# Ruby trunk - Bug #13564

## Exception message management

05/15/2017 07:30 AM - ko1 (Koichi Sasada)

| | | | |
|---|---|---|---|
| **Status:** | Open | | |
| **Priority:** | Normal | | |
| **Assignee:** | matz (Yukihiro Matsumoto) | | |
| **Target version:** | | | |
| **ruby -v:** | | **Backport:** | 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN |

**Description**

We can modify Exception#message if given String is not frozen.
Should we continue this specification?

Now, we can modify exception message with message string modification.

```
begin
  raise 'foo'
ensure
  $!.message.replace 'bar' #=> Exception with message "bar"
end
```

However, if we pass frozen string, it is not allowed.

```
begin
  raise 'foo'.freeze
ensure
  $!.message.replace 'bar' #=> `replace': can't modify frozen String (RuntimeError)
end
```

Furthermore, # frozen_string_literal: true freeze all of string literals.

```
# frozen_string_literal: true

begin
  raise 'foo'
ensure
  $!.message.replace 'bar' #=> `replace': can't modify frozen String (RuntimeError)
end
```

## Background and motivation

I want to add Exception message on ensure clause like the code in previous section. Just now, we need to re-raise another exception (with raise($!.class, new_msg, $!.backtrace)). I tried to modify $!.message and it works on small script. However, I try it on production (*1), it doesn't work because of frozen_string_literal: true.

*1: https://svn.ruby-lang.org/cgi-bin/viewvc.cgi/trunk/test/rubygems/test_gem_gem_runner.rb?r1=58723&r2=58722&pathrev=58723

I think current behavior (specification?) is easy to misusing.

## Ideas

- (1) To prevent such behavior
  - (1-1) Freeze message strings at initialize
  - (1-2) Return copy string at Exception#message
- (2) Provide Exception#message =
  - And (1-1) or (1-2)
- (3) Allow such behavior. If a frozen message is given, dup it and set as modifiable.

**History**

**#1 - 05/15/2017 11:31 AM - Eregon (Benoit Daloze)**

I think using Exception#cause for this would be a better way to address this problem.
However, there is a long-standing bug of the cause not being shown in Exception#inspect and neither by the top-level handler:
https://bugs.ruby-lang.org/issues/9918

ko1 (Koichi Sasada): Could you share your use-case? Modifying an exception message in ensure seems unusual to me.
In the test_gem_gem_runner.rb, it seems rescue Exception would be more intuitive to handle this (but it has the same problem about modifying the message).

Otherwise I think 1-1 + 2 is the best compromise.

**#2 - 05/16/2017 02:02 AM - ko1 (Koichi Sasada)**

On 2017/05/15 20:31, eregontp@gmail.com wrote:

> I think using Exception#cause for this would be a better way to address this problem.
> However, there is a long-standing bug of the cause not being shown in Exception#inspect and neither by the top-level handler:
> https://bugs.ruby-lang.org/issues/9918

I agree it is one solution. However, to make sure transparency (for
rescue clause which catch the exception later) we need to provide same
error class ($!.class).

> ko1 (Koichi Sasada): Could you share your use-case? Modifying an exception message in ensure seems unusual to me.
> In the test_gem_gem_runner.rb, it seems rescue Exception would be more intuitive to handle this (but it has the same problem about modifying the message).

My usage is a bit strange. I want to know the status about just before
suspicious code (require 'rubygems/gem_runner') and just after this
line if $! is not nil. Usually we can show such information on STDERR
but test framework (test-all with parallel option) hides all of STDERR
output so that we need to show via Exception message.

I think such usage is not so frequent so that

> Otherwise I think 1-1 + 2 is the best compromise.

I think (1) without (2) (with [Feature #9918]) is acceptable.

Thanks,
Koichi
--
// SASADA Koichi at atdot dot net

**#3 - 07/14/2017 09:01 AM - matz (Yukihiro Matsumoto)**

I understand the principle. But I think it's a programmer's fault to modify the string.
I don't think it's worth prohibiting (and making implementation more complex).

Matz.

**#4 - 07/14/2017 09:20 AM - naruse (Yui NARUSE)**

Now we have Exception#cause.
Therefore it should be

```
begin
  raise 'foo'
ensure
  raise 'bar'
end
```