

## Ruby master - Bug #15501

### private\_methods(false) changes behavior based on singleton class allocation

01/03/2019 02:15 PM - kddeisz (Kevin Deisz)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> nobu (Nobuyoshi Nakada)	
<b>Target version:</b>	
<b>ruby -v:</b>	<b>Backport:</b> 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN

#### Description

```
class Module
  prepend Module.new
end
```

```
module Foo
end
```

```
p Foo.private_methods(false)
# => [:remove_const, :method_added, :method_removed, :method_undefined, :prepend_features, :extend_object, :refine, :using, :append_features, :initialize, :initialize_clone, :initialize_copy, :public, :private, :module_function, :protected, :included, :extended, :prepend]
```

```
Foo.singleton_class
```

```
p Foo.private_methods(false)
# => []
```

I have no idea what is causing this, but I tested it and it looks like this goes back to at least 2.1.5. It looks like before the singleton class is allocated `private_methods(false)` returns a whole host of methods, while after it returns the expected output. Any help/advice would be appreciated.

#### Associated revisions

##### Revision 140f8b94 - 01/09/2019 02:04 PM - nobu (Nobuyoshi Nakada)

class.c: refactor class\_instance\_method\_list

- class.c (class\_instance\_method\_list): gather singleton and extended methods first separately from ancestors. [ruby-core:90872] [Bug #15501]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@66764 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 66764 - 01/09/2019 02:04 PM - nobu (Nobuyoshi Nakada)

class.c: refactor class\_instance\_method\_list

- class.c (class\_instance\_method\_list): gather singleton and extended methods first separately from ancestors. [ruby-core:90872] [Bug #15501]

#### History

##### #1 - 01/03/2019 05:12 PM - marcandre (Marc-Andre Lafortune)

- Assignee set to nobu (Nobuyoshi Nakada)

Good catch.

##### #2 - 01/09/2019 02:04 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset [trunk|r66764](#).

class.c: refactor class\_instance\_method\_list

- class.c (class\_instance\_method\_list): gather singleton and extended methods first separately from ancestors. [ruby-core:90872] [Bug #15501]