Ruby trunk - Bug #3562

regression in respond_to?

07/13/2010 03:45 AM - tenderlovemaking (Aaron Patterson)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:	2.0.0		
ruby -v:	ruby 1.9.3dev (2010-07-12 trunk 28620) [x86_64-darwin10.3.1]	**Backport:**	

Description

=begin
respond_to? returns false for protected methods. This breaks classes that want to inflect on themselves. For example:

```
class Foo
  protected
  def do_bar
    "bar"
  end

  def method_missing name
    super unless respond_to?(:"do_#{name}")
    send(:"do_#{name}")
  end
end

class Subclass < Foo
  protected
  def do_baz
    "baz"
  end
end

p Foo.new.bar
p Subclass.new.baz
```

This code works in 1.8, 1.9.2, but not trunk. This change to respond_to?() breaks existing ruby code[1]. Is that intended? Could we compromise by including private and protected methods when an object inflects upon itself?

1. http://github.com/svenfuchs/i18n/blob/master/lib/i18n/backend/base.rb#L203 =end

History

#1 - 07/13/2010 04:07 AM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:31217] [Bug #3562] regression in respond_to?"
on Tue, 13 Jul 2010 03:45:55 +0900, Aaron Patterson redmine@ruby-lang.org writes:

|This code works in 1.8, 1.9.2, but not trunk. This change to respond_to?() breaks existing ruby code[1]. Is that intended? Could we compromise by including private and protected methods when an object inflects upon itself?

Hmm, that check is costly for respond_to?() Could you replace

respond_to?(:"do_#{name}")

by

respond_to?(:"do_#{name}",true)

?

```
                     matz.
```

=end

#2 - 07/13/2010 06:30 AM - tenderlovemaking (Aaron Patterson)

=begin
On Tue, Jul 13, 2010 at 04:06:26AM +0900, Yukihiro Matsumoto wrote:

> Hi,
>
> In message "Re: [ruby-core:31217] [Bug #3562] regression in respond_to?"
> on Tue, 13 Jul 2010 03:45:55 +0900, Aaron Patterson redmine@ruby-lang.org writes:
>
> |This code works in 1.8, 1.9.2, but not trunk. This change to respond_to?() breaks existing ruby code[1]. Is that intended? Could we compromise by including private and protected methods when an object inflects upon itself?
>
> Hmm, that check is costly for respond_to?() Could you replace
>
> respond_to?(:"do_#{name}")
>
> by
>
> respond_to?(:"do_#{name}",true)
>
> ?

That seems fine, but someone writing code like my example against 1.9.2
will find it broken in 1.9.3.

Could we backport this to 1.9.2 before release? Or find some other
solution?

--
Aaron Patterson
http://tenderlovemaking.com/

Attachment: (unnamed)
=end

#3 - 07/13/2010 07:22 AM - wycats (Yehuda Katz)

=begin
In my opinion, small breaking changes to Ruby 1.9.x (like this) are not worth the improvement in correctness. In order to encourage adoption of Ruby 1.9, we should make it easy for library authors to write a single gem that runs against 1.8 and 1.9. For the most part, this is quite simple. The similar change to const_defined? caused a headache.

One way to achieve this desired functionality without breaking backwards compatibility would be to change the flag's default. In other words, retain the identical semantics of the new parameter, but default it to true (for compatibility with Ruby 1.8) instead of false.

EDIT: Ah, the param existed, but its semantics are changing to include protected. My position is the same. The adoption problems caused by small breaking changes like this outweigh the improvement in "correctness".
=end

#4 - 07/13/2010 05:33 PM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:31229] [Bug #3562] regression in respond_to?"
on Tue, 13 Jul 2010 07:22:30 +0900, Yehuda Katz redmine@ruby-lang.org writes:
|
|In my opinion, small breaking changes to Ruby 1.9.x (like this) are not worth the improvement in correctness. In order to encourage adoption of Ruby 1.9, we should make it easy for library authors to write a single gem that runs against 1.8 and 1.9. For the most part, this is quite simple. The similar change to const_defined? caused a headache.

I consider this change will not be merged til Ruby 2.0. The changes
to the trunk will not be merged to 1.9.3 unless they are bug fixes or
somebody explicitly request.

```
                                                        matz.
```

=end

#5 - 07/13/2010 11:10 PM - mame (Yusuke Endoh)

- *Target version set to 2.0.0*

=begin
Hi,

2010/7/13 Yukihiro Matsumoto matz@ruby-lang.org:

> I consider this change will not be merged til Ruby 2.0. ?The changes
> to the trunk will not be merged to 1.9.3 unless they are bug fixes or
> somebody explicitly request.

As far as I know, many committers think the current trunk leads to
1.9.3. In fact, the version of current trunk is "ruby 1.9.3dev".
So, the changes to the trunk will be included in 1.9.3 by default,
I think. Is now the time to create ruby_1_9 branch?

This change was done by Akinori Musha with discussion of the thread
from [ruby-dev:40461]. Akinori, I think we should revert the commit
once.

Anyway, this change will never be merged to 1.9.2. Thus I set the
target to 1.9.x.

--
Yusuke Endoh mame@tsg.ne.jp
=end

#6 - 07/14/2010 01:10 AM - tenderlovemaking (Aaron Patterson)

=begin
On Tue, Jul 13, 2010 at 05:33:41PM +0900, Yukihiro Matsumoto wrote:

> Hi,
>
> In message "Re: [ruby-core:31229] [Bug #3562] regression in respond_to?"
> on Tue, 13 Jul 2010 07:22:30 +0900, Yehuda Katz redmine@ruby-lang.org writes:
> |
> |In my opinion, small breaking changes to Ruby 1.9.x (like this) are not worth the improvement in correctness. In order to encourage adoption of Ruby 1.9, we should make it easy for library authors to write a single gem that runs against 1.8 and 1.9. For the most part, this is quite simple. The similar change to const_defined? caused a headache.
>
> I consider this change will not be merged til Ruby 2.0. The changes
> to the trunk will not be merged to 1.9.3 unless they are bug fixes or
> somebody explicitly request.

Ah, that's good. I think this is a good thing to put in Ruby 2.0.

Where should I be committing changes that I want to go in 1.9.x? I've
made many changes on trunk that I want to be in 1.9.

Thanks!

--
Aaron Patterson
http://tenderlovemaking.com/

Attachment: (unnamed)
=end

#7 - 07/14/2010 01:24 AM - tenderlovemaking (Aaron Patterson)

=begin
On Tue, Jul 13, 2010 at 11:10:09PM +0900, Yusuke Endoh wrote:

> Issue #3562 has been updated by Yusuke Endoh.
>
> Target version set to 1.9.x
>
> Hi,
>
> 2010/7/13 Yukihiro Matsumoto matz@ruby-lang.org:
>
>> I consider this change will not be merged til Ruby 2.0. ?The changes

to the trunk will not be merged to 1.9.3 unless they are bug fixes or
somebody explicitly request.

As far as I know, many committers think the current trunk leads to
1.9.3. In fact, the version of current trunk is "ruby 1.9.3dev".
So, the changes to the trunk will be included in 1.9.3 by default,
I think. Is now the time to create ruby_1_9 branch?

Yes, I thought trunk would be 1.9.3. If trunk is actually Ruby 2.0, it
would be good if we made a branch.

This change was done by Akinori Musha with discussion of the thread
from [ruby-dev:40461]. Akinori, I think we should revert the commit
once.

If trunk is 1.9.x, I agree.

Anyway, this change will never be merged to 1.9.2. Thus I set the
target to 1.9.x.

Sounds good!

--
Aaron Patterson
http://tenderlovemaking.com/

Attachment: (unnamed)
=end

#8 - 07/14/2010 01:31 AM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:31253] [Bug #3562] regression in respond_to?"
on Tue, 13 Jul 2010 23:10:09 +0900, Yusuke Endoh redmine@ruby-lang.org writes:

|As far as I know, many committers think the current trunk leads to
|1.9.3. In fact, the version of current trunk is "ruby 1.9.3dev".
|So, the changes to the trunk will be included in 1.9.3 by default,
|I think. Is now the time to create ruby_1_9 branch?

I guess so.

```
                            matz.
```

=end

#9 - 07/14/2010 12:36 PM - naruse (Yui NARUSE)

=begin
2010/7/14 Yukihiro Matsumoto matz@ruby-lang.org:

In message "Re: [ruby-core:31253] [Bug #3562] regression in respond_to?"
on Tue, 13 Jul 2010 23:10:09 +0900, Yusuke Endoh redmine@ruby-lang.org writes:

|As far as I know, many committers think the current trunk leads to
|1.9.3. In fact, the version of current trunk is "ruby 1.9.3dev".
|So, the changes to the trunk will be included in 1.9.3 by default,
|I think. Is now the time to create ruby_1_9 branch?

I guess so.

Whether create ruby_1_9 or not, current trunk is 1.9.
So I want revert this change to ease maintaining RubySpec.
Please recommit it after branching.

I think, in current situation feature changes should be in matzruby
branch (or github).
We can't maintain ruby_1_9 with current our resource.

--
NARUSE, Yui
naruse@airemix.jp

=end

#10 - 07/15/2010 03:45 PM - naruse (Yui NARUSE)

=begin
Hi,

2010/7/15 Marc-Andre Lafortune ruby-core-mailing-list@marc-andre.ca:

> Hi,
>
> On Tue, Jul 13, 2010 at 11:35 PM, NARUSE, Yui naruse@airemix.jp wrote:
>
>> So I want revert this change to ease maintaining RubySpec.
>
> RubySpec is already fixed, with the exception of lib/delegate; I'm
> waiting for decisions to be made to ask wether delegates should
> forward protected methods or not (and thus change the specs & lib
> accordingly).

I'm talking about that two failures.

> I am not saying to revert it or not, just that maintaining RubySpec is
> not a problem.
>
> It's quite unfortunate that no decision was made in time for 1.9.2.

It is needless to say that it won't merged in 1.9.2 because of feature freeze.

> Did we consider adding a warning in 1.9.2 if respond_to(:foo, false)
> is called for a protected method :foo?
>
> Another solution would be to revert the change and introduce
> respond_to_public?. If the flag of respond_to? defaulted back to
> true, then neither send(:foo) if respond_to?(:foo) nor
> send_public(:bar) if respond_to_public?(:bar) would raise a
> NoMethodError.
>
> I'll adapt RubySpec when a decision is made; I also understood that
> the change was for 1.9.3 and coded accordingly.

Matz says the change won't be in 1.9.3 but 2.0 in [ruby-core:31244].

--
NARUSE, Yui
naruse@airemix.jp

=end

#11 - 07/20/2010 09:32 AM - naruse (Yui NARUSE)

=begin
I'll commit following revert patch.

diff --git a/test/ruby/test_method.rb b/test/ruby/test_method.rb
index d135577..da17ef5 100644
--- a/test/ruby/test_method.rb
+++ b/test/ruby/test_method.rb
@@ -371,7 +371,7 @@ class TestMethod < Test::Unit::TestCase

```
 assert_equal(true,  respond_to?(:mv1))
 assert_equal(false, respond_to?(:mv2))
```

- assert_equal(false, respond_to?(:mv3))

- assert_equal(true, respond_to?(:mv3))

assert_equal(true, respond_to?(:mv1, true))
assert_equal(true, respond_to?(:mv2, true))
@@ -393,7 +393,7 @@ class TestMethod < Test::Unit::TestCase

assert_equal(true, v.respond_to?(:mv1))
assert_equal(false, v.respond_to?(:mv2))

- assert_equal(false, v.respond_to?(:mv3))

- assert_equal(true, v.respond_to?(:mv3))

assert_equal(true, v.respond_to?(:mv1, true))
assert_equal(true, v.respond_to?(:mv2, true))
diff --git a/vm_method.c b/vm_method.c
index aa5db73..50f0b12 100644
--- a/vm_method.c
+++ b/vm_method.c
@@ -565,19 +565,18 @@ rb_method_boundp(VALUE klass, ID id, int ex)
{
rb_method_entry_t *me = rb_method_entry(klass, id);

- if (!me) return 0;

- if (ex & ~NOEX_RESPONDS) { /* pub */

- [redacted]

- [redacted]

- [redacted]

- if (me != 0) {

- [redacted]

- [redacted]
  }

- [redacted]

- [redacted]

- [redacted]

- [redacted]

- [redacted]

- [redacted]

}

- if (!me->def) return 0;

- if (me->def->type == VM_METHOD_TYPE_NOTIMPLEMENTED) {

- [redacted]

- [redacted]

- }

- return 1;

- return 0;
  }

void
=end

#12 - 07/21/2010 02:07 PM - naruse (Yui NARUSE)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin
This issue was solved with changeset r28700.
Aaron, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end