Ruby - Bug #5761

Array#flatten(N) calls to_ary on the (N+1)-level objects

12/14/2011 06:48 PM - jonleighton (Jon Leighton)

Status:	Closed		
Priority:	Normal		
Assignee:	mame (Yusuke Endoh)		
Target version:			
ruby -v:	ruby 1.9.3p0 (2011-10-30 revision 33570) [x86_64-linux]	**Backport:**	

Description

$ cat flatten.rb
class Foo
def respond_to?(name, include_private = true)
puts "respond_to?(#{name})"
super
end

def method_missing(name, *args)
puts "method_missing(#{name})"
super
end
end

puts "[[Foo.new]].flatten(1)"
p [[Foo.new]].flatten(1)

puts
puts "[[[Foo.new]]].flatten(2)"
p [[[Foo.new]]].flatten(2)

puts
puts "[[[Foo.new]]].flatten(1)"
p [[[Foo.new]]].flatten(1)

$ ruby flatten.rb
[[Foo.new]].flatten(1)
method_missing(to_ary)
respond_to?(to_ary)
[#Foo:0x00000001ae41c8]

[[[Foo.new]]].flatten(2)
method_missing(to_ary)
respond_to?(to_ary)
[#Foo:0x00000001ae3d90]

[[[Foo.new]]].flatten(1)
#Foo:0x00000001ae3980

$ ruby -v
ruby 1.9.3p0 (2011-10-30 revision 33570) [x86_64-linux]

Expected behaviour: no method calls are made on the (N+1)-level objects.

For what it's worth, this is the 1.8.7 output:

$ ruby flatten.rb
[[Foo.new]].flatten(1)
respond_to?(to_ary)
[#Foo:0x7fef3ebb1888]

[[[Foo.new]]].flatten(2)

respond_to?(to_ary)
[#Foo:0x7fef3ebb16a8]

[[[Foo.new]]].flatten(1)
#Foo:0x7fef3ebb14f0

Related issues:

Copied to Ruby - Bug #10748: Array#flatten(N) calls to_ary on the (N+1)-level...	**Closed**	**12/14/2011**

History

#1 - 03/11/2012 05:03 PM - ko1 (Koichi Sasada)

- Category set to core

- Assignee set to mame (Yusuke Endoh)

#2 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#3 - 04/21/2012 01:46 AM - mame (Yusuke Endoh)

- Status changed from Assigned to Closed

This is already fixed in ruby-1.9.3-p194.

--
Yusuke Endoh mame@tsg.ne.jp

#4 - 01/16/2015 07:46 PM - ryannevell (Ryan Nevell)

- Copied to Bug #10748: Array#flatten(N) calls to_ary on the (N+1)-level objects added