Ruby - Feature #15611

Shipping Bundler as a bundled gem, not a default gem

02/18/2019 06:35 PM - Eregon (Benoit Daloze)

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

Description

I think this would simplify many things, and would allow to update or remove the shipped Bundler easily on Ruby 2.6.

Is there a particular reason to have Bundler as a default gem?

Also, given how Bundler magically switches which version is used based on the Gemfile, it seems better to have each Bundler version in its own directory,
rather thank risking to load multiple Bundler versions or parts of it from lib/, site_ruby/ and the bundler gem directory.

Related issues:

Related to Ruby - Misc #15610: Could bundler & rubygems be shipped in site_ruby?	**Rejected**

History

#1 - 02/18/2019 06:35 PM - Eregon (Benoit Daloze)

- Related to Misc #15610: Could bundler & rubygems be shipped in site_ruby? added

#2 - 02/19/2019 12:59 AM - mame (Yusuke Endoh)

As far as I know, bundler is planned to be merged to rubygems. If shipping bundler as a gem is ephemeral, it looks less significant to me whether bundler should be a bundled gem or a default one.

In fact, rubygems already includes some special code for bundler, e.g., bundler_version_finder.rb. I think that it is going to be unreasonable to separate bundler as a gem.

Just my two cents,

#3 - 02/19/2019 09:25 AM - deivid (David Rodríguez)

Yes, I was going to comment exactly along the same lines mame presents.

Also, moving bundler to a default gem would still keep the same problem in https://bugs.ruby-lang.org/issues/15610, but only for the rubygems code. Although to be honest, having duplicated versions of the rubygems code in rubylibdir and site_ruby has not yet led to real world issues that I know of. But still, moving both to site_ruby feels cleaner and safer to me.

#4 - 02/19/2019 12:26 PM - Eregon (Benoit Daloze)

> As far as I know, bundler is planned to be merged to rubygems.

What would that mean exactly? (that's a question for Bundler/RubyGems devs)
Will both repositories become one and the code be reused for both parts? That's what I would consider "merged".
Otherwise, it looks like two different softwares with different versions. Currently it doesn't look merged at all to me.

> If shipping bundler as a gem is ephemeral, it looks less significant to me whether bundler should be a bundled gem or a default one.

Not sure how ephemeral, but right now bundler is a default gem in the 2.6 releases, and as argued above I think it's an actual issue for it to be a default gem.

> Also, moving bundler to a default gem would still keep the same problem in https://bugs.ruby-lang.org/issues/15610, but only for the rubygems code. Although to be honest, having duplicated versions of the rubygems code in rubylibdir and site_ruby has not yet led to real world issues that I know of. But still, moving both to site_ruby feels cleaner and safer to me.

RubyGems has been in stdlib since 1.9, so there is some history there, and RubyGems itself cannot be a gem of course.
But I agree, it would be more consistent to always have RubyGems be installed the same directory.
I wonder what should happen when upgrading RubyGems fails while copying files though, maybe having a safe copy somewhere would be a good

way to deal with that.

I think it would be better if every part of stdlib that can be upgraded could be cleanly removed and handled like a real gem, which seems very complicated for stdlibs directly under lib/.
This also applies for other default gems part of the stdlib.

#5 - 02/19/2019 05:56 PM - deivid (David Rodríguez)

Also, moving bundler to a default gem would still keep the same problem...

When I said "default gem" before I meant "bundled gem", but I think you understood me... :)

¿What would that mean exactly? (that's a question for Bundler/RubyGems devs)
Will both repositories become one and the code be reused for both parts? That's what I would consider "merged".
Otherwise, it looks like two different softwares with different versions. Currently it doesn't look merged at all to me.

Yeah, that's what it means. But it's currently only a long term goal.

RubyGems has been in stdlib since 1.9, so there is some history there, and RubyGems itself cannot be a gem of course.
But I agree, it would be more consistent to always have RubyGems be installed the same directory.
I wonder what should happen when upgrading RubyGems fails while copying files though, maybe having a safe copy somewhere would be a good way to deal with that.

Right, I guess that's precisely why rubygems is currently shipped in rubylibdir, so that the default installation is never changed?

#6 - 02/26/2019 08:46 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

#7 - 03/08/2019 11:53 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Assigned to Rejected

Is there a particular reason to have Bundler as a default gem?

Already @mame (Yusuke Endoh) said. Bundler will merge into RubyGems in the feature. In fact, after RubyGems 2.7+ integrated the part of dependency resolver of Bundler.
Therefore We should test Bundler and RubyGems every commits of CRuby. If We shipped Bundler as bundled gems, We couldn't test its integrated parts.

PS. The RubyGems and Bundler team are going to discuss its details at before/after Dev MTG in RubyKaig 2019.