Ruby master - Bug #18066
Load did_you_mean/error_highlight even with --disable-gems
08/07/2021 08:30 PM - vo.x (Vit Ondruch)

Status: Rejected
Priority: Normal
Assignee:
Target version:
ruby -v:
ruby 3.0.1p64 (2021-04-05 revision 0fb782ee38) [x86_64-linux]
Backport: 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN

Description
I guess that did_you_mean was not possible to load without RubyGems, when it used to be bundled gem. Since it is default gem, therefore part of StdLib and always available on load path, it should not be related to --disable-gems anymore.

IOW ruby.c contains this code:

```c
if (opt->features.set & FEATURE_BIT(gems)) {
   rb_define_module("Gem");
   if (opt->features.set & FEATURE_BIT(error_highlight)) {
      rb_define_module("ErrorHighlight");
   }
   if (opt->features.set & FEATURE_BIT(did_you_mean)) {
      rb_define_module("DidYouMean");
   }
}
```

while it should look like:

```c
if (opt->features.set & FEATURE_BIT(gems)) {
   rb_define_module("Gem");
}

if (opt->features.set & FEATURE_BIT(error_highlight)) {
   rb_define_module("ErrorHighlight");
}

if (opt->features.set & FEATURE_BIT(did_you_mean)) {
   rb_define_module("DidYouMean");
}
```

(I have not checked error_highlight, but I assume it behaves similarly to did_you_mean)

History
#1 - 08/07/2021 08:35 PM - vo.x (Vit Ondruch)
However, I am not sure what is the impact in case Ruby can be build with --disable-gems configuration option (if that is possible).

#2 - 08/07/2021 08:38 PM - vo.x (Vit Ondruch)
- Subject changed from Load did_you_mean/eve/error_highlight even with --disable-gems to Load did_you_mean/error_highlight even with --disable-gems

#3 - 08/10/2021 07:22 PM - jeremyevans0 (Jeremy Evans)
I think this change makes sense. It requires quite a few test changes, though. I submitted a pull request to implement it:
https://github.com/ruby/ruby/pull/4729

#4 - 08/10/2021 11:49 PM - duerst (Martin Dürst)
I'm not sure how strongly this is related (and I'll create a new issue if necessary), but when I use a very simple script such as ruby -e 'puts 3+4', I get the following output:

```
'RubyGems' were not loaded.
'error_highlight' was not loaded.
```
I think this is total overkill. Don't get me wrong; I'm fine (in this specific case) with RubyGems not being loaded (because no Gems are needed). I'm also fine with error_highlight and did_you_mean not being loaded (because they are not needed as long as my script has no errors). But it's total overkill that I'm told about this with 3 lines when all I want is to run a tiny script.

If this is a transitory state, then please ignore this complaint.

(ruby -v: ruby 3.1.0dev (2021-08-10T10:08:38Z master 28d03ee776) [x86_64-linux] on WSL2 on Win10)

#5 - 08/11/2021 09:54 AM - jaruga (Jun Aruga)

I'm not sure how strongly this is related (and I'll create a new issue if necessary), but when I use a very simple script such as ruby -e 'puts 3+4', I get the following output:

If this is a transitory state, then please ignore this complaint.

(ruby -v: ruby 3.1.0dev (2021-08-10T10:08:38Z master 28d03ee776) [x86_64-linux] on WSL2 on Win10)

I tried to reproduce what you did on the master 28d03ee776. But in my case, I used Fedora 34, as I don't have Windows. And the result looks ok on my environment.

$ cat /etc/fedora-release
Fedora release 34 (Thirty Four)
$ ~:/local/ruby-28d03ee776/bin/ruby -v
ruby 3.1.0dev (2021-08-10T10:08:38Z master 28d03ee776) [x86_64-linux]
$ ~:/local/ruby-28d03ee776/bin/ruby -e 'puts 3+4'
7

#6 - 08/11/2021 10:53 AM - vo.x (Vit Ondruch)

duerst (Martin Dürst) wrote in #note-4:

I'm not sure how strongly this is related (and I'll create a new issue if necessary), but when I use a very simple script such as ruby -e 'puts 3+4', I get the following output:

'RubyGems' were not loaded.
'error_highlight' was not loaded.
'did_you_mean' was not loaded.
7

Under what conditions you see these reports? This 1 is the place where those comes from, but you also need to have the relevant files removed from StdLib.

#7 - 08/11/2021 04:01 PM - jaruga (Jun Aruga)

vo.x (Vit Ondruch) I would just share how the motivation of some maintainers look like to fix this kind of issue, as I chat with some maintainers on Slack today.

I understand that gems moving from bundled gem to default gem as StdLib tend to depend on RubyGems at an early time. I think your request comes from the fact that the "ruby" RPM package including StdLib has not depended on the "rubygems" sub RPM package on Fedora. And I see that the motivation is that we want to install only what we need, and good to reduce the installed files size.

Seeing Debian and Ubuntu ruby packages, the "ruby" deb package depends on "ruby-rubygems" deb package. I am curious about the situation of the "ruby" package on other Linux/UNIX platforms such as Arch, Gentoo, FreeBSD.
https://packages.debian.org/experimental/ruby
https://packages.ubuntu.com/impish/ruby

When chatting today, I see some maintainers were less motivated. Because historically in the age of Ruby 1.8, we installed RubyGems in addition. But nowadays splitting Ruby from RubyGems is not really a supported use case. It's a Linux, BSD distribution's ruby package specific use case. So, the way to make it possible nowadays is that some motivated people (Thanks Jeremy this time!) will send a PR, or Linux/*BSD packagers will rethink adding "rubygems" as a required dependency of "ruby" like Debian/Ubuntu. So, I think this kind of task is not "must" but "nice to have".

#8 - 08/11/2021 05:26 PM - vo.x (Vit Ondruch)

Just for the context, this issue was triggered when looking at 1. And as long as there is ruby --disable-gems as well as .configure --disable-rubygems, I believe that it is correct to fix this.

#9 - 08/11/2021 06:46 PM - xtkoba (Tee KOBAYASHI)
jaruga (Jun Aruga) wrote in [#note-7]:

Seeing Debian and Ubuntu ruby packages, the “ruby” deb package depends on “ruby-rubygems” deb package. I am curious about the situation of the “ruby” package on other Linux/UNIX platforms such as Arch, Gentoo, FreeBSD.

Both Arch and Gentoo require ruby to be installed together with rubygems:
https://archlinux.org/packages/extra/x86_64/ruby/
https://packages.gentoo.org/packages/dev-lang/ruby/dependencies

#10 - 10/25/2021 09:05 AM - mame (Yusuke Endoh)

- Status changed from Open to Rejected

We discussed this ticket at the dev-meeting, and matz (Yukihiro Matsumoto) rejected this.

Now the --disable-gems option is just for debugging, not for casual users. We, the core developers, mainly use the option to investigate the startup overhead of the interpreter core, so typically we want to disable did_you_mean and error_highlight when we use --disable-gems. If we really want to enable did_you_mean under --disable-gems, we can use --disable-gems -rdid_you_mean.