Ruby master - Bug #17419

`binding.eval` backtrace differente

12/21/2020 05:38 PM - vo.x (Vit Ondruch)

Status: Closed
Priority: Normal
Assignee: 
Target version: 

ruby -v: ruby 3.0.0rc1 (2020-12-20 master 8680ae9cbd) [x86_64-linux] 
Backport: 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN

Description
Can somebody elaborate why there is different output for Ruby 2.7 vs Ruby 3.0?

$ ruby -v
ruby 2.7.2p137 (2020-10-01 revision 5445e04352) [x86_64-linux]

$ ruby -e 'begin
binding.eval(%|raise "oops"|)
rescue Exception => exc
puts exc.backtrace
end
'-e:3:in `<main>'
-e:3:in `eval'
-e:3:in `<main>'

$ ruby -v
ruby 3.0.0rc1 (2020-12-20 master 8680ae9cbd) [x86_64-linux]

$ ruby -e 'begin
binding.eval(%|raise "oops"|)
rescue Exception => exc
puts exc.backtrace
end
'(eval):1:in `<main>'
-e:3:in `eval'
-e:3:in `<main>'

IOW the first line of backtrace is different. Is this expected? This is causing test failures:

https://github.com/rails/web-console/issues/301

Related issues:
Related to Ruby master - Bug #17160: Wrong exception backtrace
Rejected

Associated revisions
Revision 4a18cc7f - 12/22/2020 12:04 AM - mame (Yusuke Endoh)
NEWS.md: mention the behavior change of Binding#eval

[Bug #17419]

History
#1 - 12/21/2020 05:44 PM - vo.x (Vit Ondruch)
- Related to Bug #17160: Wrong exception backtrace added
#2 - 12/21/2020 07:10 PM - shevegen (Robert A. Heiler)

I think this is an intended change; it was explained via a link to a comment in another issue (I think by mame or someone else, where matz commented on reverting the behaviour). If I remember the reason, the changed backtrace confused many ruby users over the years, and it seemed better to go back to the state prior to the change, hopefully lessening the confusion in the process. :D

(Not sure if NEWS mentions this, but if not then perhaps a link should be there, or just a short explanation in NEWS itself.)

#3 - 12/21/2020 07:12 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

This change is expected, see #4352.

#4 - 12/22/2020 12:11 AM - mame (Yusuke Endoh)

Replacing b.eval(src) with b.eval(src, *b.source_location) will solve the issue.

However, #4352 focused on only Kernel#eval. I was not aware about the behavior change of Binding#eval. I've just added an explanation to NEWS.md at 4a18cc7f604c259415d933acd9fd00234733cab6.