Ruby master - Bug #13997

Bundler gem binstub broken

10/10/2017 10:54 PM - hone (Terence Lee)

Status:	Closed		
Priority:	Normal		
Assignee:	hsbt (Hiroshi SHIBATA)		
Target version:	2.5		
ruby -v:	ruby 2.5.0preview1 (2017-10-10 trunk 60153) [x86_64-linux]	**Backport:**	

Description

Hi,

In Ruby 2.5.0-preview1, I'm seeing the following error if the Rubygems binstub PATH precedes the binstubs from ruby and the Bundler version installed is <= 1.15.4 (what ruby vendors).

```
/home/hone/.rvm/gems/ruby-2.4.1@global/bin/bundle:22:in `load': /tmp/bundler/lib/ruby/gems/2.5.0/g
ems/bundler-1.15.4/exe/bundle:4: syntax error, unexpected tSTRING_BEG, expecting keyword_do or '{'
 or '(' (SyntaxError)
exec "$bindir/ruby" -x "$0" "$@"
                        ^
/tmp/bundler/lib/ruby/gems/2.5.0/gems/bundler-1.15.4/exe/bundle:9: syntax error, unexpected keywor
d_do_block, expecting end-of-input
Signal.trap("INT") do
```

The reason for this is because the generated bundler binstub by rubygems runs code like this:

```
load Gem.bin_path('bundler', 'bundle', '>= 0.a')
```

The problem is that the binstub from the vendored bundler in ruby is not pure ruby:

```
#!/bin/sh
# -*- ruby -*-
bindir="${0%/*}"
exec "$bindir/ruby" -x "$0" "$@"
```

The top of this file is shell and not ruby. Kernel#load chokes on this.

Thanks,
Terence

History

#1 - 10/11/2017 09:07 AM - Eregon (Benoit Daloze)

That binstub looks wrong, I would guess it's Bundler's bug.
After /bin/sh should be shell code, and after the # ruby comment Ruby code.

#2 - 10/11/2017 10:40 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Feedback

Thanks, Terence.

/tmp/bundler/lib/ruby/gems/2.5.0/gems/bundler-1.15.4/exe/bundle is not default gem environment on Ruby 2.5.0preview1.

Can you show reproduce instructions for this issue?

#3 - 10/11/2017 08:09 PM - hone (Terence Lee)

Eregon (Benoit Daloze) wrote:

> That binstub looks wrong, I would guess it's Bundler's bug.
> After /bin/sh should be shell code, and after the # ruby comment Ruby code.

This is the binstub generated from Ruby's vendored bundler (in the exe folder)

```
#!/bin/sh
# -*- ruby -*-
bindir="${0%/*}"
exec "$bindir/ruby" -x "$0" "$@"
#!/usr/bin/env ruby
# frozen_string_literal: true

# Exit cleanly from an early interrupt
Signal.trap("INT") do
  Bundler.ui.debug("\n#{caller.join("\n")}") if defined?(Bundler)
  exit 1
end

require "bundler"
# Check if an older version of bundler is installed
$LOAD_PATH.each do |path|
  next unless path =~ %r{/bundler-0\.(\d+)} && $1.to_i < 9
  err = String.new
  err << "Looks like you have a version of bundler that's older than 0.9.\n"
  err << "Please remove your old versions.\n"
  err << "An easy way to do this is by running `gem cleanup bundler`."
  abort(err)
end

require "bundler/friendly_errors"
Bundler.with_friendly_errors do
  require "bundler/cli"

  # Allow any command to use --help flag to show help for that command
  help_flags = %w(--help -h)
  help_flag_used = ARGV.any? {|a| help_flags.include? a }
  args = help_flag_used ? Bundler::CLI.reformatted_help_args(ARGV) : ARGV

  Bundler::CLI.start(args, :debug => true)
end
```

Bundler's binstub normally looks like this:

```
#!/usr/bin/env ruby
# frozen_string_literal: true

# Exit cleanly from an early interrupt
Signal.trap("INT") { exit 1 }

update = "update".start_with?(ARGV.first || " ") && ARGV.find {|a| a.start_with?("--bundler") }
update &&= update =~ /--bundler(?:=(.+))?/ && $1 || "> 0.a"
ENV["BUNDLER_VERSION"] = update if update
require "bundler/postit_trampoline"

require "bundler"
# Check if an older version of bundler is installed
$LOAD_PATH.each do |path|
  next unless path =~ %r{/bundler-0\.(\d+)} && $1.to_i < 9
  err = String.new
  err << "Looks like you have a version of bundler that's older than 0.9.\n"
  err << "Please remove your old versions.\n"
  err << "An easy way to do this is by running `gem cleanup bundler`."
  abort(err)
end

require "bundler/friendly_errors"
Bundler.with_friendly_errors do
  require "bundler/cli"

  # Allow any command to use --help flag to show help for that command
  help_flags = %w(--help -h)
  help_flag_used = ARGV.any? {|a| help_flags.include? a }
  args = help_flag_used ? Bundler::CLI.reformatted_help_args(ARGV) : ARGV

  Bundler::CLI.start(args, :debug => true)
end
```

#4 - 10/11/2017 10:29 PM - hone (Terence Lee)

hsbt (Hiroshi SHIBATA) wrote:

> Thanks, Terence.
>
> /tmp/bundler/lib/ruby/gems/2.5.0/gems/bundler-1.15.4/exe/bundle is not default gem environment on Ruby 2.5.0preview1.
>
> Can you show reproduce instructions for this issue?

/tmp/bundler/ is just the directory where I unpacked Ruby 2.5.0.

I investigated some more and it looks like Benoit is partially right. Thanks for pointing me in the right direction! On Heroku we use --enable-load-relative configure option for compiling ruby, which adds the shell portion to the bundle file.

Without --enable-load-relative, the binstub looks like this:

```
#!/tmp/rubytest2/bin/ruby
# frozen_string_literal: true

# Exit cleanly from an early interrupt
Signal.trap("INT") do
  Bundler.ui.debug("\n#{caller.join("\n")}") if defined?(Bundler)
  exit 1
end

require "bundler"
# Check if an older version of bundler is installed
$LOAD_PATH.each do |path|
  next unless path =~ %r{/bundler-0\.(\d+)} && $1.to_i < 9
  err = String.new
  err << "Looks like you have a version of bundler that's older than 0.9.\n"
  err << "Please remove your old versions.\n"
  err << "An easy way to do this is by running `gem cleanup bundler`."
  abort(err)
end

require "bundler/friendly_errors"
Bundler.with_friendly_errors do
  require "bundler/cli"

  # Allow any command to use --help flag to show help for that command
  help_flags = %w(--help -h)
  help_flag_used = ARGV.any? {|a| help_flags.include? a }
  args = help_flag_used ? Bundler::CLI.reformatted_help_args(ARGV) : ARGV

  Bundler::CLI.start(args, :debug => true)
end
```

With --enable-load-relative set:

```
#!/bin/sh
# -*- ruby -*-
bindir="${0%/*}"
exec "$bindir/ruby" -x "$0" "$@"
#!/usr/bin/env ruby
# frozen_string_literal: true

# Exit cleanly from an early interrupt
Signal.trap("INT") do
  Bundler.ui.debug("\n#{caller.join("\n")}") if defined?(Bundler)
  exit 1
end

require "bundler"
# Check if an older version of bundler is installed
$LOAD_PATH.each do |path|
  next unless path =~ %r{/bundler-0\.(\d+)} && $1.to_i < 9
  err = String.new
  err << "Looks like you have a version of bundler that's older than 0.9.\n"
  err << "Please remove your old versions.\n"
  err << "An easy way to do this is by running `gem cleanup bundler`."
  abort(err)
end
```

```
require "bundler/friendly_errors"
Bundler.with_friendly_errors do
  require "bundler/cli"

  # Allow any command to use --help flag to show help for that command
  help_flags = %w(--help -h)
  help_flag_used = ARGV.any? {|a| help_flags.include? a }
  args = help_flag_used ? Bundler::CLI.reformatted_help_args(ARGV) : ARGV

  Bundler::CLI.start(args, :debug => true)
end
```

As for steps to reproduce:

Install Ruby 2.5.0-preview1

- ./configure --enable-load-relative --prefix /tmp/rubytest
- make
- make install

Setup $PATH

Ensure rubygems binstubs is on the $PATH that precedes ruby's binstubs. We want rubygems binaries to win over ruby's.
For example, assuming the rubygems binstubs live at /home/hone/.gems/bin, $PATH should be setup like this:

```
export PATH=/home/hone/.gems/bin:/tmp/rubytest/bin:$PATH
```

Run the bundle command

```
$ bundle -v
Traceback (most recent call last):
        1: from /home/hone/.gems/bin/bundle:22:in `<main>'
/home/hone/.gems/bin/bundle:22:in `load': /tmp/rubytest/lib/ruby/gems/2.5.0/gems/bundler-1.15.4/exe/bundle:4:
syntax error, unexpected tSTRING_BEG, expecting keyword_do or '{' or '(' (SyntaxError)
exec "$bindir/ruby" -x "$0" "$@"
                    ^
/tmp/rubytest/lib/ruby/gems/2.5.0/gems/bundler-1.15.4/exe/bundle:9: syntax error, unexpected keyword_do_block,
 expecting end-of-input
Signal.trap("INT") do
```

#5 - 10/12/2017 09:27 AM - Eregon (Benoit Daloze)

So #load indeed does not support this skipping-until-ruby-shebang logic that ruby file does.

I wonder why Bundle is trying to load a binary.
To save some time over creating another Ruby process?

#6 - 10/12/2017 06:34 PM - hone (Terence Lee)

Eregon (Benoit Daloze) wrote:

> So #load indeed does not support this skipping-until-ruby-shebang logic that ruby file does.
>
> I wonder why Bundle is trying to load a binary.
> To save some time over creating another Ruby process?

Ah, I see what you're saying.

After some more inspection, it looks like there's something weird going on with how bundler's files in exe/ are generated. This is bundler's file:

```
#!/bin/sh
# -*- ruby -*-
bindir="${0%/*}"
exec "$bindir/ruby" -x "$0" "$@"
#!/usr/bin/env ruby
# frozen_string_literal: true

load File.expand_path("../bundle", __FILE__)
```

This is rake's file:

```
#!/usr/bin/env ruby
```

```
#--
# Copyright (c) 2003, 2004, 2005, 2006, 2007  Jim Weirich
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to
# deal in the Software without restriction, including without limitation the
# rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
# sell copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
# IN THE SOFTWARE.
#++

require "rake"

Rake.application.run
```

#7 - 11/28/2017 01:26 AM - indirect (André Arko)

hsbt (Hiroshi SHIBATA) do you think the difference between Terence's rake and bundle files is because one is a default gem and the other is not?

#8 - 11/29/2017 08:48 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Feedback to Third Party's Issue

hone (Terence Lee)

Can you reproduce without rvm environment?

I think it's problem for ruby_executable_hooks with rvm. like followings.

- https://stackoverflow.com/questions/32308978/ruby-file-ruby-executable-hooks-syntax-error
- https://stackoverflow.com/questions/32186242/rake-dbmigrate-doesnt-work-after-ruby-2-2-2-to-2-2-3-upgrade-via-rvm

#9 - 11/30/2017 06:40 AM - hone (Terence Lee)

hsbt (Hiroshi SHIBATA) wrote:

> hone (Terence Lee)
>
> Can you reproduce without rvm environment?
>
> I think it's problem for ruby_executable_hooks with rvm. like followings.
>
> - https://stackoverflow.com/questions/32308978/ruby-file-ruby-executable-hooks-syntax-error
> - https://stackoverflow.com/questions/32186242/rake-dbmigrate-doesnt-work-after-ruby-2-2-2-to-2-2-3-upgrade-via-rvm

Hi hsbt (Hiroshi SHIBATA), we don't use RVM on Heroku at all. This is just my local machine.

#10 - 11/30/2017 07:32 AM - hsbt (Hiroshi SHIBATA)

I've confused because your first report says.

```
>/home/hone/.rvm/gems/ruby-2.4.1@global/bin/bundle:22...
```

I try your instructions. but I couldn't reproduce it. In current trunk:

```
./configure --with-openssl-dir=/usr/local/opt/openssl --prefix=$HOME/.rbenv/versions/trunk --enable-load-relat
ive
make
make install
```

I got this.

```
~ > cat ~/.rbenv/versions/trunk/bin/bundle
```

```
#!/bin/sh
# -*- ruby -*-
_=_\
=begin
bindir="${0%/*}"
exec "$bindir/ruby" "-x" "$0" "$@"
=end
#!/usr/bin/env ruby
# frozen_string_literal: true

# Exit cleanly from an early interrupt
Signal.trap("INT") do
  Bundler.ui.debug("\n#{caller.join("\n")}") if defined?(Bundler)
  exit 1
(snip)
```

It fixed r60174 maybe. Can you try current HEAD(r60928)?

#11 - 11/30/2017 07:32 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Third Party's Issue to Assigned

#12 - 12/07/2017 08:51 PM - hone (Terence Lee)

- Status changed from Assigned to Closed

hsbt (Hiroshi SHIBATA) wrote:

> It fixed r60174 maybe. Can you try current HEAD(r60928)?

Hi hsbt (Hiroshi SHIBATA), thanks for looking into this. It's fixed as far as I can tell. I'm going to close it. I did a git bisect and r60171 fixed it. Thank you nobu (Nobuyoshi Nakada)!