

Ruby master - Bug #2206

lib/delegate: inconsistency between respond_to? and send

10/14/2009 01:06 PM - marcandre (Marc-Andre Lafortune)

Status: Closed	
Priority: Normal	
Assignee:	
Target version: 1.9.2	
ruby -v: ruby 1.9.2dev (2009-10-13 trunk 25317) [x86_64-darwin10.0.0]	Backport:

Description

```
=begin
require 'delegate'
class A
private
def private_method
:bar
end
end
```

```
x = SimpleDelegator.new(A.new)
x.respond_to?(:private_method, true) # ==> true
x.send(:private_method) # ==> NoMethodError
```

I expected the call to private_method to be delegated. The attached patch fixes the problem. It is not very elegant but I couldn't think of anything better.

```
=end
```

History

#1 - 10/16/2009 08:07 PM - matz (Yukihiko Matsumoto)

- Status changed from Open to Closed

- % Done changed from 0 to 100

```
=begin
This issue was solved with changeset r25372.
Marc-Andre, thank you for your reporting of the issue.
You have greatfully contributed toward Ruby.
May Ruby be with you.
```

```
=end
```

#2 - 10/18/2009 10:42 AM - matz (Yukihiko Matsumoto)

```
=begin
Hi,
```

In message "Re: [ruby-core:26080] [Bug #2206] lib/delegate: inconsistency between respond_to? and send" on Wed, 14 Oct 2009 13:06:09 +0900, Marc-Andre Lafortune redmine@ruby-lang.org writes:

```
|require 'delegate'
|class A
| private
| def private_method
| :bar
| end
|end
|
|x = SimpleDelegator.new(A.new)
|x.respond_to?(:private_method, true) # ==> true
|x.send(:private_method) # ==> NoMethodError
```

I expected the call to private_method to be delegated. The attached patch fixes the problem. It is not very elegant but I couldn't think of anything better.

No, delegators should only delegate public methods. I will make #respond_to? to warn if you specify true for its optional argument.

```
matz.
```

```
=end
```

Files

delegate_send.diff	1007 Bytes	10/14/2009	marcandre (Marc-Andre Lafortune)
--------------------	------------	------------	----------------------------------