Ruby master - Feature #17684
Remove `--disable-gems` from release version of Ruby
03/10/2021 12:51 PM - hsbt (Hiroshi SHIBATA)

Status: Open
Priority: Normal
Assignee: 
Target version: 

Description
In my understand, --disable-gems is only debugging feature for ruby-core team.
But some users enabled its option in test environment for performance or etc. So, --disable-gems option is wrong usage for some users.

- https://github.com/rubygems/bundler/issues/7487#issuecomment-569901549
- https://github.com/rubygems/rubygems/pull/4440#issue-587031184

We should remove it from package version of ruby.

History
#1 - 03/10/2021 12:59 PM - chrisseaton (Chris Seaton)
How should people who were using it for performance migrate off the option?

#2 - 03/10/2021 01:00 PM - naruse (Yui NARUSE)
Agreed. Unfortunately this increases the cost of development to support --disable-gems.
In other words, this feature breaks the assumption that every Ruby has RubyGems.
This feature should be removed.

#3 - 03/10/2021 01:02 PM - naruse (Yui NARUSE)
chrisseaton (Chris Seaton) wrote in #note-1:

How should people who were using it for performance migrate off the option?

--disable-gems should be just ignored in the future version, and the performance will be degraded.

#4 - 03/10/2021 01:21 PM - chrisseaton (Chris Seaton)
I don't have anything invested in this myself, but just for information to help the discussion, here's two examples of apps that use the option

- https://github.com/Shopify/shopify-app-cli/blob/master/bin/shopify

#5 - 03/10/2021 01:35 PM - xtkoba (Tee KOBAYASHI)
Will we be able to re-enable the --disable-gems option when we build ruby from source code? I think it is useful for narrowing problems down. For that purpose it might be sufficient that mini-ruby does not depend on Gems, though.

For performance we can use mruby (https://github.com/mruby/mruby) after all.

#6 - 03/10/2021 02:53 PM - vo.x (Vit Ondruch)
I'd love to see this clarified. IMO, there should be other ways to manage $LOAD_PATH then RubyGems. For such use cases, this option is useful IMO.

#7 - 03/10/2021 03:41 PM - deivid (David Rodriguez)
I disagree on removing this option, I don't think the benefits of removing it are strong enough, even if it didn't make any difference in performance.

#8 - 03/11/2021 12:40 AM - hsbt (Hiroshi SHIBATA)
- Description updated

#9 - 03/13/2021 11:02 AM - NuriYuri (Youri Nouri)
I'm also using the --disable-gems option. I make games on Ruby but I'm not using gems and since the main audience is Windows, I have to distribute portable Ruby version that HAS to work on any path (including those with accents). I noticed that the rubygems are causing a lot of issues when the game needs to load native extensions (FMOD, SFML, sockets...) so I disable the gems and if they're needed, we include them explicitly using the require 'rubygems' command.

I would suggest that people who needs gems require rubygems manually or that ruby load rubygems automatically only if the script has a Gemfile (I bet this might already be handled by bundler & the bundle exec command).

#10 - 03/13/2021 12:50 PM - Eregon (Benoit Daloze)

I think --disable-gems is also useful to investigate issues, e.g. RubyGems used to override Kernel#warn (fixed in Ruby 3).
It can significantly simplify investigating issues with require.
I agree it's not very common to need it for debugging, but it turned up useful a couple times.
If it's not available for releases then it becomes inconvenient to investigate such cases.

Also the impact on startup time is still significant, so for very short scripts it can be noticeable (especially if calling out to subprocesses or running many small scripts).
That's something that can potentially be improved in CRuby and/or RubyGems.
For instance by loading RubyGems lazily (autoload :Gem, try loading rubygems the first time require fails), which TruffleRuby does.

Finally, I guess for JITs it's quite useful to limit the amount of Ruby code loaded and run on startup, so having --disable-gems to investigate JIT issues seems useful.
If it only works on development builds, then again it's inconvenient to investigate a bug report from a user using a release.