

Ruby trunk - Bug #8886

TracePoint API inconsistency when raise used

09/10/2013 08:03 PM - deivid (David Rodríguez)

Status: Closed	
Priority: Normal	
Assignee: matz (Yukihiro Matsumoto)	
Target version: 2.1.0	
ruby -v: ruby 2.0.0p247 (2013-06-27 revision 41674) [i686-linux]	Backport: 1.9.3: UNKNOWN, 2.0.0: UNKNOWN

Description

=begin

When raise command is used, the TracePoint API triggers the following events in the following order

1. RUBY_EVENT_C_CALL to the `raise` method
2. RUBY_EVENT_RAISE
3. RUBY_EVENT_C_RETURN

But what ruby actually does is

1. Push frame into the stack when calling the `raise` c method.
2. Pop frame from the stack
3. And after popping the frame, raise the exception.

As you can see, 2 and 3 are reversed and this messes up byebug and (I guess) other users of the API.

To illustrate I use a similar program as I used in (invalid) [#8538](#)

```
tp = TracePoint.trace do |tp|
  warn "%-8s %-11p" % [tp.event, tp.method_id]
end
raise "bang"
```

Actual output

```
c_return :trace
line    nil
c_call  :raise
c_call  :new
c_call  :initialize
c_return :initialize
c_return :new
c_call  :backtrace
c_return :backtrace
raise   nil
c_return :raise
test_bug.rb:4:in `': bang (RuntimeError)
```

Expected output

```
c_return :trace
line     nil
c_call   :raise
c_call   :new
c_call   :initialize
c_return :initialize
c_return :new
c_return :raise
c_call   :backtrace
c_return :backtrace
raise    nil
test_bug.rb:4:in `<main>': bang (RuntimeError)
```

I've made a patch that solves the issue and, as a result, the problems byebug is having. Please review and excuse me if I'm not being able to properly explain myself.

Thanks a lot!
=end

Associated revisions

Revision a4cbd641 - 12/11/2013 03:39 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): pop frame after setup exception. Patches by deivid (David Rodriguez). [Bug #8886]
- test/minitest/test_minitest_unit.rb: catch up this change.
- test/ruby/test_backtrace.rb: ditto.
- test/ruby/test_settracefunc.rb: ditto.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44133 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44133 - 12/11/2013 03:39 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): pop frame after setup exception. Patches by deivid (David Rodriguez). [Bug #8886]
- test/minitest/test_minitest_unit.rb: catch up this change.
- test/ruby/test_backtrace.rb: ditto.
- test/ruby/test_settracefunc.rb: ditto.

Revision 44133 - 12/11/2013 03:39 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): pop frame after setup exception. Patches by deivid (David Rodriguez). [Bug #8886]
- test/minitest/test_minitest_unit.rb: catch up this change.
- test/ruby/test_backtrace.rb: ditto.
- test/ruby/test_settracefunc.rb: ditto.

Revision 44133 - 12/11/2013 03:39 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): pop frame after setup exception. Patches by deivid (David Rodriguez). [Bug #8886]
- test/minitest/test_minitest_unit.rb: catch up this change.
- test/ruby/test_backtrace.rb: ditto.
- test/ruby/test_settracefunc.rb: ditto.

Revision 44133 - 12/11/2013 03:39 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): pop frame after setup exception. Patches by deivid (David Rodriguez). [Bug #8886]
- test/minitest/test_minitest_unit.rb: catch up this change.
- test/ruby/test_backtrace.rb: ditto.
- test/ruby/test_settracefunc.rb: ditto.

Revision 44133 - 12/11/2013 03:39 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): pop frame after setup exception. Patches by deivid (David Rodriguez). [Bug #8886]
- test/minitest/test_minitest_unit.rb: catch up this change.
- test/ruby/test_backtrace.rb: ditto.
- test/ruby/test_settracefunc.rb: ditto.

Revision 919f1438 - 12/11/2013 07:53 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): call c_return hook immediately after popping `raise` frame. Patches by deivid (David Rodriguez). [Bug #8886]
- test/ruby/test_settracefunc.rb: catch up this fix.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44139 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44139 - 12/11/2013 07:53 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): call c_return hook immediately after popping `raise` frame. Patches by deivid (David Rodriguez). [Bug #8886]
- test/ruby/test_settracefunc.rb: catch up this fix.

Revision 44139 - 12/11/2013 07:53 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): call c_return hook immediately after popping `raise` frame. Patches by deivid (David Rodriguez). [Bug #8886]
- test/ruby/test_settracefunc.rb: catch up this fix.

Revision 44139 - 12/11/2013 07:53 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): call c_return hook immediately after popping `raise` frame. Patches by deivid (David Rodriguez). [Bug #8886]
- test/ruby/test_settracefunc.rb: catch up this fix.

Revision 44139 - 12/11/2013 07:53 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): call c_return hook immediately after popping `raise` frame. Patches by deivid (David Rodriguez). [Bug #8886]
- test/ruby/test_settracefunc.rb: catch up this fix.

Revision 44139 - 12/11/2013 07:53 AM - ko1 (Koichi Sasada)

- eval.c (rb_raise_jump): call c_return hook immediately after popping `raise` frame. Patches by deivid (David Rodriguez). [Bug #8886]

- test/ruby/test_settracefunc.rb: catch up this fix.

History

#1 - 09/20/2013 03:42 PM - devid (David Rodríguez)

Ping. Could [ko1 \(Koichi Sasada\)](#) or someone review this and (hopefully) merge it? I've been using it for the last week without any issues.

Thanks!

#2 - 09/20/2013 03:51 PM - drbrain (Eric Hodel)

- Status changed from Open to Assigned

#3 - 09/30/2013 08:23 PM - ko1 (Koichi Sasada)

- Category set to core

- Target version set to 2.1.0

#4 - 10/10/2013 05:35 PM - ko1 (Koichi Sasada)

devid (David Rodríguez) wrote:

Expected output

```
c_return :trace
line     nil
c_call   :raise
c_call   :new
c_call   :initialize
c_return :initialize
c_return :new
c_return :raise
c_call   :backtrace
c_return :backtrace
raise    nil
test_bug.rb:4:in `<main>': bang (RuntimeError)
```

Could you tell me why it is expected?

:raise event occurs in raise method. So the current behavior works for me.

#5 - 10/11/2013 10:42 PM - devid (David Rodríguez)

- File `raise_exception_inside_raise_method.patch` added

I don't think ruby works like that right now. If `:raise` event occurs inside `raise` method, then when I print the backtrace of an exception, I should get the `raise` method as the first frame, just like when I do `Thread.current.backtrace_locations` I get `backtrace_locations` as the first frame.

In my first patch I tried to adapt `TracePoint` API's behaviour to ruby's behaviour, but I agree that `":raise` event occurs in `raise` method" behaviour makes more sense.

So I attach another patch to fix that.

Thanks [ko1 \(Koichi Sasada\)](#)!

#6 - 10/21/2013 10:45 PM - devid (David Rodríguez)

So, any opinions on this [ko1 \(Koichi Sasada\)](#)?

I'm handling this edge case in `byebug` manually so it's not a big deal but it'd be nice to get this fixed in ruby, either adapting the `TracePoint` API to ruby's behavior or changing how ruby behaves.

Thanks.

#7 - 12/11/2013 11:50 AM - ko1 (Koichi Sasada)

Sorry for long absence.

Your patch makes sense for me.
I'll apply it.

Thank you very much.

#8 - 12/11/2013 12:39 PM - ko1 (Koichi Sasada)

- Status changed from *Assigned* to *Closed*

- % Done changed from 0 to 100

This issue was solved with changeset [r44133](#).

David, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

-
- `eval.c (rb_raise_jump)`: pop frame after setup exception. Patches by `devid` (David Rodríguez). [Bug [#8886](#)]
 - `test/minitest/test_minitest_unit.rb`: catch up this change.
 - `test/ruby/test_backtrace.rb`: ditto.
 - `test/ruby/test_settracefunc.rb`: ditto.

#9 - 12/11/2013 04:06 PM - ko1 (Koichi Sasada)

- Status changed from Closed to Open
- Assignee changed from ko1 (Koichi Sasada) to matz (Yukihiro Matsumoto)
- Priority changed from Normal to 5

I committed it and rubyspec found an issue of incompatibility.

```
raise
#=>
ruby 2.1.0dev (2013-12-11 trunk 44136) [i386-mswin32_110]
t.rb:2:in raise': unhandled exception
from t.rb:2:in'

ruby 2.0.0p317 (2013-09-15 revision 42947) [i386-mswin32_110]
t.rb:2:in ``': unhandled exception
```

On backtrace, `raise' frame is available. I'm not sure it is acceptable or not.

One minor advantage of output `raise' frame on backtrace is, we can understand this error is occurred by raise method.

#10 - 12/11/2013 04:08 PM - ko1 (Koichi Sasada)

A bit strange for the following case?

```
raise("should not reach here")
#=>
ruby 2.1.0dev (2013-12-11 trunk 44136) [i386-mswin32_110]
t.rb:1:in raise': should not reach here (RuntimeError)
from t.rb:1:in'

#=>
ruby 2.0.0p317 (2013-09-15 revision 42947) [i386-mswin32_110]
t.rb:1:in ``': should not reach here (RuntimeError)
```

#11 - 12/11/2013 04:34 PM - ko1 (Koichi Sasada)

I asked Matz and he said "I prefer to remove `raise' line in backtrace".
So I will revert [r44133](#) and introduce the first patch.

#12 - 12/11/2013 04:41 PM - deivid (David Rodríguez)

It doesn't make a lot of difference to me, so if this is actually spec, I agree with keeping the current behaviour and just adapting the TracePoint API.

Thanks a lot Koichi! :)

#13 - 12/11/2013 04:53 PM - ko1 (Koichi Sasada)

- Status changed from Open to Closed

This issue was solved with changeset [r44139](#).
David, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- eval.c (rb_raise_jump): call c_return hook immediately after popping `raise` frame. Patches by deivid (David Rodriguez). [Bug [#8886](#)]
- test/ruby/test_settracefunc.rb: catch up this fix.

#14 - 12/11/2013 05:08 PM - ko1 (Koichi Sasada)

deivid (David Rodríguez) wrote:

It doesn't make a lot of difference to me, so if this is actually spec, I agree with keeping the current behaviour and just adapting the TracePoint API.

[r44139](#) makes it consistent with your patch. Thanks!

#15 - 12/11/2013 05:10 PM - deivid (David Rodríguez)

Great, thanks a lot!!

Files

tracepoint_raise.patch	2.44 KB	09/10/2013	deivid (David Rodríguez)
raise_exception_inside_raise_method.patch	3.83 KB	10/11/2013	deivid (David Rodríguez)