CRuby adds an invalid header to bin/bundle (and others) which makes it unusable in Bash on Windows

08/20/2022 01:54 PM - Eregon (Benoit Daloze)

<table>
<thead>
<tr>
<th>Status</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>Backport:</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN</td>
</tr>
</tbody>
</table>

Description

Same as https://github.com/oneclick/rubyinstaller2/issues/299, but I figured it's extremely likely to be a bug in CRuby and not in RubyInstaller2.

The original user issue is: https://github.com/ruby/setup-ruby/issues/371.

bundle does not work in a Bash shell on Windows -- without an extra gem install bundler --, and the reason is building CRuby on Windows either does not produce a bin/bundle or it has the wrong permissions and the wrong start.

I downloaded all latest releases from https://github.com/ruby/setup-ruby/blob/master/windows-versions.json and extracted them (I'm on Linux FWIW).

```
$ ls
rubyinstaller-2.4.10-1-x64    rubyinstaller-2.6.10-1-x64    rubyinstaller-3.0.4-1-x64    rubyinstaller-head-x64
rubyinstaller-2.4.10-1-x64.7z rubyinstaller-2.6.10-1-x64.7z rubyinstaller-3.0.4-1-x64.7z rubyinstaller-head-x64.7z
rubyinstaller-2.5.9-1-x64    rubyinstaller-2.7.6-1-x64    rubyinstaller-3.1.2-1-x64    rubyinstaller-2.5.9-1-x64.7z
rubyinstaller-2.7.6-1-x64.7z rubyinstaller-3.1.2-1-x64.7z rubyinstaller-2.5.9-1-x64.7z
rubyinstaller-2.5.9-1-x64.7z
```

Of course only Ruby 2.7+ ships with Bundler, so for <=2.6 it's expected to be missing.

```
$ ls -l */bin/bundle
-rw-r--r--. 1 eregon eregon 707 Apr 19 22:22 rubyinstaller-3.1.2-1-x64/bin/bundle
-rw-r--r--. 1 eregon eregon 707 Apr 19 22:40 rubyinstaller-head-x64/bin/bundle
-rw-rw-rw- 1 eregon eregon 702 Aug 20 10:15 ruby-mingw/bin/bundle
-rw-rw-rw- 1 eregon eregon 829 Aug 20 11:09 ruby-mswin/bin/bundle
-rw-rw-rw- 1 eregon eregon 704 Aug 20 11:20 ruby-ucrt/bin/bundle
```

So only 3.1 and head have bin/bundle.

But those 2 bin/bundle do not have the executable bit set.

They also start like this which sounds invalid for Bash:

```
$ cat rubyinstaller-3.1.2-1-x64/bin/bundle
""|{{ "q<=-* ruby -*
"%~dp0ruby" -x %~f0* %*
@exit /b $ERRORLEVEL%
}};#{
bindir="$0%/*" #
exec "$bindir/ruby" "-x" "$0" "$1" #
}
}!/usr/bin/env ruby
# This file was generated by RubyGems.
...
```

On https://github.com/eregon/setup-ruby/runs/7843304711?check_suite_focus=true we can see 3.0, 3.1 and head fail for echo ~ && which -a bundle in bash.

08/25/2022
2.7 avoids the issue in that CI run because the Bundler version is considered too old by setup-ruby and so gem install bundler is done there.

**Needed fix**

In general, I think it is very important that CRuby does NOT modify files in bin/, and so that they are exactly the same as when RubyGems would write them when installing the corresponding gem.

This has been a problem not only here but also in these two other issues:

- gem install bundler fails on Windows, needs --force to workaround: [issues/5245](https://github.com/rubygems/rubygems/issues/5245)
- gem install bundler fails on --enable-load-relative Rubies: [issues/98](https://github.com/ruby/setup-ruby/issues/98#issuecomment-719950719)

tool/rbininstall.rb seems to be responsible for changing the bin/ files and therefore causing those bugs: [issues/5245](https://github.com/rubygems/rubygems/issues/5245)

Can we remove that?

Not that this issue cannot be solved in RubyGems, it's CRuby breaking bin/bundle (and others) in Bash on Windows.

**History**

#1 - 08/20/2022 02:20 PM - Eregon (Benoit Daloze)

Ah, maybe this cryptic header actually works for Bash too?

Then the problem is that CRuby adds it instead of RubyGems.

It should be RubyGems adding it (if it's necessary at all), so that bin/bundle is the same before and after gem install bundler and gem install bundler doesn't fail as mentioned above.

The missing executable permission is likely still an issue in CRuby.

#2 - 08/21/2022 04:40 AM - nobu (Nobuyoshi Nakada)

Eregon (Benoit Daloze) wrote in #note-1:

Ah, maybe this cryptic header actually works for Bash too?

The header is polyglot (triglot?) code.

For .bin/sh, the first "*" successes and the block after || is not executed, then in the next block, starting with { #, execs the ruby in the same directory with -x option, which direct to search #!ruby shebang line.

```ruby
""|| ""=> &q<-- ruby --
""-%-dp0ruby"" -x "%-f0 missing" %
@exit /b %ERRORLEVEL%

bindir="${O%/*}" #
exec "$bindir/ruby" "-x" "$0" "$@" >,
}
```

For cmd.exe, the first line is a comment (precisely a label which cannot use), then executes the ruby and exits.

```ruby
""|| ""=> &q<-- ruby --
""-%-dp0ruby"" -x "%-f0 missing" %
@exit /b %ERRORLEVEL%

bindir="${O%/*}" #
exec "$bindir/ruby" "-x" "$0" "$@" >,
}
```

For ruby, the first symbol literal is truthy, then the next hash literal is ignored till the end.

```ruby
""|| ""=> &q<-- ruby --
""-%-dp0ruby"" -x "%-f0 missing" %
@exit /b %ERRORLEVEL%

bindir="${O%/*}" #
exec "$bindir/ruby" "-x" "$0" "$@" >,
}
```
Then the problem is that CRuby adds it instead of RubyGems. It should be RubyGems adding it (if it's necessary at all), so that bin/bundle is the same before and after gem install bundler and gem install bundler doesn't fail as mentioned above.

The rest should be the content generated by RubyGems. The missing executable permission is likely still an issue in CRuby.

The rubyinstaller is for Windows, there is no concept of the executable permission. On Windows, bundle.bat should be executed.

#3 - 08/21/2022 11:11 AM - Eregon (Benoit Daloze)
@nobu (Nobuyoshi Nakada) Thank you for the explanation, quite clever code.

I think it is still an issue due to being fragile/inconsistent that this header only exists for bin/* files shipped with Ruby but not for bin/* files installed by RubyGems later.

The rubyinstaller is for Windows, there is no concept of the executable permission.
On Windows, bundle.bat should be executed.

Some people/users do use bash on Windows. Does bash look at bundle.bat?
$PATHEXT is .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC so I guess maybe yes?

I'm not quite sure what's wrong, I'm no Windows expert, but somehow bundle install/which bundle doesn't work on Windows without gem install bundler.

2.7 with gem install bundler:
https://github.com/eregon/setup-ruby/runs/7938304106?check_suite_focus=true
That works fine, because of the gem install bundler.

-rwxr-xr-x 1 runneradmin None 567 Aug 21 10:56 bundle
-rw-r--r-- 1 runneradmin None 41 Aug 21 10:56 bundle.bat
-rw-r--r-- 1 runneradmin None 672 Apr 19 20:22 bundle.cmd

Is -l in bash reports bundle as executable, and there is all 3 files.

bundle install in bash shell works: https://github.com/eregon/setup-ruby/runs/7938351861?check_suite_focus=true#step:26:1

3.0:
https://github.com/eregon/setup-ruby/runs/7938304140?check_suite_focus=true

-rw-r--r-- 1 runneradmin None 672 Apr 19 20:22 bundle.cmd

There is no bin/bundle file (and also no bundle.bat)! And bundle does not work, when used from a bash shell, I guess as a result of that.
And indeed in https://github.com/oneclick/rubyinstaller2/releases/download/RubyInstaller-3.0.4-1/rubyinstaller-3.0.4-1-x64.7z I also see no bin/bundle or bundle.bat.

Is it an issue of RubyInstaller2 or maybe an issue of the build process on Windows or of tool.rbinstall.rb maybe?

bundle install in bash shell fails: https://github.com/eregon/setup-ruby/runs/7938351901?check_suite_focus=true#step:26:24
D:\_temp\99c37f86-615f-4118-ab92-92b0a5b4f61f.sh: line 1: bundle: command not found

3.1:
https://github.com/eregon/setup-ruby/runs/7938304170?check_suite_focus=true

-rw-r--r-- 1 runneradmin None 707 Apr 19 20:22 bundle
-rw-r--r-- 1 runneradmin None 41 Apr 19 20:22 bundle.bat

There is bundle and bundle.bat but it does not work for which bundle. Maybe bundle.cmd is needed for which bundle? Note also how ls -l reports bundle is not executable here. Maybe because of the missing bundle.cmd?

bundle install in bash shell works: https://github.com/eregon/setup-ruby/runs/7938351950?check_suite_focus=true#step:26:24
But which bundle fails: https://github.com/eregon/setup-ruby/runs/7938351950?check_suite_focus=true#step:30:25

#4 - 08/21/2022 11:17 AM - Eregon (Benoit Daloze)
Another thought, maybe -ls -l in bash shows the x bit if the file starts with a shebang or is a native executable, but due to this header it's neither?

#5 - 08/21/2022 11:27 AM - Eregon (Benoit Daloze)
Here is another run, where none of the Rubies does a gem install Bundler.  
https://github.com/eregon/setup-ruby/runs/7938431059?check_suite_focus=true

2.6, 2.7 and 3.0 all fail bundle install in bash with bundle: command not found.  
All these only have bundle.cmd and no bundle/bundle.bat file.

For 3.1 and master, bundle install in bash works, but which bundle fails.  
These 2 have bundle (with the polyglot header) and bundle.bat.

As a side note, bundle install and where bundle in PowerShell does work for all of them.  
So it's an issue when using bundle from the bash shell on Windows.

#6 - 08/21/2022 11:38 AM - Eregon (Benoit Daloze)

I also noted from the run above, on 2.6-3.0:

```bash
# Output from 2.6
$ ls -l $(dirname $(which ruby))
```

```bash
total 3627
-rw-r--- 1 runneradmin None 672 Apr 19 20:27 bundle.cmd
-rw-r--- 1 runneradmin None 674 Apr 19 20:27 bundler.cmd
-rwxr-xr-x 1 runneradmin None 508 Apr 19 20:27 erb
-rw-r--- 1 runneradmin None 522 Apr 19 20:27 erb.cmd
-rwxr-xr-x 1 runneradmin None 54 Apr 19 20:27 gem
-rwxr-xr-x 1 runneradmin None 68 Apr 19 20:27 gem.cmd
-rwxr-xr-x 1 runneradmin None 508 Apr 19 20:27 irb
-rwxr-xr-x 1 runneradmin None 685 Apr 19 20:27 irb.cmd
-rwxr-xr-x 1 runneradmin None 502 Aug 21 11:20 rake
-rwxr-xr-x 1 runneradmin None 41 Aug 21 11:20 rake.bat
-rwxr-xr-x 1 runneradmin None 514 Apr 19 20:27 rake.cmd
-rwxr-xr-x 1 runneradmin None 656 Apr 19 20:27 rdoc
-rwxr-xr-x 1 runneradmin None 510 Apr 19 20:27 ri
-rwxr-xr-x 1 runneradmin None 652 Apr 19 20:27 ri.cmd
-rwxr-xr-x 1 runneradmin None 694 Apr 19 20:27 ridk.cmd
-rwxr-xr-x 1 runneradmin None 876 Apr 19 20:27 ridk.ps1
-rwxr-xr-x 1 runneradmin None 34304 Apr 19 20:27 ruby.exe
drwxr-xr-x 1 runneradmin None  0 Apr 19 20:27 ruby_builtin_dlls
-rwxr-xr-x 1 runneradmin None 34304 Apr 19 20:27 rubyw.exe
-rwxr-xr-x 1 runneradmin None 312 Apr 19 20:27 typeprof
-rwxr-xr-x 1 runneradmin None 3597824 Apr 19 20:27 x64-ucrt-ruby310.dll
```

vs on 3.1-master:

```bash
$ ls -l $(dirname $(which ruby))
```

```bash
total 4148
-rw-r--- 1 runneradmin None 707 Apr 19 20:22 bundle
-rwxr-xr-x 1 runneradmin None 709 Apr 19 20:22 bundler
-rwxr-xr-x 1 runneradmin None 668 Apr 19 20:22 erb
-rw-r--- 1 runneradmin None 41 Apr 19 20:22 erb.bat
-rwxr-xr-x 1 runneradmin None 504 Apr 19 20:22 gem
-rwxr-xr-x 1 runneradmin None 502 Apr 19 20:22 gem.cmd
-rwxr-xr-x 1 runneradmin None 68 Apr 19 20:22 irb
-rwxr-xr-x 1 runneradmin None 668 Apr 19 20:22 irb.bat
-rwxr-xr-x 1 runneradmin None 668 Apr 19 20:22 racc
-rwxr-xr-x 1 runneradmin None 674 Apr 19 20:22 racc.bat
-rwxr-xr-x 1 runneradmin None 674 Apr 19 20:22 rake
-rwxr-xr-x 1 runneradmin None 668 Apr 19 20:22 rake.bat
-rwxr-xr-x 1 runneradmin None 668 Apr 19 20:22 rbs
-rwxr-xr-x 1 runneradmin None 674 Apr 19 20:22 rbs.bat
-rwxr-xr-x 1 runneradmin None 674 Apr 19 20:22 rdoc
-rwxr-xr-x 1 runneradmin None 670 Apr 19 20:22 ri
-rwxr-xr-x 1 runneradmin None 670 Apr 19 20:22 ri.bat
-rwxr-xr-x 1 runneradmin None 694 Apr 19 20:22 ridk.cmd
-rwxr-xr-x 1 runneradmin None 876 Apr 19 20:22 ridk.ps1
-rwxr-xr-x 1 runneradmin None 35840 Apr 19 20:22 ruby.exe
drwxr-xr-x 1 runneradmin None  0 Apr 19 20:22 ruby_builtin_dlls
-rwxr-xr-x 1 runneradmin None 35840 Apr 19 20:22 rubyw.exe
-rwxr-xr-x 1 runneradmin None 312 Apr 19 20:22 typeprof
-rwxr-xr-x 1 runneradmin None 41 Apr 19 20:22 typeprof.bat
-rwxr-xr-x 1 runneradmin None 4140544 Apr 19 20:22 x64-ucrt-ruby310.dll
```

08/25/2022
So erb/gem/irb/rdoc/ri are all seen as executables on 2.6-3.0, and they all start with #!/usr/bin/env ruby (so no header).

rake has the header on 2.6-3.0 though and is seen as executable:

-rw-r--r-- 1 runneradmin None 672 Apr 19 20:22 bundle.cmd
...
-rwrxr-x-x 1 runneradmin None 519 Aug 21 11:19 rake
-rw-r--r-- 1 runneradmin None 41 Aug 21 11:19 rake.bat
-rw-r--r-- 1 runneradmin None 41 Apr 19 20:22 rake.cmd

So maybe the .cmd is needed to be seen as executable?

On 3.1-master, none of the bin/* are seen as executable and all have the header. Except for gem which is seen as executable and has no header.

#7 - 08/22/2022 01:54 AM - nobu (Nobuyoshi Nakada)

Does this work? https://github.com/nobu/ruby/tree/prolog_script

#8 - 08/22/2022 02:49 PM - Eregon (Benoit Daloze)

Unfortunately I can't build ruby from source on Windows myself to test, maybe @MSP-Greg (Greg L) can test that?

From above:

rake has the header on 2.6-3.0 though and is seen as executable:

Specifically:

```
$ cat rubyinstaller-3.0.4-1-x64/bin/rake
""||{""=>%q<-*-ruby-*-
@"%~dp0ruby"-x"%~f0"$*>
@exit /b %ERRORLEVEL%
};{
bindir="$\{0%/\}"#
exec "$bindir/ruby" "-x" "$0" "$@" 
>
},

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

So that seems to disprove my guess that the shebang makes the script seen as executable.

I think best would be:

- no triglot header or other header (since RubyGems doesn't generate it, and it causes various issues when RubyGems installs over such a file with header)
- For every bin/file: file, file.bat and file.cmd (on Windows only of course for the last two)

#9 - 08/22/2022 04:48 PM - MSP-Greg (Greg L)

@nobu (Nobuyoshi Nakada) - I will try that build, but I can't until later.

I don't know what the best approach is for Windows binstubs. Ruby-loco and RubyInstaller2 have always supplied 'portable' builds, not sure if that should be the case for a standard ruby/ruby build.

1. Not sure how far to go with getting things to run 'perfectly' when binstubs are ran in an install folder that is not the ENV Ruby.

2. Not sure about modifying ENV['Path'] in binstub scripts.

Re the current scripts, when run from a Windows cmd shell, I believe they worked. But, when run from a PowerShell window, they opened a separate cmd window...

#10 - 08/22/2022 07:31 PM - austin (Austin Ziegler)

MSP-Greg (Greg L) wrote in #note-9:

@nobu (Nobuyoshi Nakada) - I will try that build, but I can't until later.

I don't know what the best approach is for Windows binstubs. Ruby-loco and RubyInstaller2 have always supplied 'portable' builds, not sure if that should be the case for a standard ruby/ruby build.
1. Not sure how far to go with getting things to run 'perfectly' when binstubs are ran in an install folder that is not the ENV Ruby.

2. Not sure about modifying ENV['Path'] in binstub scripts.

Re the current scripts, when run from a Windows cmd shell, I believe they worked. But, when run from a PowerShell window, they opened a separate cmd window...

Do we need the multi-language header anymore with ruby -S binary? It seems to me that bin/bundle could be what it is now with only the Unix headers, and bin/bundle.bat and bin/bundle.cmd just call ruby -S bin/bundle or something like that?

I haven’t used Windows for a very long time, but it feels like this could be something that is part of the Ruby installation and possibly glommed onto binstub generation for bundler and/or Rubygems so that on systems that need .bat or .cmd files, those are added regardless because it ends up calling ruby -s bin/bundle. More or less.

#11 - 08/23/2022 04:52 PM - Eregon (Benoit Daloze)

I also noticed which rake does not work on Ruby 2.5+, but it does work on Ruby < 2.5:
https://github.com/ruby/setup-ruby/runs/7938245817?check_suite_focus=true

austin (Austin Ziegler) wrote in #note-10:

Do we need the multi-language header anymore with ruby -S binary? It seems to me that bin/bundle could be what it is now with only the Unix headers, and bin/bundle.bat and bin/bundle.cmd just call ruby -S bin/bundle or something like that?

ruby -S on CRuby doesn't actually look in that CRuby's bin/ dir first, so that's not helpful or better than just binary:

{on Linux, with `puts "3.0.3 bundle"` in ~/.rubies/ruby-3.0.3/bin/bundle}
$ which bundle
~/.rubies/ruby-3.0.3/bin/bundle
$ ./rubies/ruby-master/bin/ruby -S bundle --version
3.0.3 bundle
Bundler version 2.4.0.dev

On TruffleRuby and JRuby it does look in bin/ first and it's helpful for such cases.

Agreed we shouldn't need the multi-language header.
When using Bash on Windows, it's IMHO fine to expect the ruby used is first in PATH.

#12 - 08/23/2022 10:37 PM - MSP-Greg (Greg L)

@nobu (Nobuyoshi Nakada) & @Eregon (Benoit Daloze)

I just tried the prolog_script script branch. Wondering if the following example would be fine with everyone? It seems to work for me locally. If they're run as 'in path' commands, they work, if they're run from the folder of a 'non-system' Ruby, they run, although, since they don't set Path, any Ruby sub-processes hopefully will use Gem.ruby or something similar...

I tested the below with MSYS2 bash and Git bash.

Windows bash file 'top'

```bash
[bindir=$(dirname "$0")]
exec "$bindir/ruby" -x "$0" "$@
```

#!/usr/bin/env ruby
# This file was generated by RubyGems.
-- remaining code --

Windows *.cmd or *.bat binstub

```bash
@ECHO OFF
@"%~dp0ruby.exe" -x "%~dpn0" %*
```

I built a test, link is https://github.com/MSP-Greg/ruby-loco-test/releases/download/ruby-master/ruby-mswin.7z. It's the same as the builds used in CI.

Note that the bash scripts work the same as the Windows (*.bat, *.cmd) scripts. They launch with the ruby.exe in the same folder, not the ENV Ruby.

Finally, the ruby/ruby code isn't generating a 'gem' bash file.

#13 - 08/24/2022 05:08 PM - MSP-Greg (Greg L)
The Ruby build linked to above had incorrect bash scripts. Fixed as of now. Sorry, bad day yesterday, too many interruptions...

#14 - 08/25/2022 02:44 PM - nobu (Nobuyoshi Nakada)

MSP-Greg (Greg L) wrote in #note-12:

```
Windows bash file 'top'
{
bindir=$(dirname "$0")
exec "$bindir/ruby" "-x" "$0" "$@"
}
#!/usr/bin/env ruby
```

I'm not sure where the header came from. rbinstall.rb doesn't use dirname command. Maybe here?

```
Windows *.cmd or *.bat binstub

@ECHO OFF
@"%~dp0ruby.exe" -x "%~dpn0" %*
```

Nor @ECHO OFF.