

Ruby master - Bug #2221

lib/delegate: freeze has odd effects

10/16/2009 03:43 PM - marcandre (Marc-Andre Lafortune)

Status: Closed	
Priority: Normal	
Assignee:	
Target version: 1.9.2	
ruby -v: ruby 1.9.2dev (2009-10-13 trunk 25317) [x86_64-darwin10.0.0]	Backport:
Description =begin Freeze doesn't quite freeze, and can't be cloned: require 'delegate' d = SimpleDelegator.new([1,2,3]) d.freeze d[0] = :hello; d # ==> [:hello, 2, 3], I expected a runtime error d.clone # ==> RuntimeError, I expected some kind of copy of d Maybe #freeze and #frozen should be forwarded: diff --git a/lib/delegate.rb b/lib/delegate.rb index 57c479c..f4bd453 100644 --- a/lib/delegate.rb +++ b/lib/delegate.rb @@ -115,7 +115,7 @@ # implementation, see SimpleDelegator. # class Delegator • [:to_s,:inspect,::~,::~,::].each do m • [:to_s,:inspect,::~,::~,::freeze, ::frozen?].each do m undef_method m end But maybe the Delegator itself should also be frozen? I mean should a call to #setobj raise a runtime error or not? Also, there are other methods that should probably be forwarded (eql?, hash, etc...); see issue #1333 . =end	
Related issues: Is duplicate of Ruby master - Bug #2679: rubyspec: SimpleDelegator when froze... Closed	

History

#1 - 10/17/2009 04:52 AM - marcandre (Marc-Andre Lafortune)

=begin
Hi,

On Fri, Oct 16, 2009 at 7:24 AM, Yukihiro Matsumoto matz@ruby-lang.org wrote:

I think both should be frozen. I checked in the patch. I'd like to see if it works (or not).

Indeed, I thought about it and came to the same conclusion when thinking of other non-trivial classes that would extend Delegate. It was too cold outside my bed though :-)

(a) Just to be sure: SimpleDelegator([1,2,3].freeze).frozen? should return false, right?

(b) Still can't clone it, though.

Is there a nice way to avoid the exception? The best I can think of is the inelegant:

```
diff --git a/lib/delegate.rb b/lib/delegate.rb
index 5d25632..5ef9c1c 100644
```

```
--- a/lib/delegate.rb
+++ b/lib/delegate.rb
@@ -188,11 +188,14 @@ class Delegator
setobj(obj)
end
```

- alias_method :**original_dup**, :dup
- private :**original_dup** + # Clone support for the object returned by __getobj__. def clone
- new = super
- new = frozen? ? **original_dup** : super new.**setobj**(getobj.clone)
- new
- frozen? ? new.freeze : new end # Duplication support for the object returned by __getobj__. def dup

=end

#2 - 10/18/2009 10:40 AM - matz (Yukihiko Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:26118] [Bug #2221] lib/delegate: freeze has odd effects"
on Fri, 16 Oct 2009 15:43:40 +0900, Marc-Andre Lafortune redmine@ruby-lang.org writes:

|Freeze doesn't quite freeze, and can't be cloned:

|But maybe the Delegator itself should also be frozen? I mean should a call to #**setobj** raise a runtime error or not?

I think both should be frozen. I checked in the patch. I'd like to see if it works (or not).

|Also, there are other methods that should probably be forwarded (eq!?, hash, etc...); see issue [#1333](#).

I have tried once, but code became vary clumsy, mostly because no constant can be accessed or initialized from BasicObject.

```
matz.
```

=end

#3 - 04/17/2010 12:09 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

=begin
Hi,

This ticket was duplicated by [#2679](#).

After that, lib/delegate.rb has been changed bit by bit.

As a result of matz and nobu's effort, this seems to work now.

--

Yusuke Endoh mame@tsg.ne.jp

=end