

Ruby - Bug #7564

r38175 introduces incompatibility

12/15/2012 03:02 AM - tenderlovmaking (Aaron Patterson)

Status:	Closed	Backport:
Priority:	Normal	
Assignee:	nobu (Nobuyoshi Nakada)	
Target version:	2.0.0	
ruby -v:	ruby 2.0.0dev (2012-12-15 trunk 38381) [x86_64-darwin12.2.1]	
Description		
r38175 introduces incompatibility with 1.9.3. Before r38175, when looking for <code>_dump</code>, Marshal would not call <code>method_missing</code>. Now marshal calls <code>method_missing</code> when trying to dump. The following example exits with no error on 1.9.3, but on trunk it raises an exception (<code>_dump()</code> must return string (TypeError)): <pre>class TR def initialize calls = [] @calls = calls end def method_missing name, *args @calls << [name, args] end end Marshal.dump TR.new</pre> I've attached a test case.		
Related issues:		
Related to Ruby - Bug #7638: trunk rails activesupport		Closed 12/30/2012

Associated revisions

Revision 433999fd5c82a8925baf312155351252320a44ce - 01/20/2013 02:55 PM - nobu (Nobuyoshi Nakada)

marshal.c: get back to the old behavior

- marshal.c (w_object, r_object0): separate respond_to checks and calling, and get back to the old behavior for 2.0. [Bug #7564]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38888 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 433999fd - 01/20/2013 02:55 PM - nobu (Nobuyoshi Nakada)

marshal.c: get back to the old behavior

- marshal.c (w_object, r_object0): separate respond_to checks and calling, and get back to the old behavior for 2.0. [Bug #7564]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38888 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 12/18/2012 02:33 AM - ko1 (Koichi Sasada)

- Category set to core
- Assignee set to mame (Yusuke Endoh)
- Target version set to 2.0.0

mame-san, how about this ticket?

#2 - 12/19/2012 02:46 AM - tenderlovmaking (Aaron Patterson)

Bump. /cc nobu

#3 - 12/19/2012 10:04 AM - zenspider (Ryan Davis)

I'm getting bit by this in my multi-version CI on Flay and any other project that uses the sexp gem and calls my deep_clone:

```
def deep_clone
  Marshal.load(Marshal.dump(self))
end
```

#4 - 12/21/2012 10:19 PM - usa (Usaku NAKAMURA)

- Status changed from Open to Assigned

#5 - 12/23/2012 10:01 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Rejected

If method_missing does not deal with a method call, it should raise NoMethodError.

```
def method_missing name, *args
  @calls << [name, args]
  super
end
```

#6 - 12/24/2012 04:59 AM - Anonymous

On Sun, Dec 23, 2012 at 10:01:33AM +0900, nobu (Nobuyoshi Nakada) wrote:

Issue [#7564](#) has been updated by nobu (Nobuyoshi Nakada).

Status changed from Assigned to Rejected

If method_missing does not deal with a method call, it should raise NoMethodError.

```
def method_missing name, *args
  @calls << [name, args]
  super
end
```

Before this changeset, method_missing *did* deal with all method calls. That's why this is not backwards compatible.

--

Aaron Patterson

<http://tenderlovmaking.com/>

#7 - 12/29/2012 01:35 PM - zenspider (Ryan Davis)

I still think this is a bug, as shown by needing a respond_to? that does nothing more than call super:

```
class Sexp < Array
  def inspect
    "s(#{map(&:inspect).join ', '})"
  end

  def respond_to? meth
    super
  end if ENV["WHY_DO_I_NEED_THIS"]

  def method_missing meth, delete = false
    raise "shouldn't be here: #{meth.inspect}"
  end
end

def s *args
  Sexp.new args
end

p Marshal.load Marshal.dump s(1, 2, 3)
puts

# % WHY_DO_I_NEED_THIS=1 ruby20 trunk_bug.rb && ruby20 trunk_bug.rb
# s(1, 2, 3)
```

```
#
# trunk_bug.rb:11:in `method_missing': shouldn't be here: :marshal_dump (RuntimeError)
#   from trunk_bug.rb:19:in `dump'
#   from trunk_bug.rb:19:in `<main>'
```

#8 - 01/03/2013 07:25 AM - zenspider (Ryan Davis)

- Status changed from Rejected to Open

No, really. This is a bug that needs more eyeballs.

A respond_to? with just a super should be equivalent to no code at all.

Can we get Matz and Mame to weigh in?

#9 - 01/04/2013 01:02 PM - nobu (Nobuyoshi Nakada)

Anonymous wrote:

Before this changeset, method_missing *did* deal with all method calls.

No, previously respond_to? was called so method_missing did *not* get called.

That's why this is not backwards compatible.

It depended on the internal behavior too much.

And the bug that overriding method_missing without respond_to_missing? has been revealed.

Just like overriding hash without eql?.

#10 - 01/11/2013 09:37 AM - zenspider (Ryan Davis)

This seems highly inconsistent to me. Specifically, MM + RT is the only working solution, but RT implementation is entirely meaningless. It doesn't make sense to me that I need a method table entry that does nothing but super. That should be the same as not existing in the method table.

Notes:

Runtime	Passes	Failures
(none)		1.8 1.9 2.0 fails #blah (expected)
MM	1.8 1.9	2.0 raises w/ marshal_dump
MM + RT	1.8 1.9 2.0	
MM + RTM	1.8 1.9	2.0 raises w/ marshal_dump (doesn't use RTM)
MM + RT + RTM	1.8 1.9 2.0	
MM + ARTM	1.8	1.9 SSE, 2.0 raises
MM + RT + ARTM	1.8	1.9 SSE, 2.0 SSE (SystemStackError)
RT		1.8 1.9 2.0 fails #blah (expected)
RTM		1.8 1.9 2.0 fails #blah (expected)
ARTM		1.8 2.0 fails #blah (expected), 1.9 SSE

```
class Sexp < Array
  def inspect
    "s(#{map(&:inspect).join ', '})"
  end

  def method_missing meth, delete = false
    x = nil
    return x if x = find { |o| Sexp === o && o.first == meth }
    raise "shouldn't be here: #{inspect}.#{meth}"
  end if ENV["MM"]

  def respond_to? meth, private = false
    p :respond_to? => meth if ENV["V"]
    super
  end if ENV["RT"]
end
```

```

def respond_to_missing? meth, private = false
  p :respond_to_missing? => meth if ENV["V"]
  super
end if ENV["RTM"]

alias respond_to_missing? respond_to? if ENV["ARTM"]
end

def s *args
  Sexp.new args
end

END { puts }
p Marshal.load Marshal.dump s(1, 2, 3)
p s(1, 2, s(:blah)).blah

```

#11 - 01/15/2013 09:55 AM - ko1 (Koichi Sasada)

- Priority changed from Normal to 7

#12 - 01/19/2013 03:48 AM - matz (Yukihiro Matsumoto)

Since this is incompatibility, I propose to get back to the old behavior, for 2.0.
We have to discuss the issue that [@zenspider \(Ryan Davis\)](#) mentioned in separated thread.

Matz.

#13 - 01/20/2013 04:55 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned
- Assignee changed from mame (Yusuke Endoh) to nobu (Nobuyoshi Nakada)

I agree with Matz. Nobu, please handle this.

#14 - 01/20/2013 11:55 PM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r38888.
Aaron, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

marshal.c: get back to the old behavior

- marshal.c (w_object, r_object0): separate respond_to checks and calling, and get back to the old behavior for 2.0. [Bug [#7564](#)]

Files

bug.patch	675 Bytes	12/15/2012	tenderlovemaking (Aaron Patterson)
-----------	-----------	------------	------------------------------------