

## Ruby - Feature #17684

### Remove `--disable-gems` from release version of Ruby

03/10/2021 12:51 PM - hsbt (Hiroshi SHIBATA)

<b>Status:</b>	Assigned	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	hsbt (Hiroshi SHIBATA)	
<b>Target version:</b>		
<b>Description</b> 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.  <ul style="list-style-type: none"><li>• <a href="https://github.com/rubygems/bundler/issues/7487#issuecomment-569901549">https://github.com/rubygems/bundler/issues/7487#issuecomment-569901549</a></li><li>• <a href="https://github.com/rubygems/rubygems/pull/4440#issue-587031184">https://github.com/rubygems/rubygems/pull/4440#issue-587031184</a></li></ul> We should remove it from package version of ruby.		
<b>Related issues:</b> Related to Ruby - Feature #18376: Version comparison API <b>Open</b> Related to Ruby - Feature #18568: Explore lazy RubyGems boot to reduce need f... <b>Feedback</b> Related to Ruby - Misc #19178: How does CRuby handle CVE issues in stdlib gem... <b>Closed</b>		

#### Associated revisions

**Revision d58daad37bd5fd09c7ff74be17074fc51faac8d3 - 03/17/2021 06:36 AM - naruse (Yui NARUSE)**

[Feature #17684] Declare --disable-gems is for debugging

**Revision d58daad37bd5fd09c7ff74be17074fc51faac8d3 - 03/17/2021 06:36 AM - naruse (Yui NARUSE)**

[Feature #17684] Declare --disable-gems is for debugging

**Revision d58daad3 - 03/17/2021 06:36 AM - naruse (Yui NARUSE)**

[Feature #17684] Declare --disable-gems is for debugging

**Revision 2c0de392870232b56b28e14e8cbc6d993d8648ac - 12/13/2021 05:19 AM - mame (Yusuke Endoh)**

NEWS.md: --disable-gems is now just for debugging. [Feature #17684]

**Revision 2c0de392870232b56b28e14e8cbc6d993d8648ac - 12/13/2021 05:19 AM - mame (Yusuke Endoh)**

NEWS.md: --disable-gems is now just for debugging. [Feature #17684]

**Revision 2c0de392 - 12/13/2021 05:19 AM - mame (Yusuke Endoh)**

NEWS.md: --disable-gems is now just for debugging. [Feature #17684]

#### 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>
- <https://github.com/penelopezone/rubyfmt/blob/trunk/librubyfmt/script/run.sh>

#### #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 miniruby 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 Rodríguez)

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 (FMODE, 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.

#### #11 - 03/15/2021 05:41 PM - tenderlovmaking (Aaron Patterson)

I like using --disable-gems for scripts that need to be fast. For example, I call ruby from Vim to get some information about the Ruby environment:

```
" Add stdlib of environment's ruby to path
let g:stdlib = system('ruby --disable-gems -rrbconfig -e"print RbConfig::CONFIG[\"rubylibdir\"]"')
let &path .= "," . stdlib
let g:ruby_path = &path
```

Adding 60ms (or more) for these "fast commands" would be pretty annoying.

```
[aaron@tc-lan-adapter ~]$ time ruby --disable-gems -rrbconfig -e"print RbConfig::CONFIG[\"rubylibdir\"]"
/Users/aaron/.rubies/ruby-trunk/lib/ruby/3.1.0
```

Executed in	20.60 millis	fish	external
usr time	12.12 millis	146.00 micros	11.98 millis
sys time	6.33 millis	741.00 micros	5.59 millis

```
[aaron@tc-lan-adapter ~]$ time ruby -rrbconfig -e"print RbConfig::CONFIG[\"rubylibdir\"]"
```

/Users/aaron/.rubies/ruby-trunk/lib/ruby/3.1.0

Executed in	79.02 millis	fish	external
usr time	61.54 millis	115.00 micros	61.43 millis
sys time	14.46 millis	679.00 micros	13.78 millis

#### #12 - 03/15/2021 05:56 PM - lugray (Lisa Ugray)

I also have a use case where I use `--disable-gems` for performance. I maintain a tool that provides some shell hooks in ruby, and any slow down in getting to a prompt is incredibly noticeable.

#### #13 - 03/15/2021 10:23 PM - Eregon (Benoit Daloze)

Another use case: Bundler's `--standalone` mode knows all gem directories and doesn't need to load either RubyGems or Bundler, which can save quite some time on application startup.

It seems a shame if it would still load RubyGems needlessly due to removing `--disable-gems`.

#### #14 - 03/15/2021 11:31 PM - tenderlovmaking (Aaron Patterson)

I forgot another place I use `--disable-gems`. I think RubyGems does lots of IO at startup, and IO is extremely slow on my Raspberry PI. I'm using Ruby on Raspberry PI for some IoT stuff around my house and I use `--disable-gems` in those environments. Here is the difference:

```
pi@lcd-screen:~ $ time ruby -v --disable-gems -e ' '
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [armv6l-linux-eabi]

real 0m0.291s
user 0m0.118s
sys 0m0.082s
pi@lcd-screen:~ $ time ruby -v -e ' '
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [armv6l-linux-eabi]

real 0m2.395s
user 0m1.998s
sys 0m0.129s
```

My Ruby installation only has the default gems installed, but RubyGems adds 2 seconds to startup.

#### #15 - 03/17/2021 06:31 AM - shyouhei (Shyouhei Urabe)

My gut feeling is that it's miserable for people to confuse `--disable-gems` as a zero-cost speed boost option. Maybe rubygems is too slow now.

#### #16 - 03/17/2021 06:37 AM - naruse (Yui NARUSE)

- Status changed from Open to Closed

Applied in changeset [gitId58daad37bd5fd09c7ff74be17074fc51faac8d3](https://github.com/rubygems/rubygems/commit/gitId58daad37bd5fd09c7ff74be17074fc51faac8d3).

---

[Feature [#17684](#)] Declare `--disable-gems` is for debugging

#### #17 - 03/17/2021 06:39 AM - naruse (Yui NARUSE)

At this time we just update `--help`.

The conclusion of ruby core's discussion is

- As the first step, let's update the `--help` to explicitly state that `--disable-gems` is just for debugging
- If rubygems startup becomes so fast in future, we can consider removal of `--disable-gems` (but still it is desirable for debugging)
- Or someone who happen to report about `--disable-gems` again, we will consider again.

We also discuss RubyGem accesses too many files in startup. We expect gem team improve that.

<https://gist.github.com/ko1/f0d8c899d580f7f69bb5608ab6d8215f>

#### #18 - 12/02/2021 08:10 AM - naruse (Yui NARUSE)

- Related to Feature [#18376](#): Version comparison API added

#### #19 - 02/02/2022 08:32 PM - Eregon (Benoit Daloze)

- Related to Feature [#18568](#): Explore lazy RubyGems boot to reduce need for `--disable-gems` added

#### #20 - 12/13/2022 02:27 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Closed to Open

**#21 - 12/13/2022 02:28 AM - hsbt (Hiroshi SHIBATA)**

- *Related to Misc #19178: How does CRuby handle CVE issues in stdlib gems which get patched?* added

**#22 - 03/10/2023 11:57 AM - hsbt (Hiroshi SHIBATA)**

- *Status changed from Open to Assigned*

- *Assignee set to hsbt (Hiroshi SHIBATA)*