r38175 introduces incompatibility

r38175 introduces incompatibility with 1.9.3. Before r38175, when looking for _dump, Marshal would not call method_missing. Now marshal calls method_missing when trying to dump.

The following example exits with no error on 1.9.3, but on trunk it raises an exception (_dump() must return string (TypeError)):

```ruby
class TR
  def initialize calls = []
    @calls = calls
  end

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

Marshal.dump TR.new
```
I've attached a test case.

Related issues:
Related to Ruby master - Bug #7638: trunk で rails の activesupport のテストが失敗してしまう

Associated revisions
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]

Revision 38888 - 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]

Revision 38888 - 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]

Revision 38888 - 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]

Revision 38888 - 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]

Revision 38888 - 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]

Revision 38888 - 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]
Revision 38888 - 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]

Revision 38888 - 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]

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 - tenderlovemaking (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:

```ruby
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.

```ruby
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.

```ruby
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://tenderlovemaking.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:

```ruby
class Sexp < Array
  def inspect
    "#{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 *args
    Sexp.new args
  end
end
```

```ruby
p Marshal.load Marshal.dump S(1, 2, 3)
puts
```

#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:

<table>
<thead>
<tr>
<th>Runtime</th>
<th>Passes</th>
<th>Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>(none)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MM</td>
<td>1.8 1.9</td>
<td>2.0 raises w/ marshal_dump</td>
</tr>
<tr>
<td>MM + RT</td>
<td>1.8 1.9 2.0</td>
<td></td>
</tr>
<tr>
<td>MM + RTM</td>
<td>1.8 1.9</td>
<td>2.0 raises w/ marshal_dump (doesn't use RTM)</td>
</tr>
<tr>
<td>MM + RT + RTM</td>
<td>1.8 1.9 2.0</td>
<td></td>
</tr>
<tr>
<td>MM + ARTM</td>
<td>1.8</td>
<td>1.9 SSE, 2.0 raises</td>
</tr>
</tbody>
</table>

03/13/2022
<table>
<thead>
<tr>
<th>Runtime</th>
<th>Passes</th>
<th>Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM + RT + ARTM</td>
<td>1.8</td>
<td>1.9 SSE, 2.0 SSE (SystemStackError)</td>
</tr>
<tr>
<td>RT</td>
<td>1.8 1.9 2.0 fails #blah (expected)</td>
<td></td>
</tr>
<tr>
<td>RTM</td>
<td>1.8 1.9 2.0 fails #blah (expected)</td>
<td></td>
</tr>
<tr>
<td>ARTM</td>
<td>1.8 2.0 fails #blah (expected), 1.9 SSE</td>
<td></td>
</tr>
</tbody>
</table>

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"]

  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

03/13/2022
- 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

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>bug.patch</td>
<td>675 Bytes</td>
<td>12/15/2012</td>
<td>tenderlovemaking (Aaron Patterson)</td>
</tr>
</tbody>
</table>