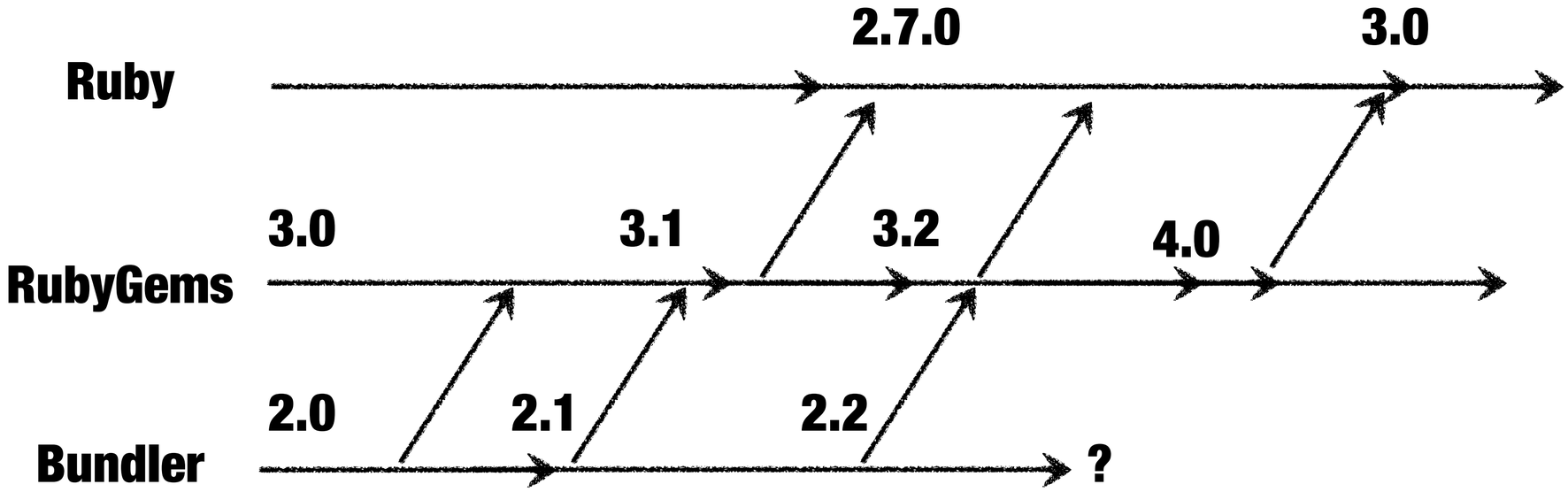
Changes from **all commits** | **File filter...** | **Jump to...**

26 .changelog.yml

```
@@ -0,0 +1,26 @@
+ ---
+
+ header_template: "=== %new_version / %release_date"
+
+ entry_template: "* %pull_request_title. Pull request #%pull_request_number by %pull_request_author"
+
+ release_date_format: "%Y-%m-%-d"
+
+ entry_wrapping: 74
+
+ changelog_label_mapping:
+   "rubygems: security fix": "Security fixes:"
+   "rubygems: breaking change": "Breaking changes:"
+   "rubygems: major enhancement": "Major enhancements:"
+   "rubygems: deprecation": "Deprecations:"
+   "rubygems: feature": "Features:"
+   "rubygems: performance": "Performance:"
+   "rubygems: documentation": "Documentation:"
+   "rubygems: minor enhancement": "Minor enhancements:"
+   "rubygems: bug fix": "Bug fixes:"
+   "rubygems: backport": null
+
+ patch_level_labels:
+   - "rubygems: security fix"
+   - "rubygems: minor enhancement"
+   - "rubygems: bug fix"
```

Bump up RubyGems/Bundler

We will merge into RubyGems 3.2 and Bundler 2.2 into Ruby 2.8.0-dev. After that, RubyGems 4.0 will be merge Ruby 3.



Gemification for Ruby 3

Gemification for Ruby 2.7

This matrix shows number of standard libraries and their classifications in Ruby 2.7.

	Standard Libraries	Default Gems	Bundled Gems
Pure Ruby	22	36	6
C extensions	12	17	0



Gemification for Ruby 2.8/3.0

Notable Changes

- `net-telnet` was removed from the bundled gems
- `rexml` and `rss` promoted the bundled gems from the default gems.

I'll work to publish the all of stdlib to the default gems

- Except to use `internal.h` or `internal/*.h` and depends on Ruby version
- Ex. mkmf.rb, objspace, pty, ripper, socket, coverage



Gamification on Ruby 3.0(TBD)

We will extract the standard libraries to the default gems.

```
Abbrev
Base64
DEBUGGER__
DRb
ERB
Find
OpenURI
OptionParser
PP
PrettyPrinter
resolv-replace.rb
Resolv
Rinda
SecureRandom
Set
Shellwords
Time
TSort
un.rb
Digest
IO(wait and nonblock)
Monitor(Ruby 2.6 version)
NKF
Pathname
Syslog
Win32API
WIN32OLE
```



Activation issues about the 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. Because Bundler still use the original implementation for YAMLSerializer.

master rubygems / bundler / lib / bundler / yaml_serializer.rb / <> Jump to

deivid-rodriguez Normalize bundler's access modifier indentations ...

3 contributors

89 lines (80 sloc) | 2.36 KB

```ruby
# frozen_string_literal: true

module Bundler
  # A stub yaml serializer that can handle only hashes and strings (as of now).
  module YAMLSerializer
    module_function

    def dump(hash)
      yaml = String.new("---")
      yaml << dump_hash(hash)
    end

    def dump_hash(hash)
      yaml = String.new("\n")
      hash.each do |k, v|
        yaml << k << ":"
        if v.is_a?(Hash)
          yaml << dump_hash(v).gsub(/^(?!$)/, "  ") # indent all non-empty lines
        elsif v.is_a?(Array) # Expected to be array of strings
          yaml << "\n- " << v.map {|s| s.to_s.gsub(/\s+/, " ").inspect }.join("\n- ") << "\n"
        else
          yaml << " " << v.to_s.gsub(/\s+/, " ").inspect << "\n"
        end
      end
      yaml
    end
```

What should we do?

1. Don’t use the default gems in `Bundler.setup`, `Kernel#require` and etc.

   → 😆

2. Vendoring approach when RubyGems uses only pure ruby libraries like URI.

   → 😩

3. Rewrite pure ruby code without the default gems.

   → 😩



Rewrite the default gems to Ruby code

We should rewrite the URI and FileUtils without the default gems.

Ex. Bundler uses FileUtils with the following methods.

- FileUtils.mkdir_p
- FileUtils.cp_r
- FileUtils.rm_rf
- FileUtils.touch



Gamification on Ruby 3.0(TBD)

We should keep the following gems with irb and rubygems. We will extract the default gems to the bundled gems without them.

```
did_you_mean
fileutils
forwardable
io/console
reline
readline-ext
delegate
pathname
tempfile
timeout
tmpdir
```

And also keep the following gems for `bundle exec`.

```
set
tsort
pathname
```



The major problem for the bundled gems

If you use Bundler, you need to add the bundled gems into your Gemfile.

```
source "https://rubygems.org"

git_source(:github) {|repo_name| "https://github.com/#{repo_name}" }

gem "english"
gem “uri”
gem “openssl”
…
```

I need to consider to transition and migration plan for this. But I have no idea yet. Maybe, I will add the some mechanism to Bundler internal to care about this.



Conclusion

Conclusion

- Finally, We will start to merge RubyGems and Bundler for unifying the one command.
- I’m working to extract stdlib to the default gems in Ruby3
- We should remove the current dependency from rubygems and bundler.

Let’s join to #rubygems in ruby-jp slack

