Ruby master - Feature #16652

Use RubyGems facilities to install Gems during Ruby installation

02/24/2020 05:10 PM - vo.x (Vit Ondruch)

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

Description

Currently, tools/rbinstall uses quite some custom code to install gems instead of using functionality available in RubyGems. I don't think this is good, because the code is non transparent ATM and it is duplicating functionality, which is already available and tested in RubyGems. Also, it intoduces its own concepts such as $script_mode vs $prog_mode. So I opened two PRs, gradually implementing this.

The first PR 1 uses Gem::Package like object and duck typing instead of monkey patching.

1. This is similar to what RubyGems do and it is less magic 2.
2. It avoids deprecated code paths in RubyGems 3.

The second PR 4 builds on the top of the first one and uses UnpackedInstaller to install default gems (it would be probably nice to get UnpackedInstaller into RubyGems, but I am not sure how it could be invoked). As a bonus, it removes the pregenerated binstubs. These are now generated by Gem::Installer instead.

History

#1 - 02/25/2020 02:24 AM - shevegen (Robert A. Heiler)

I glanced at rbinstall.rb briefly; for those interested, the link should be here:

https://github.com/ruby/ruby/blob/master/tool/rbinstall.rb

That seems quite old code, lots of global variables in it as well. Props to everyone able to improve on it. My best bet is that it was written a long time ago. :-)

(I wondered if we could actually put all the .rb files that remain as part of the ruby tarball into its own gem, but I am not sure if the effort may be worth it. So props to you and everyone else who improves on old code!)

#2 - 02/26/2020 05:48 AM - hsbt (Hiroshi SHIBATA)

- Assignee set to hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

#3 - 02/27/2020 11:08 AM - vo.x (Vit Ondruch)

Since the first PR was merged, I have updated the second PR 1 to apply cleanly.