

Ruby master - Feature #8960

Add Exception#backtrace_locations

09/27/2013 08:11 PM - headius (Charles Nutter)

Status:	Assigned	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:		
Description		
All parties agreed this would be useful to add in https://bugs.ruby-lang.org/issues/7895 and ko1 suggested I file a feature against ruby-trunk. So here it is.		
I might be able to come up with a patch, but I'm not sure when. Help wanted.		
Can we agree this will go into 2.1?		
Related issues:		
Related to CommonRuby - Feature #7895: Exception#backtrace_locations to go wi...	Open	02/21/2013

Associated revisions

Revision 0aed7c68 - 12/13/2013 04:31 AM - ko1 (Koichi Sasada)

- error.c: add Exception#backtrace_locations. Now, there are no setter and independent from Exception#backtrace. [Feature #8960]
- eval.c (setup_exception): set backtrace locations for `bt_location` special attribute.
- vm_backtrace.c (rb_backtrace_to_location_ary): added.
- internal.h: ditto.
- test/ruby/test_backtrace.rb: add a test for Exception#backtrace_locations.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44170 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44170 - 12/13/2013 04:31 AM - ko1 (Koichi Sasada)

- error.c: add Exception#backtrace_locations. Now, there are no setter and independent from Exception#backtrace. [Feature #8960]
- eval.c (setup_exception): set backtrace locations for `bt_location` special attribute.
- vm_backtrace.c (rb_backtrace_to_location_ary): added.
- internal.h: ditto.
- test/ruby/test_backtrace.rb: add a test for Exception#backtrace_locations.

Revision 44170 - 12/13/2013 04:31 AM - ko1 (Koichi Sasada)

- error.c: add Exception#backtrace_locations. Now, there are no setter and independent from Exception#backtrace. [Feature #8960]
- eval.c (setup_exception): set backtrace locations for `bt_location` special attribute.
- vm_backtrace.c (rb_backtrace_to_location_ary): added.
- internal.h: ditto.
- test/ruby/test_backtrace.rb: add a test for Exception#backtrace_locations.

Revision 44170 - 12/13/2013 04:31 AM - ko1 (Koichi Sasada)

- error.c: add Exception#backtrace_locations. Now, there are no setter and independent from Exception#backtrace. [Feature #8960]
- eval.c (setup_exception): set backtrace locations for `bt_location` special attribute.
- vm_backtrace.c (rb_backtrace_to_location_ary): added.
- internal.h: ditto.
- test/ruby/test_backtrace.rb: add a test for Exception#backtrace_locations.

Revision 44170 - 12/13/2013 04:31 AM - ko1 (Koichi Sasada)

- error.c: add Exception#backtrace_locations. Now, there are no setter and independent from Exception#backtrace. [Feature #8960]
- eval.c (setup_exception): set backtrace locations for `bt_location` special attribute.
- vm_backtrace.c (rb_backtrace_to_location_ary): added.
- internal.h: ditto.
- test/ruby/test_backtrace.rb: add a test for Exception#backtrace_locations.

Revision 44170 - 12/13/2013 04:31 AM - ko1 (Koichi Sasada)

- error.c: add Exception#backtrace_locations. Now, there are no setter and independent from Exception#backtrace. [Feature #8960]
- eval.c (setup_exception): set backtrace locations for `bt_location` special attribute.
- vm_backtrace.c (rb_backtrace_to_location_ary): added.
- internal.h: ditto.
- test/ruby/test_backtrace.rb: add a test for Exception#backtrace_locations.

Revision 44170 - 12/13/2013 04:31 AM - ko1 (Koichi Sasada)

- error.c: add Exception#backtrace_locations. Now, there are no setter and independent from Exception#backtrace. [Feature #8960]
- eval.c (setup_exception): set backtrace locations for `bt_location` special attribute.
- vm_backtrace.c (rb_backtrace_to_location_ary): added.
- internal.h: ditto.
- test/ruby/test_backtrace.rb: add a test for Exception#backtrace_locations.

History

#1 - 09/30/2013 11:51 AM - ko1 (Koichi Sasada)

- Category set to core
- Assignee set to ko1 (Koichi Sasada)
- Target version set to 2.1.0

I'll try before preview-2.

JRuby already has a implementation and tests?

#2 - 09/30/2013 10:16 PM - headius (Charles Nutter)

JRuby has implementation but no tests. The test should look pretty much like Thread#backtrace_locations tests, though.

I can work on some tests if you work on impl for MRI.

#3 - 12/13/2013 12:19 PM - nobu (Nobuyoshi Nakada)

What will be returned from an exception set by Exception#set_backtrace in JRuby?

#4 - 12/13/2013 01:31 PM - ko1 (Koichi Sasada)

- Status changed from Open to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r44170.
Charles, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

-
- error.c: add Exception#backtrace_locations. Now, there are no setter and independent from Exception#backtrace. [Feature [#8960](#)]
 - eval.c (setup_exception): set backtrace locations for `bt_location` special attribute.
 - vm_backtrace.c (rb_backtrace_to_location_ary): added.
 - internal.h: ditto.
 - test/ruby/test_backtrace.rb: add a test for Exception#backtrace_locations.

#5 - 12/13/2013 01:32 PM - ko1 (Koichi Sasada)

- Status changed from Closed to Feedback
- Target version changed from 2.1.0 to 2.6
- % Done changed from 100 to 0

Now, I implemented half-baked feature (read-only backtrace locations).

How about to continue how to change it?

#6 - 12/30/2013 01:11 AM - headius (Charles Nutter)

ko1: I have no problem whatsoever with backtrace_locations being read-only now and forever. I'm glad to see this made it into 2.1 as at least a basic feature.

#7 - 08/17/2016 09:08 PM - ryannevell (Ryan Nevell)

- Assignee changed from ko1 (Koichi Sasada) to headius (Charles Nutter)

Is there any way to re-raise an exception and propagate `backtrace_locations`? When I want to trap an exception and provide a different exception class and error message, I'm used to doing something like:

```
rescue => e
  raise MyCustomException, "Error details", e.backtrace
```

But this is calling `set_backtrace`, leaving the resulting exception's `backtrace_locations` nil (on MRI 2.3.1) or an empty array (on jruby 9.1.2.0).

If only some exceptions will have a valid `backtrace_locations` array (matching the information in `backtrace`), it is difficult to write code that takes advantage of this feature.

#8 - 08/18/2016 01:44 AM - nobu (Nobuyoshi Nakada)

Ryan Nevell wrote:

Is there any way to re-raise an exception and propagate `backtrace_locations`? When I want to trap an exception and provide a different exception class and error message, I'm used to doing something like:

Nothing right now.

If only some exceptions will have a valid `backtrace_locations` array (matching the information in `backtrace`), it is difficult to write code that takes advantage of this feature.

Agree.

#9 - 11/25/2016 09:21 AM - shyouhei (Shyouhei Urabe)

- Assignee changed from headius (Charles Nutter) to ko1 (Koichi Sasada)

- Status changed from Feedback to Assigned

#10 - 11/25/2016 02:15 PM - shyouhei (Shyouhei Urabe)

We looked at this issue today. It turned out that ruby do not fully implement what was needed originally now. Reopened the ticket to assign ko1.