

Ruby - Bug #17413

--backtrace-limit: wrong level counter

12/20/2020 06:49 PM - zverok (Victor Shepelev)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:		Backport: 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN
Description Having this test.rb: <pre>def inner raise 'test' end def outer inner end outer</pre> ...one might observe the following: \$ ruby test.rb test.rb:2:in `inner': test (RuntimeError) from test.rb:6:in `outer' from test.rb:9:in `<main>' # Print me 0 levels (only the report): \$ ruby --backtrace-limit=0 test.rb test.rb:2:in `inner': test (RuntimeError) ... 3 levels... # <== Umm, which 3? I saw 2 there! \$ ruby --backtrace-limit=1 test.rb test.rb:2:in `inner': test (RuntimeError) from test.rb:6:in `outer' ... 2 levels... # <== Which 2?.. # So, the next increase of the limit will leave just 1 hidden, right?.. \$ ruby --backtrace-limit=2 test.rb test.rb:2:in `inner': test (RuntimeError) from test.rb:6:in `outer' from test.rb:9:in `<main>' # Nope!		
Related issues: Related to Ruby - Feature #8661: Add option to print backtrace in reverse ord... <div>Closed</div>		

History

#1 - 12/21/2020 02:05 AM - mame (Yusuke Endoh)

- Related to Feature #8661: Add option to print backtrace in reverse order (stack frames first and error last) added

#2 - 12/21/2020 02:05 AM - mame (Yusuke Endoh)

Good catch! I will fix it soon. Thanks

#3 - 12/21/2020 09:05 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

Fixed at eee93bc9330cb069abb679a0a0d48a49b10f6a4e

#4 - 12/21/2020 10:41 AM - zverok (Victor Shepelev)

[@mame \(Yusuke Endoh\)](#) Thanks for the quick fix, now it works as expected!