Ruby master - Feature #17873
Update of default gems in Ruby 3.1
05/20/2021 10:45 AM - hsbt (Hiroshi SHIBATA)

Status: Assigned
Priority: Normal
Assignee: hsbt (Hiroshi SHIBATA)
Target version:

**Description**

I promoted many standard libraries to the default gems. You can see them at [https://stdgems.org/#default-gems-ruby-301](https://stdgems.org/#default-gems-ruby-301).

I propose the following libraries to promote the bundled gems from the default gems.

- `matrix`
- `prime`
- `net/ftp`
- `net/smtp`
- `net/pop`
- `net/imap`

They have primary maintainer in the canonical repository in github. So, they can release the new version with their convenience.

*shugo (Shugo Maeda)*
*marcandre (Marc-Andre Lafortune)*

Any thought?

And I also propose to extract the following libraries from the default gems.

- `tracer`
- `dbm`
- `gdbm`

They are no longer actively maintained and not widely used today.

**History**

#1 - 05/20/2021 10:45 AM - hsbt (Hiroshi SHIBATA)
- Backport deleted (2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN)
- Tracker changed from Bug to Feature

#2 - 05/20/2021 11:01 AM - Eregon (Benoit Daloze)

And I also propose to extract the following libraries from the default gems.

Those are already default gems.
So I guess you mean no longer including them with Ruby, and gem install NAME would be needed to use them?

Seems fine in general.

I’m hesitant about fontl since it’s so small (just a bunch of constants) and I think it doesn’t make so much sense to have that as a separate gem (it’s already a default gem, though I don’t see the point of that).

#3 - 05/20/2021 11:18 AM - hsbt (Hiroshi SHIBATA)

So I guess you mean no longer including them with Ruby, and gem install NAME would be needed to use them?

Yes. I meant to remove them ruby repository and release.

#4 - 05/20/2021 12:24 PM - byroot (Jean Boussier)
They are no longer actively maintained and not widely used today.

mutex_m

In my opinion mutex_m is widely used. Even Rails uses it: [https://github.com/rails/rails/search?q=mutex_m](https://github.com/rails/rails/search?q=mutex_m)

#5 - 05/20/2021 12:33 PM - hsbt (Hiroshi SHIBATA)
- Description updated

In my opinion mutex_m is widely used. Even Rails uses it: [https://github.com/rails/rails/search?q=mutex_m](https://github.com/rails/rails/search?q=mutex_m)

Thanks for your information. I agreed your opinion and removed mutex_m from this proposal.

#6 - 05/20/2021 03:17 PM - universato (Yoshimine Sato)
I would like prime library to remain in the default gems or the bundled gems.
I do competitive programming as a hobby, and I sometimes use it and find it useful. Since external libraries are usually not available for competitive programming, I would like prime library to remain in internal gems.

Some Japanese competitors have also mentioned prime library as one of the advantages of Ruby.
Rubyで競プロする Tips - ARMERIA
【AtCoder】Rubyで競プロに入門する - Qiita
【不安解消】Rubyでも競技プログラミングで問題が解ける→茶色に行けました! | RyuCoding
Rubyで競プロをしていた時に使っていたライブラリを供養する - Qiita

#7 - 05/21/2021 04:06 AM - hsbt (Hiroshi SHIBATA)
- Status changed from Open to Assigned
- Description updated

I would like prime library to remain in the default gems or the bundled gems.

Thanks, I moved prime to the bundled gems list. And I removed fcntl from this proposal. It's also widely used in other gems.

#8 - 05/21/2021 04:34 AM - marcandre (Marc-Andre Lafortune)
Thanks, I agree about prime.
I imagine matrix users are limited and will be ok. I'm not sure about ostruct.
Could we consider to bundle them in next major version but with some warning? If you "update" the gem then the conversion warning would not be there. This could ease conversion.

#9 - 05/21/2021 05:39 AM - shugo (Shugo Maeda)
I agree about net/ftp and net/imap.

#10 - 05/21/2021 08:55 AM - hsbt (Hiroshi SHIBATA)
- Description updated

#11 - 05/21/2021 09:05 AM - hsbt (Hiroshi SHIBATA)
- Description updated

I'm not sure about ostruct.

Thanks, I'm also aware this. So, I removed ostruct from this proposal.
I added dbm and gdbm to removal list with other commiter's opinion.

I don't see a good reason to remove the syslog library from ruby. It is so small, simple, and almost maintenance-free because the syslog facility is unlikely to change anymore.

Syslog has long been a standard logging facility in Unix/Linux and I believe it is good for ruby to have access to it out of the box. I can imagine it is used by many admins in their system scripts.

Also, there seem to be many gems that load syslog.

```bash
knu@aluminium:$ gem-codesearch "^ *require ['\"]syslog['\"]" | ruby -ne 'puts $_.split('"')[3]' | sort -u | wc -l
259
```

I have no strong opinion to remove syslog in Ruby 3.1. I withdraws it from this proposal.

I applied this proposal at https://github.com/ruby/ruby/pull/4530

But I keep to open this until releasing preview or rc version of Ruby 3.1.

Just for the record:

- The matrix removal broke the capybara gem: https://github.com/teamcapybara/capybara/pull/2468
- The net-smtp remove broke the mail gem: https://github.com/mikel/mail/pull/1439

Both are Rails dependencies.

Doesn't mean it should be reverted, I personally think the change make sense, but I figured it should be reported here.

Actually, there's a bit of a problem. If you add a default gem to the gemspec it cause double loading issues, e.g. with mail, net-smtp pulls digest:

```
/opt/rubies/2.7.2/lib/ruby/2.7.0/digest.rb:6: warning: previous definition of REQUIRE_MUTEX was here
~/.gem/ruby/2.7.2/gems/digest-3.0.0/lib/digest.rb:8: warning: method redefined; discarding old const_missing
```

So I'm not sure what the path forward is for these gems.

This sometimes happens when rubygems loads a default gem too early before bundler has been setup. Then when user code loads the actual gem it uses the $LOAD_PATH setup by bundler instead of the default one, causing the double load. Up until now we always found ways to fix these issues by making usages of default gems inside rubygems as lazy as possible, or by vending copies of the default gems that we need under our namespace so that we don't interfere with user's choice, but it seems like we're still loading digest too early.

I'll have a look at this.

I'll have a look at this.

Thank you!

I used digest as an example, but there is also net/protocol: https://buildkite.com/rails/rails/builds/77709#09/e4e4f7a-153d-403c-9ad0-6579b3c04af1/999-1010
The net-protocol one seems similar to: https://github.com/ruby/net-imap/issues/16.

#21 - 05/28/2021 06:25 PM - marcandre (Marc-Andre Lafortune)
byroot (Jean Boussier) wrote in #note-16:

   Just for the record:
   
   - The matrix removal broke the capybara gem: https://github.com/teamcapybara/capybara/pull/2468
   - The net-smtp remove broke the mail gem: https://github.com/mikel/mail/pull/1439

   Both are Rails dependencies.

   Doesn't mean it should be reverted, I personally think the change make sense, but I figured it should be reported here.

Thanks.

I'll repeat: could we consider to still bundle these gems in next major version but with some warning? If you "update" the gem then the conversion warning would not be there. This could ease conversion.

#22 - 07/06/2021 12:38 PM - Eregon (Benoit Daloze)
byroot (Jean Boussier)deivid (David Rodríguez) I think it would best to track this double inclusion problem as an issue on https://github.com/rubygems/rubygems/issues, could you file one?

#23 - 07/06/2021 04:50 PM - deivid (David Rodríguez)
Thanks for the remainder, I forgot. I did try to repro this at the time by simulating locally what Rails CI does, but I couldn't trigger the double load. Best to open an issue at the rubygems repository with repro steps.

#24 - 07/07/2021 08:32 AM - byroot (Jean Boussier)

   but I couldn't trigger the double load. Best to open an issue at the rubygems repository with repro steps.

So I was able to repro, but I can confirm that it is fixed on latest rubygems/bundler.

The problem is that lots of users don't keep both of these up to date, so that will cause lots of confusion.

#25 - 07/07/2021 10:06 AM - deivid (David Rodríguez)
Great to hear that it's fixed!

Regarding your concern, this is a change for ruby 3.1, so as long as ruby 3.1 provides a fixed version of bundler & rubygems, it should be ok, right?

#26 - 07/07/2021 11:23 AM - byroot (Jean Boussier)

   so as long as ruby 3.1 provides a fixed version of bundler & rubygems, it should be ok, right?

No, the problem is that to be compatible with 3.1, these gems will have to specify these dependencies. But anyone using gems defining these dependencies but still using older rubygems/bundler will have the double loading issue.

So there's really nothing we can do '/', people will have to upgrade their rubygems/bundler.

#27 - 07/07/2021 11:44 AM - deivid (David Rodríguez)
I see. It doesn't seem like a big deal to me.

#28 - 07/07/2021 11:53 AM - Eregon (Benoit Daloze)

   Doesn't it mean that in addition to the double-loading warnings (e.g., when using Rails) it might also break on current Ruby releases with the version of RubyGems they ship? (due to the two loaded versions creating some incompatibility)
   It's also likely hard to guess that double loading means "update to latest RubyGems and Bundler" (they can be other causes for it).

   In a Gemfile, it would be possible to specify those gems are only needed for Ruby 3.1+, but in a gemspec that does not seem possible unfortunately.

byroot (Jean Boussier) Which Ruby versions did you see was affected? If it's only unsupported Ruby versions at least it'd be a smaller concern.

FWIW Rails main currently does this, the same can be done for generated applications's Gemfile.
It is not pretty, but if it avoids double-loading maybe it is better than specifying the dependencies in the gem and potentially breaking running Rails on many Ruby releases.

#29 - 07/07/2021 12:12 PM - deivid (David Rodríguez)

Yeah, technically it could cause issues, that's why ruby prints a warning. But I've never seen an actual issue in real life, just users bothered by the warning. Concerned users usually look for the root cause, and eventually figure out they need to upgrade rubygems/bundler or report the issue to the affected repository. That's how I've been normally finding out about and fixing these issues in the past years since the gemification of the stdlib started.

#30 - 07/07/2021 02:31 PM - byroot (Jean Boussier)

Hum, actually I used a docker image to try to pinpoint when it was fixed, and I'm no longer sure it is fixed, or at least another issue cropped up:

```ruby
# frozen_string_literal: true

require "bundler/inline"

gemfile(true) do
  source "https://rubygems.org"
  gem "mail", github: "Shopify/mail", branch: "net-smtp-dependency"
end

require "mail"

$ docker run -v /tmp:/tmp -it ruby:3.0.1 bash -c 'gem update --system --no-document & & gem install --no-document bundler & & ruby /tmp/repro-double-loading.rb'

....

Installing net-imap 0.2.2
Using mail 2.8.0.edge from https://github.com/Shopify/mail.git (at net-smtp-dependency@58704be)
/usr/local/bundle/gems/bundler-2.2.22/lib/bundler/runtime.rb:302:in `check_for_activated_spec!': You have already activated net-protocol 0.1.0, but your Gemfile requires net-protocol 0.1.1. Since net-protocol is a default gem, you can either remove your dependency on it or try updating to a newer version of bundler that supports net-protocol as a default gem. (Gem::LoadError)

So now I get a hard failure. Not sure why it works on my host.

#31 - 07/07/2021 05:34 PM - deivid (David Rodríguez)

Alright, thanks. I'll have a look at that, then.

#32 - 07/08/2021 10:40 AM - deivid (David Rodríguez)

At first glance, it looks specific to bundler/inline, when installing gems and using gems happens in the same process. I'll fix it by vending net/http and its dependencies under the Bundler:: namespace like we do with the other pure ruby libraries that bundler depends on.