

Ruby master - Feature #16684

Use the word "to" instead of "from" in backtrace

03/10/2020 04:00 PM - sawa (Tsuyoshi Sawada)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
The most-recent-call-last order of backtrace introduced by #8661 :	
<pre>def a; raise end def b; a end def c; b end c</pre>	
Current	
Traceback (most recent call last): 3: from foo.rb:4:in ` <main>' 2: from foo.rb:3:in `c' 1: from foo.rb:2:in `b' foo.rb:1:in `a': unhandled exception</main>	
is intuitive to me, and I hope it is retained. However, there are people complaining that it is confusing. I believe the unnaturalness is (at least partly) due to the fact that the word "from" is used, which made sense when backtrace was displayed in most-recent-call-first order,	
<pre>foo.rb:1:in `a': unhandled exception 1: from foo.rb:2:in `b' 2: from foo.rb:3:in `c' 3: from foo.rb:4:in `<main>'</main></pre>	
but not any more. Here, my understanding is that "from" means that the previous line was called from that line.	
I propose that, so long as the most-recent-call-last order is adopted, the word "to" should be used rather than "from", which would mean that the previous line leads to that line:	
Proposed 1	
Traceback (most recent call last): 3: to foo.rb:4:in ` <main>' 2: to foo.rb:3:in `c' 1: to foo.rb:2:in `b' foo.rb:1:in `a': unhandled exception</main>	
Or, as an alternative, if it looks unnatural to have "to" in the first line, and to lack one before the message line, we may put it at the end of a line:	
Proposed 2	
Traceback (most recent call last) 3: foo.rb:4:in ` <main>' to: 2: foo.rb:3:in `c' to: 1: foo.rb:2:in `b' to: foo.rb:1:in `a': unhandled exception</main>	
By using different words, it would become easier to understand the display order at a glance, and even by just looking at a single line.	
Related issues:	
Related to CommonRuby - Feature #8661: Add option to print backtrace in rever...	Closed

History

#1 - 03/10/2020 04:10 PM - sawa (Tsuyoshi Sawada)

- Description updated

#2 - 03/10/2020 04:15 PM - sawa (Tsuyoshi Sawada)

- Description updated

#3 - 03/10/2020 06:56 PM - sawa (Tsuyoshi Sawada)

- Description updated

#4 - 03/10/2020 11:33 PM - mame (Yusuke Endoh)

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

#5 - 03/11/2020 01:01 AM - shevegen (Robert A. Heiler)

I do not necessarily have a strong preference between "to" and "from". I am not a native english speaker either; maybe it is best if a native english speaker could share an opinion in regards to (perceived) differences, when comparing "to" and "from" here.

As for the backtrace situation: I think different ruby users and devs may have a different preference. I do not have a strong preference; I can adjust fine to the changed backtrace. I may perhaps slightly prefer the older one but I have no strong preference here really - whatever the ruby team wishes to use as the default. I do, however had, think that ultimately it would be best if these aspects could be controlled in some way - I think I used comparisons to that with warnings in the past, and Warning has been extended over the last months (or years). Perhaps in the long run, when it becomes more clear, we can look which use cases may be the most common for people to use, and support these (e. g. make the "best" as the default, but allow people to customize what they want to use, ideally, including things such as warnings, but also the display of the backtrace and so on; I think there was once a japanese ruby user who wrote that he was confused by the changed backtrace, so simple means to regulate that may be helpful, e. g. like the idea behind rubocop allowing customization for a project/team). But back to your original suggestion, I do not have a strong preference either way. Actually if it were up to me I may even use a much longer, more descriptive error reporting, if it can be made more useful. Something like some "report" tool or so that could tell things such as:

"On line 33 you possibly forgot an 'end'."

Or something like that ... it is actually quite difficult to provide good error messages that are easily understandable, without letting the user guess. :)

And I think the ruby core team and matz prefer shorter/more succinct variants, so this should, if anything, more be an option - but anyway, I digress. Don't want to distract from sawa's suggestion and should rather make a separate one in that case; consider this more like some loose thoughts about the topic. I remember that it can be really difficult to know what is going on if the backtrace strace is very long; like having to scroll up hundreds to thousand of lines, only to see the error on top was due to a single letter typo. That is also a reason why perhaps bottom-error reporting may seem more convenient, for quick results. But again, sorry - I digress.

#6 - 03/11/2020 10:47 AM - Eregon (Benoit Daloze)

I'm negative, I think to doesn't read well in this context.

They are still stack entries, coming *from* some file, line in some method.

#7 - 04/06/2020 11:04 AM - sawa (Tsuyoshi Sawada)

- Description updated

#8 - 04/10/2020 07:52 AM - matz (Yukihiro Matsumoto)

I don't think to is sufficient. I'd rather remove from altogether if there's no compatibility issue.

Should we remove them?

Matz.

#9 - 04/10/2020 07:58 AM - sawa (Tsuyoshi Sawada)

matz (Yukihiro Matsumoto) wrote in [#note-8](#):

I don't think to is sufficient. I'd rather remove from altogether if there's no compatibility issue.
Should we remove them?

Matz.

I agree. Removing from will make it at least better than as is now.