

Ruby - Bug #14884

msys2 mingw32 'rake' has unexpected text at top of file

06/30/2018 08:19 AM - GardenTools (Garden Tools)

Status:	Feedback	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 2.5.1p57 (2018-03-29 revision 63029) [i386-mingw32]	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

'rake' from mingw-w64-i686-ruby 2.5.1-1 has what looks like fragments of a batch file and a shell script as the first few lines of what should be just shell that calls ruby. The shebang on line 9 clearly should be the first line. The file indicates it is auto generated, so it seems that some of the generation script has been echoed into this file rather than being run.

Ruby 2.5 installed from rubyinstaller.org installs a 'rake' file that suffers the same issue.

```
$ cat /mingw32/bin/rake
:"||{ "=> %q<*- ruby *-
@%~dp0ruby" -x "%~f0" %*
@exit /b %ERRORLEVEL%
};{#
bindir="%{0%/*}" #
exec "$bindir/ruby" "-x" "$0" "$@" #
>,
}
#!/usr/bin/env ruby
#
# This file was generated by RubyGems.
#
# The application 'rake' is installed as part of a gem, and
# this file is here to facilitate running it.
#

require 'rubygems'

version = ">= 0.a"

if ARGV.first
  str = ARGV.first
  str = str.dup.force_encoding("BINARY") if str.respond_to? :force_encoding
  if str =~ /\A_(.*)_z/ and Gem::Version.correct?($1) then
    version = $1
    ARGV.shift
  end
end

if Gem.respond_to?(:activate_bin_path)
load Gem.activate_bin_path('rake', 'rake', version)
else
gem "rake", version
load Gem.bin_path("rake", "rake", version)
end

rbinstall.rb around line 440 seems to be related to causing this.
```

History

#1 - 06/30/2018 08:28 AM - GardenTools (Garden Tools)

- Description updated

#2 - 06/30/2018 08:45 PM - MSP-Greg (Greg L)

@GardenTools

has unexpected text at top of file

One could say that other things are actually unexpected. I believe you have a rake.cmd file in your bin folder. Rename it, and name rake to rake.cmd. Run rake --help and you should have the correct console output. Assuming you're running from a cmd/ps shell. Looking at the first few lines of the script, it may also run with a bash shell...

I'm not sure why two files are used when one would do...

Greg

#3 - 07/02/2018 09:27 PM - GardenTools (Garden Tools)

Hi Greg,

Assuming you're running from a cmd/ps shell. Looking at the first few lines of the script, it may also run with a bash shell...

The difficulty is running from bash, which finds 'rake' (and becomes confused when it tries to run it). Calling rake.cmd does work but that then loses portability to *nix platforms of the invocation of rake. Chopping out everything above the #! makes the invocation of 'rake' work.

That said I see what you mean about the rest of the content of that file though - having the content of bin/rake simply be
#!/usr/bin/ruby require "rake" Rake.application.run

(the same as it is on linux) works for win32 ruby invoked from either cmd or msys2's bash.

#4 - 07/03/2018 12:35 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

It's polyglot code for sh and cmd.exe.
What error do you have with it actually?

#5 - 07/04/2018 06:50 AM - GardenTools (Garden Tools)

Error message is

/c/Ruby25/bin/rake: line 4: {#: command not found

#6 - 07/04/2018 03:11 PM - MSP-Greg (Greg L)

GardenTools (Garden Tools) wrote:

Error message is

/c/Ruby25/bin/rake: line 4: {#: command not found

Try changing line 4 from };{# to };{ # (add a space before #). Worked for me in a Git bash shell running rake --help.

I don't know what shells you and [@nobu \(Nobuyoshi Nakada\)](#) use...

#7 - 07/04/2018 11:20 PM - GardenTools (Garden Tools)

Try changing line 4 from };{# to };{ # (add a space before #). Worked for me in a Git bash shell running rake --help.

I don't know what shells you and [@nobu \(Nobuyoshi Nakada\)](#) use...

I'm using msys2's bash (with ruby installed via msys2's package manager). Putting a space there (or indeed removing those # altogether) seems to make it happy.

I note that all the other commands in the same directory have two files per command, but the multi-script-language part is in the .cmd file for those.

e.g. bin/gem simply starts with "#!/usr/bin/env ruby" whereas bin/gem.cmd opens with :""|{ ""=> %q<-- ruby --

All the other .cmd files have the };{# in them too, but unix-type shells never end up touching those lines. It doesn't seem possible to have both unix-type shells and cmd.exe execute the same script file in that way due to the extension.