# Ruby - Bug #6408

## DelegateClass#eql? and <=> don't work as expected

05/07/2012 04:48 AM - tenderlovemaking (Aaron Patterson)

| | | | |
|---|---|---|---|
| **Status:** | Closed | | |
| **Priority:** | Normal | | |
| **Assignee:** | tenderlovemaking (Aaron Patterson) | | |
| **Target version:** | | | |
| **ruby -v:** | ruby 2.0.0dev (2012-05-06 trunk 35548) [x86_64-darwin11.3.0] | **Backport:** | 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN |

**Description**

It seems these two methods aren't delegating to the delegate object when compared against itself.

I've attached a patch with tests and a fix.  It seems nobody is the maintainer for delegate.rb (according to the wiki), so if nobody objects, I will apply this patch.

**Related issues:**

| | | |
|---|---|---|
| Has duplicate Ruby - Bug #7045: DelegateClass array subtraction | **Closed** | **09/22/2012** |

### History

#### #1 - 05/07/2012 07:15 AM - marcandre (Marc-Andre Lafortune)

Hi,

In your patch, for both if obj.equal? self, shouldn't it be if obj.is_a? Delegator?

For many classes that define their comparison operators, this might not be sufficient if you want comparison operators to work well, as they typically will first check on the class of the object to compare with.

Defining is_a? to return true for either Delegator or the class delegated to might be more helpful.

tenderlovemaking (Aaron Patterson) wrote:

> It seems these two methods aren't delegating to the delegate object when compared against itself.
>
> I've attached a patch with tests and a fix.  It seems nobody is the maintainer for delegate.rb (according to the wiki), so if nobody objects, I will apply this patch.

#### #2 - 05/07/2012 07:24 AM - marcandre (Marc-Andre Lafortune)

marcandre (Marc-Andre Lafortune) wrote:

> Hi,
>
> In your patch, for both if obj.equal? self, shouldn't it be if obj.is_a? Delegator?

Or if you did mean obj.equal? self, then you can return true / 0, at least that what != and == are doing. Better not delegate to NaN :-)

#### #3 - 05/07/2012 08:23 AM - Anonymous

*- File noname added*

On Mon, May 07, 2012 at 07:15:34AM +0900, marcandre (Marc-Andre Lafortune) wrote:

> Issue #6408 has been updated by marcandre (Marc-Andre Lafortune).
>
> Hi,
>
> In your patch, for both if obj.equal? self, shouldn't it be if obj.is_a? Delegator?
>
> For many classes that define their comparison operators, this might not be sufficient if you want comparison operators to work well, as they typically will first check on the class of the object to compare with.
>
> Defining is_a? to return true for either Delegator or the class delegated to might be more helpful.

I was thinking that too, but the current implementation of != and == don't do the is_a? check.  Maybe those should be changed?

Anyway, I don't know about other changes, but the failing tests I have in this patch are causing issues for some rails users:

https://github.com/rails/rails/issues/5974

I feel like we probably need better tests surrounding the "correct" behavior of DelegateClass, but I'd rather not shave that yak at the moment and just fix the issues we're having today. :)

--
Aaron Patterson
http://tenderlovemaking.com/

#### #4 - 05/07/2012 08:34 AM - marcandre (Marc-Andre Lafortune)

Hi,

> I was thinking that too, but the current implementation of != and == don't do the is_a? check.  Maybe those should be changed?
>
> Anyway, I don't know about other changes, but the failing tests I have in this patch are causing issues for some rails users:
>
> https://github.com/rails/rails/issues/5974
>
> I feel like we probably need better tests surrounding the "correct" behavior of DelegateClass, but I'd rather not shave that yak at the moment and just fix the issues we're having today. :)

Sounds good. In any case, +1 from me.

== and != should also be modified to call self == self and self != self in case other.equal?(self), even though the only object I know of that is not == to itself is NaN...

#### #5 - 05/07/2012 09:53 AM - jeremyevans (Jeremy Evans)

On 05/07 07:15, marcandre (Marc-Andre Lafortune) wrote:

> Hi,
>
> In your patch, for both if obj.equal? self, shouldn't it be if obj.is_a? Delegator?
>
> For many classes that define their comparison operators, this might not be sufficient if you want comparison operators to work well, as they typically will first check on the class of the object to compare with.
>
> Defining is_a? to return true for either Delegator or the class delegated to might be more helpful.

When I use DelegateClass, it's often because I want a proxy object that mostly acts like the given class, but is specifically not treated as the given class (in terms of the is_a? and === methods).

Changing this behavior would likely break existing code.  So if this to be considered, it should probably not change until 3.0 (unless matz makes an exception for this case).

Jeremy

#### #6 - 05/07/2012 10:20 AM - marcandre (Marc-Andre Lafortune)

Hi,

jeremyevans (Jeremy Evans) wrote:

> Changing this behavior would likely break existing code.  So if this to be considered, it should probably not change until 3.0 (unless matz makes an exception for this case).

I had misunderstood that the goal was to make equality symmetric. In any case, you're right, I don't think it should be considered.

#### #7 - 05/17/2012 12:10 AM - mame (Yusuke Endoh)

*- Status changed from Open to Assigned*

*- Assignee set to tenderlovemaking (Aaron Patterson)*

*- Target version set to 3.0*

Hello, Aaron

What do you think about Jeremy's opinion?

I'm just wondering but why do you want to delegate #eql? ?
I guess that is because you are inserting Delegate objects
to a Hash.  Such a code is still dangerous even if the
patch is applied:

```
require "delegate"

class Foo; end
class Bar < DelegateClass(Foo); end
foo = Foo.new
bar = Bar.new(foo)

p foo.eql?(foo)  #=> true
p bar.eql?(foo)  #=> true
p bar.eql?(bar)  #=> false (this returns true with your patch)

p foo.eql?(bar)  #=> false (this is NOT fixed)

h = { bar => 42 }
p h[foo] #=> nil, not 42
```

I think it is difficult to "fix."

The same holds for #<=>.
You should not sort an Array that includes Delegate objects.

--
Yusuke Endoh mame@tsg.ne.jp

#### #8 - 05/20/2012 05:29 AM - Anonymous

*- File noname added*

On Thu, May 17, 2012 at 12:10:56AM +0900, mame (Yusuke Endoh) wrote:

> Issue #6408 has been updated by mame (Yusuke Endoh).
>
> Status changed from Open to Assigned
> Assignee set to tenderlovemaking (Aaron Patterson)
> Target version set to 3.0
>
> Hello, Aaron
>
> What do you think about Jeremy's opinion?

I agree with Jeremy's opinion.  I think that changing to is_a? would
probably be be better, but I think my patch should be applied for Ruby
2.0.

> I'm just wondering but why do you want to delegate #eql? ?
> I guess that is because you are inserting Delegate objects
> to a Hash.  Such a code is still dangerous even if the
> patch is applied:
>
> require "delegate"
>
> class Foo; end
> class Bar < DelegateClass(Foo); end
> foo = Foo.new
> bar = Bar.new(foo)
>
> p foo.eql?(foo)  #=> true

> p bar.eql?(foo)  #=> true
> p bar.eql?(bar)  #=> false (this returns true with your patch)

Yes, this is the bug I'm trying to fix.

> p foo.eql?(bar)  #=> false (this is NOT fixed)

Yes, you are correct.  I'm not sure how or if we should fix this.

> h = { bar => 42 }
> p h[foo] #=> nil, not 42
>
> I think it is difficult to "fix."

It's difficult to "fix" 100%, yes.  However, we can at least reduce our broken windows. :-)

> The same holds for #<=>.
> You should not sort an Array that includes Delegate objects.

I agree.  The problem is that people can create delegate objects, then feed the delegate object to third party code (say some gem, or some library in stdlib) and expect it to work.

I think we should either make it work as best we can, or remove DelegateClass from stdlib.  It should not be a requirement that DelegateClass users understand how all third party code interacts with their delegate object (I think).

--
Aaron Patterson
http://tenderlovemaking.com/

#### #9 - 06/23/2016 03:52 AM - ioquatix (Samuel Williams)

I was bitten by something similar, expecting equal? to work correctly between non-delegate and delegate instances. Where are we at with this patch?

#### #10 - 09/03/2019 01:05 AM - jeremyevans0 (Jeremy Evans)

*- Status changed from Assigned to Closed*

Delegate now handles eql? after changes from #12684.  However, it always returns true if the object is the same:

```
nan = 0/0.0
nan.eql? nan # false
s = SimpleDelegator.new(nan)
s.eql? s # true
```

I'm not sure it is worth it to add explicit support for making it so eql? can return false if the target doesn't consider itself to be equal to itself.

As mame mentioned, <=> is for sorting, and it probably isn't helpful to have a sorting method that handles multiple copies of the same delegate object in an array, but not multiple different delegates of the same target in an array.

Considering the eql? issue has been fixed, which was the underlying cause of the Rails issue that caused this ticket to be filed, I think this is safe to close.  If I'm wrong, please reopen.

### Files

| | | | |
|---|---|---|---|
| fix.patch | 1.45 KB | 05/07/2012 | tenderlovemaking (Aaron Patterson) |
| noname | 500 Bytes | 05/07/2012 | Anonymous |
| noname | 500 Bytes | 05/20/2012 | Anonymous |