

Ruby master - Feature #15974

Warn in verbose mode on defining a finalizer that captures the object

07/02/2019 02:00 PM - chrisseaton (Chris Seaton)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
There's a really common mistake people make when using <code>define_finalizer</code> - they capture the object in the finalizer.	
https://www.mikeperham.com/2010/02/24/the-trouble-with-ruby-finalizers/	
This PR warns about that, when in verbose mode (it's too slow to do always.)	
https://github.com/ruby/ruby/pull/2264	

Associated revisions

Revision 928260c2 - 07/02/2019 07:05 PM - chrisseaton (Chris Seaton)

Warn in verbose mode on defining a finalizer that captures the object

[Feature #15974]

Closes: <https://github.com/ruby/ruby/pull/2264>

Revision 796eeb63 - 07/02/2019 07:22 PM - nobu (Nobuyoshi Nakada)

Fix up [Feature #15974]

- Fixed warning condition
- Fixed function signature
- Use ident hash

Revision 89cef1c5 - 07/03/2019 06:22 AM - nobu (Nobuyoshi Nakada)

Version guard for [Feature #15974]

Revision d0cd0866 - 07/03/2019 03:58 PM - nobu (Nobuyoshi Nakada)

Disable GC during `rb_objspace_reachable_object_p`

Try to fix CI breakage by [Feature #15974].

Revision 23c92b6f - 07/03/2019 07:01 PM - nobu (Nobuyoshi Nakada)

Revert self-referencing finalizer warning [Feature #15974]

It has caused CI failures.

- d0cd0866d82a58933e5dccc073c753c0c2ad4eb5

Disable GC during `rb_objspace_reachable_object_p`

- 89cef1c56b3a0f9c5e6ccc22a5044477a4fd16c1

Version guard for [Feature #15974]

- 796eeb6339952d92ae1b353d450c7883e589852d.

Fix up [Feature #15974]

- 928260c2a613bbdd4402c300e0bf86ae7562e52a.

Warn in verbose mode on defining a finalizer that captures the object

History

#1 - 07/02/2019 02:41 PM - ko1 (Koichi Sasada)

it makes sense.

#2 - 07/02/2019 03:35 PM - mame (Yusuke Endoh)

Matz has approved this proposal.

#3 - 07/02/2019 04:15 PM - shevegen (Robert A. Heiler)

Matz has approved this proposal.

\o/

#4 - 07/02/2019 04:50 PM - nobu (Nobuyoshi Nakada)

This is a good feature, but I don't think this should belong to the "spec" as seems an implementation detail.

#5 - 07/02/2019 07:05 PM - chriseaton (Chris Seaton)

I think Ruby Spec considers warnings to be user-visible features, so does normally spec them. This prompts alternative implementations like JRuby and TruffleRuby to implement the same warning so that the user experience is the same on all implementations.

[Eregon \(Benoit Daloze\)](#) will be able to give a definitive opinion on whether it should be spec'd or not.

#6 - 07/02/2019 07:26 PM - chriseaton (Chris Seaton)

- Status changed from Open to Closed

Applied in changeset [git|928260c2a613bbdd4402c300e0bf86ae7562e52a](https://github.com/ruby/ruby/pull/2264).

Warn in verbose mode on defining a finalizer that captures the object

[Feature [#15974](#)]

Closes: <https://github.com/ruby/ruby/pull/2264>

#7 - 07/02/2019 08:16 PM - Eregon (Benoit Daloze)

chriseaton (Chris Seaton) wrote:

I think Ruby Spec considers warnings to be user-visible features, so does normally spec them. This prompts alternative implementations like JRuby and TruffleRuby to implement the same warning so that the user experience is the same on all implementations.

[Eregon \(Benoit Daloze\)](#) will be able to give a definitive opinion on whether it should be spec'd or not.

Yes, exactly, it's a useful feature for other implementations to have too and it's user-visible so it's good to add to ruby/spec, similar to warnings.

#8 - 08/06/2019 11:50 PM - Dan0042 (Daniel DeLorme)

Question: shouldn't this be an error rather than a warning? Defining a finalizer that immortalizes its object and can never run seems like an outright bug, not something that you should merely warn about in verbose mode only.

#9 - 08/06/2019 11:55 PM - chriseaton (Chris Seaton)

Question: shouldn't this be an error rather than a warning?

No, because it's possible to have a reference from the finaliser to the object when the finaliser is created, but then to later clear that reference and the finaliser to work properly. The specs cover this.

However this PR was reverted due to some bugs (I wrote it in a couple of hours at a hackathon and it was merged then and there, so not surprising it wasn't really fully baked yet!) I hope to get it back in at some point.

#10 - 08/07/2019 12:41 AM - Dan0042 (Daniel DeLorme)

Ah, good point.

But maybe `ObjectSpace.define_finalizer(obj, aProc)` should raise an error if `aProc.binding.receiver.equal?(obj)`, because that's a reference that can never be cleared.

#11 - 08/07/2019 12:42 AM - chrisseton (Chris Seaton)

Yes we could do that, and enable it always as that'd be cheap.

#12 - 08/22/2020 02:54 AM - chrisseton (Chris Seaton)

A partial solution to this but with a better implementation is at <https://github.com/ruby/ruby/pull/3444>.