GMOペパボ

Gemification for Ruby 2.5/3.0

Future of Ruby standard libraries

SHIBATA Hiroshi / GMO Pepabo, inc.

2017.11.16 RubyConf 2017

self.introduce

self.introduce

```
=> {
  name: “SHIBATA Hiroshi”,
  nickname: “hsbt”,
  organizations: [“pepabo”, “ruby_core_team”, “asakusarb”],
  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



Ruby Sponsors

Sponsors of the Ruby language

- heroku:
  Unlimited dyno resources for websites
- fastly:
  OSS plan of CDN for *.ruby-lang.org
- NaCl:
  Network and Compute resources for website
- Microsoft:
  Provides Azure environment via MSDN Enterprise



Sponsors of the Ruby language(2)

- Ruby Association:
  Grant of development
- Nihon Ruby no Kai:
  macOS Server * 2
- Sugaya research laboratory:
  Network resources for macOS Server



1.

What’s a standard library?

What’s the Standard library?

- We called its “標準添付ライブラリ” in Japanese.
- It needs to `require` differently from embedded libraries like String, Thread, etc.
- It can be used without Bundler or RubyGems



Classification of standard libraries

This matrix shows number of standard libraries and their classifications in Ruby 2.4.

	Standard Libraries	Default Gems	Bundled Gems
Ruby	69	1	7
C	23	5	0



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(github.com/ruby)
  - Release cycle: 1year or maintainer’s convenience
- **Bundled Gems**
  - Upstream: Only GitHub
  - Release cycle: Maintainer’s convenience



What number of these libraries - 2.4.0 to 2.5.0

In Ruby 2.4

Standard Libraries

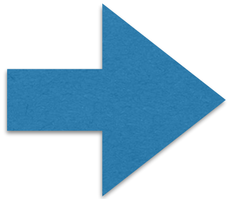
- **Pure ruby: 69**
- **Extensions: 23**

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**

2.

Default gems

Inside default gems

- `tool/rbinstall.rb` put gemspec files for default gems on Ruby core repository.
- We can release default gems to the rubygems.org. It's a Standard library that seems to be installed as a gem.
- Rubygems have a detection method for default gems.

```
>> Gem.loaded_specs["did_you_mean"].default_gem?
=> false
>> require 'openssl'
=> true
>> Gem.loaded_specs["openssl"].default_gem?
=> true
```



ruby/openssl

Status of OpenSSL binding

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



Security release

Releasing a new security version of Ruby is hard. All release maintainer are volunteers.

But If your Ruby supports Default gems/Bundled gems, You can upgrade these gems without upgrading ruby interpreter.

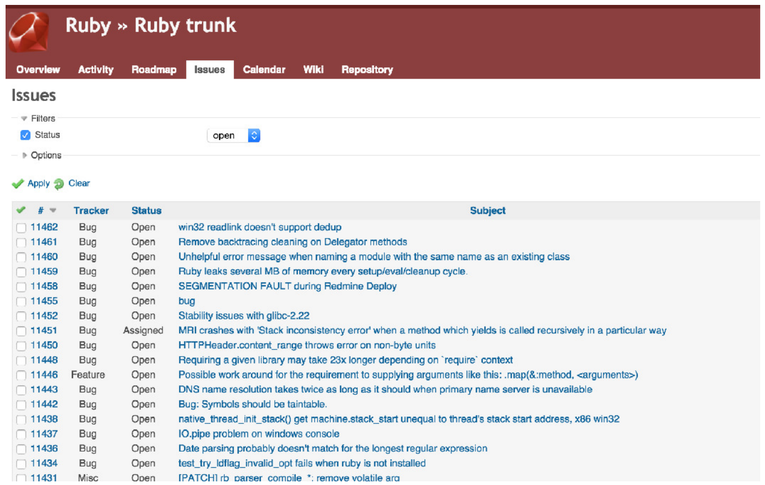
Upgrading Ruby is hard. Upgrading Gems is easy(er)



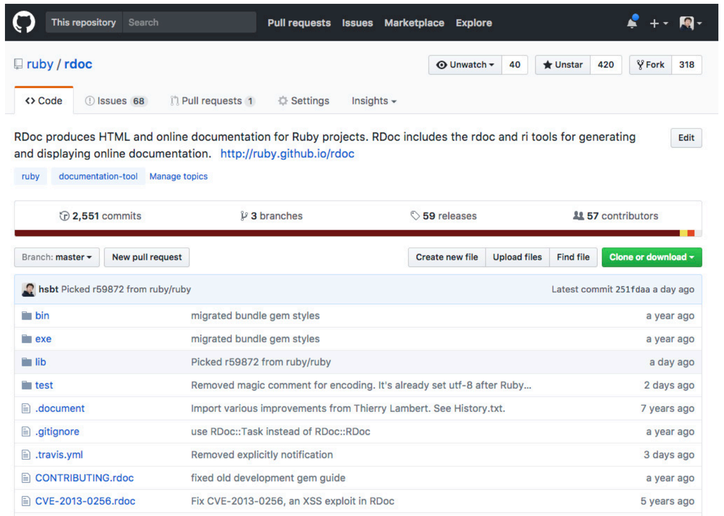
ruby/psych
and
ruby/rdoc

Current status of psych and rdoc

I moved canonical repository under the ruby organization. Our official tracker named “bugs.ruby-lang.org” is high threshold. But GitHub resolve this by pull requests.



VS



3.

Bundled gems

Inside Bundled gems

- We bundled *.gem and unpacked files to tarball package for Bundled gems.
- `make install` installed Bundled gem your box.

- It was installed by `tool/rbinstall.rb` on Ruby core repository.
- You can see list of Bundled gems via `gems/bundled_gems`



List of bundled gems on Ruby 2.5

- did_you_mean
- test-unit
- minitest
- net-telnet
- power_assert
- rake
- xmlrpc



Cross-compilation feature

We still have these concerns.

- **We couldn’t support to compile C extension on Bundled gems.**
- **Bundled gems couldn’t support cross compilation.**

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



We can test to bundled gems now

I created task for bundled gems named `test-bundled-gems`

```
~/D/g/r/ruby.trunk (trunk) > mk -C .x86_64-darwin test-bundled-gems
updating did_you_mean ...
HEAD is now at 982b11a... Version bump to 1.1.2
(snip)
updating xmlrpc ...
HEAD is now at aa29de3... bump version to 0.3.0
./miniruby -I../lib -I. -I.ext/common  ../tool/runruby.rb --extout=.ext  -- --disable-gems -C ".." bin/gem install --no-ri
--no-rdoc \
    --install-dir .bundle --conservative "minitest:~> 5" 'test-unit' 'rake' 'hoe' 'yard' 'pry' 'packnga'
testing did_you_mean gem
(snip)
DidYouMean version: 1.1.2
Run options: --seed 9770

# Running:

....................................

Fabulous run in 0.043525s, 804.1356 runs/s, 3009.7645 assertions/s.

35 runs, 131 assertions, 0 failures, 0 errors, 0 skips
…
```



4.

What’s Gemification?

Gemification for standard library

https://bugs.ruby-lang.org/issues/5481

- We extracted stdlibs like net-telnet, xmlrpc, rake 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.



Pros of Gemification(1)

- Maintainers can release gem for bugfix, new feature independent with Ruby core.
- Easily backport stable version from develop version. Ruby users can use new feature on stable version.



Pros of Gemification(2)

- If upstream is available on GitHub, Ruby users easily send patch via Pull request.
- Ruby interpreter developper can concentrate development Ruby internal.



Cons of Gemification(1)

- Abandoned and complex dependency on rubygems and bundler.
- Maintainers need to maintain ruby core and GitHub repositories both.
- It’s hard to maintain compatibility.



Cons of Gemification(2)

Backport is hard. Rubygems still supports Ruby 1.8.

```
% g show a34fb569e41cd87866e644d92a9df4be89b3cad2 test/rubygems/test_gem_package.r
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
```



5.

RubyGems and Gemification for Ruby 2.5

What’s rubygems

RubyGems is a package management framework for Ruby.

- **rubygems/rubygems.org:**
  - **The Ruby community's gem host.**
  - **rubygems.org is maintain by infrastructure team of rubygems. It is different team from rubygems cli team.**

- **rubygems/rubygems:**
  - **Command line tool of rubygems**
  - **Rubygems are created by Seattle.rb**



Reserved words on rubygems.org

Rubygems block reserved name same as standard libraries.

Ref. https://github.com/rubygems/rubygems.org/blob/master/lib/patterns.rb#L10



Problem of fileutils

You can see: https://rubygems.org/gems/fileutils/versions/0.7

If you install fileutils-0.7, fileutils-0.7 break your ruby environment.

Why???



fiddle

fiddle is standard library for wrapper of libffi. But fiddle was already reserved another implementation(https://github.com/bsm/fiddle) on rubygems.org

I did coordinate to transfer above namespace and override CRuby implementation(https://github.com/ruby/fiddle) now.



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 offered to transfer ownership of these gems to rubygems.org and overwrite it used by ruby stdlib.

My activities for reserved gems in this summer:

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



Default gems on Ruby 2.5

Current status of Default gems.

```
bigdecimal (default: 1.3.2)
bundler (default: 1.16.0)
cmath (default: 0.0.1)
csv (default: 0.1.0)
date (default: 0.0.1)
dbm (default: 1.0.0.beta1)
digest (default: 0.1.0)
etc (default: 0.2.1)
fcntl (default: 0.0.1)
fiddle (default: 1.0.0.beta2)
fileutils (default: 0.7.2)
gdbm (default: 2.0.0.beta1)
io-console (default: 0.4.6)
ipaddr (default: 1.0.0)
json (default: 2.1.0)
openssl (default: 2.1.0.beta1)
psych (default: 3.0.0.beta3)
rdoc (default: 6.0.0.beta2, 5.1.0)
scanf (default: 0.0.1)
sdbm (default: 0.0.1)
stringio (default: 0.0.1)
strscan (default: 0.0.1)
webrick (default: 1.4.0.beta1)
zlib (default: 0.1.0)
```

I'm going to promote following libraries to default gem at Ruby 2.5.0 or 2.6.0

- matrix
- digest
- ostruct
- stringio
- logger



6.

Gamification for Ruby 3.0

Reducing Ruby package size

- In Ruby 2.5, We added “bundler” to default gems.
- Bundler will be integrated RubyGems until Ruby 3.0 releasing.
- I will promote all of standard libraries to default gems.
- I will promote all of default gems without Rubygems dependencies to bundled gems.



Bundle Bundler to Ruby core

- Bundler uses rspec.
- I made `test-bundler` task that is invoking rspec examples of Bundler.

```
~/D/r/trunk > mk test-bundler
./miniruby -I../../github.com/ruby/ruby/lib -I. -I.ext/common  ../../github.com/ruby/ruby/tool/runruby.rb
--extout=.ext  -- --disable-gems -C "../../github.com/ruby/ruby" bin/gem install --no-ri --no-rdoc \
    --install-dir spec/rspec --conservative 'rspec:~> 3.5'
Run options: exclude {:ruby_repo=>true, :rubygems_master=>true, :git=>"=< 2.14.1", :rubygems=>"=< 2.6.13",
:ruby=>"=< 2.5.0", :realworld=>true, :sudo=>true}
.....................................................................................
.....................................................................................
.....................
```



Migration status of Rubygems and Bundler

- rubygems-2.7.0 partly uses bundler feature now.
- After bundler-2.0 released, We will target rubygems-3.0
- Ruby 2.5.0 have rubygems(-2.7.2) and bundler(-1.16.0), It helps migration plan of rubygems/bundler.

We can get Ruby 3.0 and Rubygems 3.0 and Bundler 3.0 in 2020. So It's a Ruby 3x3x3.



Activated problem for Rubygems

- **Rubygems have a activated problem with default gems.**
- **https://bugs.ruby-lang.org/issues/13847**

```
$ ruby -ropenssl -e 'p OpenSSL::VERSION'
"2.0.5"

$ cat Gemfile
# frozen_string_literal: true
source "https://rubygems.org"

gem 'openssl', '2.0.4'

$ bundle exec ruby -ropenssl -e 'p OpenSSL::VERSION'
/path/to/2.4.1/lib/ruby/gems/2.4.0/gems/bundler-1.15.4/lib/bundler/runtime.rb:317:in
`check_for_activated_spec!': You have already activated openssl 2.0.5, but your Gemfile
requires openssl 2.0.4. Prepending `bundle exec` to your command may solve this.
(Gem::LoadError)
```



require :into

https://bugs.ruby-lang.org/issues/10320

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

Matz has some concerns about ruby internal.

- Conflicts versions of a shared library like libyaml-., libssl-, libffi, etc...
- Order of LOADED_FEATURE

So, This feature is difficult to implement on current ruby specification now.



What are dependencies of Rubygems?

base64
benchmark
cgi
digest
English
erb
fileutils
find
io/console
monitor
net/http
net/https
openssl
optparse
pathname
pp
rbconfig
resolv
set
shellwords
socket
stringio
strscan
tempfile
thread
time
timeout
tmpdir
tsort
uri
webrick
Win32API
zlib



Does Gemification make happy Rubyists?



- We can upgrade standard library without language upgrade.
- Ruby developer uses GitHub workflow for default/ bundled gems.



- It’s heavy to maintain like github triage, gem release, resolve dependency.



Conclusion

Summary of Today's talk

- I described about a standard libraries of the Ruby language.
- I introduced specification of default gems/bundled gems of Ruby.
- I shared current status of Rubygems and Gemification projects.
- I shared my plan towards Ruby 3.0.



GMOペパボ



Hiroshi SHIBATA @hsbt

https://www.hsbt.org

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

Towards Ruby 3.0

