

Ruby master - Feature #16952

Kernel.exec gives an incorrect error message when passed a script with a non-existing shebang

06/11/2020 06:13 PM - deivid (David Rodríguez)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<pre>\$ cat myscript.rb #!/bad/ruby/shebang puts "Hei!" \$ ruby -e 'Kernel.exec("myscript.rb")'</pre>	
Traceback (most recent call last): 1: from -e:1:in ` <main>' -e:1:in `exec': No such file or directory - myscript.rb (Errno::ENOENT)</main>	
From the error, I understand that the script is not there, but the script is clearly there.	
I would've expected something more similar to what you get if you try to run the script directly from the shell:	
<pre>\$./myscript.rb bash: ./myscript.rb: /bad/ruby/shebang: bad interpreter: No such file or directory</pre>	

History

#1 - 06/11/2020 07:51 PM - Eregon (Benoit Daloze)

I would think this is just the errno you get from `execve(2)`.
Ruby itself doesn't try to read the shebang.

So not sure Ruby can do much here.
`man execve` on Linux says:

ENOENT The file pathname or a script or ELF interpreter does not exist, or a shared library needed for the file or interpreter cannot be found.

OTOH I think Bash has its own handling and actually reads the shebang (I'm not sure why, and it makes it all more confusing).

#2 - 06/11/2020 08:31 PM - jeremyevans0 (Jeremy Evans)

- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN)
- ruby -v deleted (ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [x86_64-linux])
- Tracker changed from Bug to Feature

When I run your example in my shell (OpenBSD ksh), I get:

```
$ ./myscript.rb
ksh: ./myscript.rb: No such file or directory
$ ruby -e 'Kernel.exec("myscript.rb")'
Traceback (most recent call last):
 1: from -e:1:in `'
-e:1:in `exec': No such file or directory - myscript.rb (Errno::ENOENT)
```

[Eregon \(Benoit Daloze\)](#) is correct, examining the shebang appears to be a bash-specific behavior, not a general behavior. So I don't think this qualifies as a bug in Ruby. It could potentially be a useful feature, though.

#3 - 06/11/2020 08:40 PM - deivid (David Rodríguez)

In my opinion, the error message I'm being given is a bug, because it's telling me that a file that does exist, doesn't exist. Ruby should double check if the file exists before giving that error message, and if it does exist, give a more generic message that it's not clearly incorrect.