Ruby - Bug #7768

Inherited Array class missing

02/01/2013 09:43 PM - england (Roman Ivanilov)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
ruby -v:	1.9	**Backport:**

Description

Hello. I apologize if I missed something.
I found strange behavior in ruby 1.9:

```
class Custom < Array; end
Custom.new(0){|i| i + 1}.uniq.class # => Array
Custom.new(2){|i| i + 1}.uniq.class # => Custom
```

while in 1.8 it works just as I expected.

```
class Custom < Array; end
Custom.new(0){|i| i + 1}.uniq.class # => Custom
Custom.new(2){|i| i + 1}.uniq.class # => Custom
```

- it is actual not only for the uniq method.
- tested with ree-1.8.7-2010.02, ruby-1.9.2-p290, ruby-1.9.3-p375, ruby-1.9.3-p125

Any bug here?

Related issues:

Related to Ruby - Bug #7625: Array□□□□□□□□□□□□compact□Array□□□	**Closed**	**12/26/2012**
Related to Ruby - Bug #4136: Enumerable#reject should not inherit the receive...	**Closed**	**12/09/2010**

Associated revisions

Revision 86aa98fed4de1be2e868877fd786d3616d6c6ad5 - 02/01/2013 02:05 PM - Charlie Somerville

- array.c (rb_ary_dup): make returned array the same class as the original
  array [Bug #7768] [ruby-core:51792]
- test/ruby/test_array.rb (class TestArray): add test

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39004 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 86aa98fe - 02/01/2013 02:05 PM - Charlie Somerville

- array.c (rb_ary_dup): make returned array the same class as the original
  array [Bug #7768] [ruby-core:51792]
- test/ruby/test_array.rb (class TestArray): add test

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39004 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 450a515e959d2a35d6e1a59bad0a83db2a7698da - 02/08/2013 10:58 AM - U.Nakamura

- array.c (rb_ary_dup): reverted r39004. see [Bug #7768], and the
  release manager finailly decided to revert it.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39157 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 450a515e - 02/08/2013 10:58 AM - U.Nakamura

- array.c (rb_ary_dup): reverted r39004. see [Bug #7768], and the
  release manager finailly decided to revert it.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39157 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 1994adf938afcdc562f87497156e6d4900f3f06b - 09/02/2019 08:42 PM - jeremyevans (Jeremy Evans)

Make Array#uniq return subclass instance if called on subclass instance

Previously, Array#uniq would return subclass instance if the
length of the array were 2 or greater, and would return Array
instance if the length of the array were 0 or 1.

Fixes [Bug #7768]

Revision 1994adf938afcdc562f87497156e6d4900f3f06b - 09/02/2019 08:42 PM - jeremyevans (Jeremy Evans)

Make Array#uniq return subclass instance if called on subclass instance

Previously, Array#uniq would return subclass instance if the
length of the array were 2 or greater, and would return Array
instance if the length of the array were 0 or 1.

Fixes [Bug #7768]

Revision 1994adf9 - 09/02/2019 08:42 PM - jeremyevans (Jeremy Evans)

Make Array#uniq return subclass instance if called on subclass instance

Previously, Array#uniq would return subclass instance if the
length of the array were 2 or greater, and would return Array
instance if the length of the array were 0 or 1.

Fixes [Bug #7768]

History

#1 - 02/01/2013 09:56 PM - Anonymous

- Target version set to 2.0.0

- Assignee set to Anonymous

=begin
Looks like a regression introduced in r26987
=end

#2 - 02/01/2013 11:05 PM - Anonymous

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r39004.
Roman, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- array.c (rb_ary_dup): make returned array the same class as the original array [Bug #7768] [ruby-core:51792]
- test/ruby/test_array.rb (class TestArray): add test

#3 - 02/02/2013 01:35 AM - mrkn (Kenta Murata)

#7625 is related to this issue.

#4 - 02/07/2013 11:22 PM - Anonymous

=begin
Summary of my discussion with mame in #ruby-core:

```
[00:33:08]  r30148
[00:34:16]  some people complains to your r39004
[00:34:43]  because it might be an intentional change by matz at r30148
[00:34:53]  http://svn.ruby-lang.org/cgi-bin/viewvc.cgi?revision=30148&view=revision
[00:35:09]  * array.c (rb_ary_dup): should copy contents only.  no instance
[00:35:09]   variable, no class would be copied.
[00:36:04]  The point is that don't change behavior without matz's accept
```

```
[00:36:09]  however, matz is now saying the opposite opinion against himself
[00:36:38]  sorry for changing behaviour
[00:36:44]  he now prefers the old 1.9.3 behavior
[00:36:59]  the bug looked like a regression, but i fixed the wrong place
[00:37:25]  oops, 1.9.3 behavior -> 1.9.2 behavior
[00:38:53]  indeed rb_ary_dup impacts too extensively a little
[00:39:10]  then how do you fix?
[00:39:36]  so this was the bug http://bugs.ruby-lang.org/issues/7768
[00:40:25]  fix only Array#uniq ?
[00:40:35]  mame1: that would be the least intrusive way
[00:41:16]  mame1: do you want me to do it?
[00:42:22]  could you create a patch for the way?
[00:42:36]  sure, one moment
[00:43:25]  just confirm: you didn't intend to change the behavior of other methods than Array#uniq, right?
[00:43:53]  then, it looks good to me to choose "the least intrusive way"
[00:44:26]  mame1: i thought that perhaps there were other methods that would behave inconsistently in some cases like #uniq
[00:44:47]  but i will give you a patch that undoes it
[00:45:19]  honestly i agree with you to some extent
[00:45:33]  but the timing is bad a little
[00:45:39]  yes, i agree
[00:45:41]  sorry once again
[00:46:07]  something to discuss after 2.0 is released maybe?
[00:46:18]  no prob, thank you for your cooperation
[00:46:35]  no plan :-)
[00:47:12]  ah, it may be difficult to change 2.0.0 after 2.0.0-p0 is released
[00:47:38]  for next minor, i mean
[00:47:41]  2.0.1 or 2.1.0 will include the change, which matz will decide
[00:47:49]  yes
[00:48:29]  mame1: https://gist.github.com/charliesome/adeff49e900f6b8a75fc
[00:48:55]  the test i added in r39004 still passes
[00:49:11]  so i think this is ok if you're happy to change Array#uniq's behaviour
[00:51:31]  charliesome: it is really a regression, isn't it?
[00:52:04]  mame1: i think so - see http://bugs.ruby-lang.org/projects/ruby-trunk/repository/revisions/26987/diff/array.c
[00:52:19]  it looks like an attempt at an optimization
[00:52:51]  yeah, i've seen the diff, but i've not checked the other region of code
[00:53:15]  mame1: maybe the safest option is to revert r39004 completely?
[00:53:38]  it returns a new object whose class is the same of the argument, say, when its length is >= 2?
[00:53:45]  mame1: yes
[00:54:22]  hmm, thanks. let me consider for a minute :-)
[00:54:28]  no problem
[01:15:14]  charliesome: sorry for let you wait
[01:15:25]  i decided that i'll release rc2 with the current behavior
[01:15:45]  mame1: current as in trunk or current 1.9.3?
[01:16:00]  because matz again prefered not only Array#uniq but also other Array methods to return the original class rather than Array
[01:16:04]  current trunk
[01:16:35]  iow your r39004 is accepted :-)
=end
```

#5 - 02/08/2013 02:25 PM - shugo (Shugo Maeda)

- Status changed from Closed to Open

- Assignee changed from Anonymous to mame (Yusuke Endoh)

I believe r39004 should be reverted.

Matz said "If a method is originally defined in Enumerable, i.e. its return value (Array) is a collection of values from enumerable." at http://bugs.ruby-lang.org/issues/4136#note-7.
However, Array#sort returns an instance of a subclass of Array, by r39004.

```
$ ./ruby -ve 'class Foo < Array; end; p Foo[2,1,3].sort.class'
ruby 2.0.0dev (2013-02-08 trunk 39154) [i686-linux]
Foo
```

I'm not sure Matz is right. What should Array#uniq return if Enumerable#uniq is added in the future?

Anyway, there is no enough time to discuss details, so r39004 should be reverted.
Haste makes waste.
If this issue is regarded as a bug, not as a spec change, it can be fixed after the release of 2.0.0.

#6 - 02/08/2013 03:56 PM - usa (Usaku NAKAMURA)

- Status changed from Open to Assigned

Stated as the maintainer of 1.9.3, +1 to shugo.

#7 - 02/08/2013 07:53 PM - mame (Yusuke Endoh)

@shugo (Shugo Maeda)
Okay, I agree with reverting r39004. Sorry for my poor decision.

@charliesome
Sorry for confusing you. Please commit it again after I create ruby_2_0_0 branch.

--
Yusuke Endoh mame@tsg.ne.jp

#8 - 02/08/2013 07:58 PM - usa (Usaku NAKAMURA)

- Status changed from Assigned to Closed

This issue was solved with changeset r39157.
Roman, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- array.c (rb_ary_dup): reverted r39004. see [Bug #7768], and the release manager finailly decided to revert it.

#9 - 02/08/2013 07:59 PM - usa (Usaku NAKAMURA)

- Status changed from Closed to Assigned

- Target version changed from 2.0.0 to 2.6

#10 - 02/08/2013 11:51 PM - mame (Yusuke Endoh)

- Assignee changed from mame (Yusuke Endoh) to Anonymous

@usa (Usaku NAKAMURA)
Thank you!

@charliesome
I think that it is a good idea to fix only Array#uniq first.
Then, if you want to change other Array methods, please ask matz.

--
Yusuke Endoh mame@tsg.ne.jp

#11 - 10/12/2015 01:32 PM - blazeeboy (Emad Elsaid)

Charlie Somerville wrote:

> =begin
> Looks like a regression introduced in r26987
> =end

This Bug stil exists in my current ruby version

```
ruby 2.2.0p0 (2014-12-25 revision 49005) [x86_64-darwin14]
```

and had to make this workaround in my custom class

```
class Custom < Array
def uniq
return self if empty?
super
end
end
```

#12 - 07/15/2019 09:07 PM - jeremyevans0 (Jeremy Evans)

- File array-uniq-subclass-instance.patch added

mame (Yusuke Endoh) wrote:

> I think that it is a good idea to fix only Array#uniq first.

Looks like Array#uniq was never fixed. It still returns Array instance if length <= 1 and subclass instance otherwise. Attached is a patch that fixes this issue.

#13 - 07/30/2019 07:29 AM - ko1 (Koichi Sasada)

- Description updated

- Assignee changed from Anonymous to matz (Yukihiro Matsumoto)

#14 - 09/02/2019 06:17 AM - matz (Yukihiro Matsumoto)

Array#uniq should be fixed.

Matz.

#15 - 09/02/2019 10:19 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Assigned to Closed

Fixed by https://github.com/ruby/ruby/commit/1994adf938afcdc562f87497156e6d4900f3f06b.

Files

array-uniq-subclass-instance.patch	2.15 KB	07/15/2019	jeremyevans0 (Jeremy Evans)