

Ruby trunk - Feature #15751

Add FrozenError#receiver

04/06/2019 07:06 AM - jeremyevans0 (Jeremy Evans)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>Similar to NameError#receiver, this returns the object on which the modification was attempted. This is useful as it can pinpoint exactly what is frozen, as in many cases when a FrozenError is raised, you cannot determine from the context which object is frozen that you expect not to be.</p> <p>I'm not sure that using name_err_receiver as the C function for FrozenError#receiver is acceptable, but it doesn't appear to cause problems.</p> <p>Users of the current rb_error_frozen C function will have to switch to using rb_error_frozen_object or the new rb_raise_frozen_error in order to set the receiver of the FrozenError.</p> <p>Patch to implement this feature is attached.</p>	

Associated revisions

Revision 39eadca7 - 05/26/2019 06:09 PM - jeremyevans (Jeremy Evans)

Add FrozenError#receiver

Similar to NameError#receiver, this returns the object on which the modification was attempted. This is useful as it can pinpoint exactly what is frozen. In many cases when a FrozenError is raised, you cannot determine from the context which object is frozen that you attempted to modify.

Users of the current rb_error_frozen C function will have to switch to using rb_error_frozen_object or the new rb_frozen_error_raise in order to set the receiver of the FrozenError.

To allow the receiver to be set from Ruby, support an optional second argument to FrozenError#initialize.

Implements [Feature #15751]

History

#1 - 04/06/2019 01:44 PM - nobu (Nobuyoshi Nakada)

- Description updated

Seems reasonable, but shouldn't FrozenError#initialize use a keyword argument as well as NameError#initialize?

#2 - 04/06/2019 06:04 PM - jeremyevans0 (Jeremy Evans)

- File Add-FrozenError-receiver-v2.patch added

nobu (Nobuyoshi Nakada) wrote:

Seems reasonable, but shouldn't FrozenError#initialize use a keyword argument as well as NameError#initialize?

NameError#initialize uses an additional positional argument for the name argument, and I figured that was easier as it is unlikely FrozenError#initialize would need to support additional arguments. However, I don't have strong feelings about it, so attached is a patch that switches to using a keyword argument.

#3 - 04/06/2019 07:14 PM - Eregon (Benoit Daloze)

NameError#initialize uses a keyword argument for receiver, so I think it makes some sense to be consistent.

OTOH, keyword arguments often imply extra processing, especially if the caller is not Ruby code, so I actually find the positional argument version simpler.

And indeed, I don't think extra arguments will be needed for FrozenError#initialize.

In the patch in `rb_raise_frozen_error`, `exc` is used for both the message and the exception object, could you use different variables? I find it confusing to use the same and re-assign it when there is no need.

#4 - 04/06/2019 10:09 PM - jeremyevans0 (Jeremy Evans)

Eregon (Benoit Daloze) wrote:

In the patch in `rb_raise_frozen_error`, `exc` is used for both the message and the exception object, could you use different variables? I find it confusing to use the same and re-assign it when there is no need.

Sure, I can make that change. Can you and nobu come to an agreement on whether to use the positional argument or the keyword argument so I know which patch to update?

#5 - 04/07/2019 03:04 AM - nobu (Nobuyoshi Nakada)

I haven't look C-API well, the additional argument would be ok.
`rb_raise_frozen_error` should be `rb_frozen_error_raise` like `rb_name_err_raise`?

As for the test, `assert_same` would be better for the equality of the receiver.

#6 - 04/07/2019 03:40 AM - jeremyevans0 (Jeremy Evans)

- *File Add-FrozenError-receiver-v3.patch added*

nobu (Nobuyoshi Nakada) wrote:

I haven't look C-API well, the additional argument would be ok.
`rb_raise_frozen_error` should be `rb_frozen_error_raise` like `rb_name_err_raise`?

As for the test, `assert_same` would be better for the equality of the receiver.

OK, v3 of the patch is attached, based off the initial patch with the positional argument, with the function renamed to `rb_frozen_error_raise` and using `assert_same` in the tests.

Eregon (Benoit Daloze) wrote:

In the patch in `rb_raise_frozen_error`, `exc` is used for both the message and the exception object, could you use different variables? I find it confusing to use the same and re-assign it when there is no need.

v3 patch includes this change as well.

#7 - 04/07/2019 05:07 PM - Eregon (Benoit Daloze)

The patch looks good to me.
I will let [nobu \(Nobuyoshi Nakada\)](#) commit it.

A nice extra would be specs in `spec/ruby/core/exception/frozen_error_spec.rb` (similar to `spec/ruby/core/exception/name_error_spec.rb`) but this is not a requirement.

#8 - 05/26/2019 06:10 PM - jeremyevans (Jeremy Evans)

- *Status changed from Open to Closed*

Applied in changeset [git39eadca76b48fc7841da688f6745e40897ec37ff](https://github.com/ruby/ruby/commit/git39eadca76b48fc7841da688f6745e40897ec37ff).

Add FrozenError#receiver

Similar to `NameError#receiver`, this returns the object on which the modification was attempted. This is useful as it can pinpoint exactly what is frozen. In many cases when a `FrozenError` is raised, you cannot determine from the context which object is frozen that you attempted to modify.

Users of the current `rb_error_frozen` C function will have to switch to using `rb_error_frozen_object` or the new `rb_frozen_error_raise` in order to set the receiver of the `FrozenError`.

To allow the receiver to be set from Ruby, support an optional second argument to `FrozenError#initialize`.

Implements [Feature [#15751](#)]

Files

Add-FrozenError-receiver.patch	6.48 KB	04/06/2019	jeremyevans0 (Jeremy Evans)
Add-FrozenError-receiver-v2.patch	6.66 KB	04/06/2019	jeremyevans0 (Jeremy Evans)
Add-FrozenError-receiver-v3.patch	6.49 KB	04/07/2019	jeremyevans0 (Jeremy Evans)