

Ruby - Bug #3351

stack overflow on super

05/27/2010 03:45 PM - serge\_balyuk (Serge Balyuk)

<b>Status:</b>	Closed	<b>Backport:</b>
<b>Priority:</b>	Normal	
<b>Assignee:</b>	ko1 (Koichi Sasada)	
<b>Target version:</b>		
<b>ruby -v:</b>	ruby 1.9.3dev (2010-05-26 trunk 28028) [i686-linux]	

Description

It looks like super behavior is a bit different in 1.8 and 1.9. Please find the example below:

```
class Base
  def foo
    puts "bar"
  end
end

module Override
  def foo
    puts "override"
    super
  end
end

class A < Base
end

class B < A
end

B.send(:include, Override)
A.send(:include, Override)

B.new.foo

ruby 1.8.7 (2009-06-12 patchlevel 174) [i486-linux] output:

override
override
bar
```

and ruby 1.9.3dev (2010-05-26 trunk 28028) [i686-linux] output:

```
....
override
override
override
override
override
super.rb:9: stack level too deep (SystemStackError)
```

Hope that helps.

<b>Related issues:</b>		
Related to Ruby - Bug #2502: strange behavior of anonymous class inside a proc	Closed	12/19/2009
Related to Ruby - Bug #5236: Including a module in a superclass after it has ...	Closed	08/27/2011
Related to Ruby - Feature #1586: Including a module already present in ancest...	Rejected	

---

## Associated revisions

### Revision 2d3d84155d6c114a0f76888a8e2779e887722788 - 08/03/2012 10:27 AM - shugo (Shugo Maeda)

- `insns.def (invokesuper)`: don't skip the same class. instead, use `rb_method_entry_get_with_omod()` to avoid infinite loop when `super` is used with refinements. [ruby-core:30450] [Bug #3351]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36612 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision 2d3d8415 - 08/03/2012 10:27 AM - shugo (Shugo Maeda)

- `insns.def (invokesuper)`: don't skip the same class. instead, use `rb_method_entry_get_with_omod()` to avoid infinite loop when `super` is used with refinements. [ruby-core:30450] [Bug #3351]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36612 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision c8854d2ca4be9ee6946e6d17b0e17d9ef130ee81 - 07/06/2015 09:02 AM - ko1 (Koichi Sasada)

- `vm_insnhelper.c (vm_search_super_method)`: do not skip calling same methods in `super`. [Bug #3351]
- `test/ruby/test_super.rb`: fix a test.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@51161 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision c8854d2c - 07/06/2015 09:02 AM - ko1 (Koichi Sasada)

- `vm_insnhelper.c (vm_search_super_method)`: do not skip calling same methods in `super`. [Bug #3351]
- `test/ruby/test_super.rb`: fix a test.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@51161 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

---

## History

### #1 - 05/27/2010 04:08 PM - marcandre (Marc-Andre Lafortune)

- *Category set to core*

- *Priority changed from Normal to 3*

I thought I had submitted that bug, actually, but seems I forgot!

Just to clarify: this happens when a module appears twice in the list of ancestors. Ruby typically forbids this, but if the module is included in just the right order, it is possible. In the example given, invert the two includes and `Override` is included only once.

### #2 - 05/29/2010 04:21 AM - serge\_balyuk (Serge Balyuk)

Hi Marc-Andre,

Yes, exactly. Very rare situation. This is the simplified version of include sequence that I observed in one specific rails/rspec use case. And with inverted includes everything works like charm.

Thanks

### #3 - 06/26/2011 02:58 PM - naruse (Yui NARUSE)

- *Status changed from Open to Assigned*

- *Assignee set to ko1 (Koichi Sasada)*

- *Target version changed from 2.0.0 to 1.9.3*

### #4 - 06/28/2011 11:28 PM - funny\_falcon (Yura Sokolov)

I had caught by this with rails/sequel/custom backend for `delayed_jobs`.

After figuring, I ought to do some manipulations with requiring my initializers, and that looks ugly a bit.

**#5 - 07/10/2011 02:28 PM - ko1 (Koichi Sasada)**

- Target version changed from 1.9.3 to 2.0.0

I'll challenge this issue on 1.9.4. Sorry.

**#6 - 08/03/2012 07:27 PM - shugo (Shugo Maeda)**

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r36612.

Serge, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

- 
- `insns.def (invokesuper)`: don't skip the same class. instead, use `rb_method_entry_get_with_omod()` to avoid infinite loop when `super` is used with refinements. [[ruby-core:30450](#)] [Bug [#3351](#)]

**#7 - 08/04/2012 01:35 AM - shugo (Shugo Maeda)**

- Status changed from Closed to Open

In HEAD of trunk, stack overflow doesn't occur, but `Override#foo` is called only once.

So I reopen this ticket.

**#8 - 11/28/2012 10:35 PM - ko1 (Koichi Sasada)**

shugo-san, do you know why `Override#foo` called only once?

**#9 - 12/10/2012 12:28 AM - mame (Yusuke Endoh)**

- Status changed from Open to Assigned

- Assignee changed from ko1 (Koichi Sasada) to shugo (Shugo Maeda)

Shugo-san, ko1, what's the status?

Do you think this issue important?

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#10 - 12/13/2012 02:24 PM - shugo (Shugo Maeda)**

- Assignee changed from shugo (Shugo Maeda) to matz (Yukihiro Matsumoto)

mame (Yusuke Endoh) wrote:

Shugo-san, ko1, what's the status?

`Override#foo` is called only once, because in the SVN trunk, if a method found by `super` is the current method, it's skipped to avoid an infinite loop. The check was introduced for `super` in a refinement. Without it, `super` in a refinement causes an infinite loop.

Do you think this issue important?

I don't think so. Can I leave it as is, Matz?

**#11 - 12/13/2012 02:53 PM - Anonymous**

Hi,

In message "Re: [[ruby-core:50857](#)] [ruby-trunk - Bug [#3351](#)] stack overflow on super" on Thu, 13 Dec 2012 14:24:21 +0900, "shugo (Shugo Maeda)" [redmine@ruby-lang.org](mailto:redmine@ruby-lang.org) writes:

Do you think this issue important?

I don't think so. Can I leave it as is, Matz?

OK.

```
matz.
```

**#12 - 12/29/2012 10:03 PM - shugo (Shugo Maeda)**

- Status changed from Assigned to Open
- Assignee deleted (matz (Yukihiro Matsumoto))
- Target version changed from 2.0.0 to 2.6

**#13 - 02/18/2013 08:49 AM - ko1 (Koichi Sasada)**

- Assignee set to ko1 (Koichi Sasada)

**#14 - 07/01/2015 09:11 AM - ko1 (Koichi Sasada)**

- Description updated

**#15 - 07/01/2015 09:27 AM - ko1 (Koichi Sasada)**

Just now, we avoid this issue by putting klass into each frame.

However, the workaround is remaining.

Now, ancestors is here:

```
[B, Override, A, Override, Base, Object, Kernel, BasicObject]
```

and now MRI calls Override#foo only once. It skips second Override#foo.

```
override
bar
```

However, if there is a A#foo, then call Override#foo twice.

```
override
A
override
bar
```

It is ugly workaround. So I want to remove such skipping.

Matz: Can I call Override#foo twice even if it is duplicated with last call?

**#16 - 07/06/2015 08:38 AM - matz (Yukihiro Matsumoto)**

Agreed.

Matz.

**#17 - 07/06/2015 09:03 AM - ko1 (Koichi Sasada)**

- Status changed from Open to Closed

Applied in changeset r51161.

- 
- vm\_insnhelper.c (vm\_search\_super\_method): do not skip calling same methods in super.  
[Bug [#3351](#)]
  - test/ruby/test\_super.rb: fix a test.

**Files**

---

super.rb	217 Bytes	05/27/2010	serge_balyuk (Serge Balyuk)
----------	-----------	------------	-----------------------------