Ruby 1.8 - Bug #4578

Fixnum.freeze not frozen?

04/15/2011 08:22 AM - rbevers (Russell Bevers)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
ruby -v:	-

Description

=begin
Using ruby 1.8.7 p334

a = 5; a.freeze

Expect: true == a.frozen?
Result: false == a.frozen?

I realize Fixnum is already immutable, however consider the following code:

class Demanding
attr_reader :important

```
def initialize(important)
    @important = important
    @important.freeze
end
```

end

Demanding wants to be able to rely on identifier not changing over its lifespan. I could make it demanding with some other mechanism such as:

- Force important to be a particular type
- Force important to respond_to? particular messages during initialization

These are all slippery and don't address the concern of mutability of important objects. Freeze seemed like the obvious solution, but it's probably best for me to just take a deep copy of important for Demanding's own use.

In the meantime, I thought I should share that freeze is not working for all objects in Ruby 1.8.7 p334.
=end

Related issues:

Related to Ruby trunk - Feature #6936: Forbid singleton class and instance va...	**Closed**	**08/27/2012**

History

#1 - 04/15/2011 09:34 AM - shyouhei (Shyouhei Urabe)

=begin
You agree integers are immutable. Immutable objects have no states by definition, OK?

The frozenness is a state. You should not care about.
=end

#2 - 04/15/2011 03:23 PM - judofyr (Magnus Holm)

=begin
Isn't technically all Fixnums "frozen" (because you can't modify the state)?

// Magnus Holm

On Fri, Apr 15, 2011 at 02:34, redmine@ruby-lang.org wrote:

> Issue #4578 has been updated by Shyouhei Urabe.

You agree integers are immutable. Immutable objects have no states by definition, OK?

The frozenness is a state. You should not care about.

Bug [#4578](http://redmine.ruby-lang.org/issues/4578): Fixnum.freeze not frozen?
http://redmine.ruby-lang.org/issues/4578

Author: Russell Bevers
Status: Open
Priority: Normal
Assignee:
Category:
Target version:
ruby -v: ruby 1.8.7 (2011-02-18 patchlevel 334) [i686-darwin10.7.0]

Using ruby 1.8.7 p334

a = 5; a.freeze

Expect: true == a.frozen?
Result: false == a.frozen?

I realize Fixnum is already immutable, however consider the following code:

class Demanding
attr_reader :important

def initialize(important)
@important = important
@important.freeze
end
end

Demanding wants to be able to rely on identifier not changing over its lifespan. I could make it demanding with some other mechanism such as:

- Force important to be a particular type
- Force important to respond_to? particular messages during initialization

These are all slippery and don't address the concern of mutability of important objects. Freeze seemed like the obvious solution, but it's probably best for me to just take a deep copy of important for Demanding's own use.

In the meantime, I thought I should share that freeze is not working for all objects in Ruby 1.8.7 p334.

--
http://redmine.ruby-lang.org

=end

#3 - 04/15/2011 03:23 PM - matz (Yukihiro Matsumoto)

- ruby -v changed from ruby 1.8.7 (2011-02-18 patchlevel 334) [i686-darwin10.7.0] to -

=begin
Hi,

In message "Re: [ruby-core:35767] Re: [Ruby 1.8 - Bug [#4578](#)] Fixnum.freeze not frozen?"
on Fri, 15 Apr 2011 14:31:53 +0900, Magnus Holm judofyr@gmail.com writes:

|Isn't technically all Fixnums "frozen" (because you can't modify the state)?

Not really, since fixnums could have instance variables.

Thus, theoretically, there's room for freezing fixnums. I don't think it's worth the cost though.

                            matz.

=end

#4 - 04/20/2011 07:23 AM - headius (Charles Nutter)

=begin
On Fri, Apr 15, 2011 at 12:37 AM, Yukihiro Matsumoto matz@ruby-lang.org wrote:

> Hi,
>
> In message "Re: [ruby-core:35767] Re: [Ruby 1.8 - Bug #4578] Fixnum.freeze not frozen?"
>   on Fri, 15 Apr 2011 14:31:53 +0900, Magnus Holm judofyr@gmail.com writes:
>
> |Isn't technically all Fixnums "frozen" (because you can't modify the state)?
>
> Not really, since fixnums could have instance variables.
>
> Thus, theoretically, there's room for freezing fixnums. I don't think
> it's worth the cost though.

I guess you mean the cost in MRI. In JRuby, Fixnums *could* be frozen
(we don't enable it to match MRI), but the same value is not
guaranteed to be the same object throughout the system. Tradeoffs!

- Charlie =end

#5 - 04/27/2011 01:09 AM - rbevers (Russell Bevers)

=begin
BTW, freeze & frozen? work fine on Fixnums in 1.9.x on my platform. I agree that it seems unnecessary to correct in 1.8.x. I can move to 1.9.
=end