

## Ruby trunk - Feature #6539

### public and private for core methods

06/03/2012 07:55 AM - marcandre (Marc-Andre Lafortune)

|   |                         |
|---|-------------------------|
| <b>Status:</b>  | Closed                  |
| <b>Priority:</b>  | Normal                  |
| <b>Assignee:</b>  | nobu (Nobuyoshi Nakada) |
| <b>Target version:</b>  |                         |
| <b>Description</b>  |                         |
| I feel there are inconsistencies in which methods are public and which are private.   |                         |
| For example:  |                         |
| <pre>obj = [] prev = obj.instance_variable_get(:@foo) # allowed obj.instance_variable_set(:@foo, nil) # allowed # But these are almost equivalent to: prev = obj.remove_instance_variable(:@foo) # =&gt; NoMethodError: private method `remove_instance_variable' called for []:Array</pre> |                         |
| Another example:  |                         |
| <pre>module M   def foo     42   end end  M.module_function :foo # private method `module_function' called for M:Module M.extend M # allowed M.foo # =&gt; 42</pre>   |                         |
| Reverse example:  |                         |
| <pre>{}.method_missing :foo # =&gt; private method `method_missing' called {}.respond_to_missing? :foo, false # =&gt; allowed, why?</pre>   |                         |
| Which methods should be private is a different question for Ruby than for apps and libraries; the "real" private methods of Ruby are in C!  |                         |
| For Ruby, I feel that a method should be private if it is <i>not meant to be called</i> except by Ruby itself (callbacks, etc...), or if it's a "global" methods of Kernel that is meant to be called directly (i.e. puts instead of 42.puts)   |                         |
| Otherwise, it should be public. This includes methods like Module#include.  |                         |
| I don't know what the rationale was to make include and the like private.   |                         |
| I feel it is now quite common to use metaprogramming, e.g. to include modules from outside a class. It's part of a Class' API that it can be extended and modified, so these methods should be public.  |                         |
| Concrete proposal:  |                         |
| Should be made private:   |                         |
| <pre>Object and descendants   #initialize_clone   #initialize_dup   #respond_to_missing? Rational &amp; Complex   #marshal_dump   #marshal_load Time   #_dump   ._load</pre>  |                         |

Note that Delegate#initialize\_{clone|dup} are already private

Should be made public:

```
Object
  #remove_instance_variable
Module
  #attr
  #attr_reader
  #attr_writer
  #attr_accessor
  #remove_const
  #include
  #remove_method
  #undef_method
  #alias_method
  #public
  #protected
  #private
  #module_function
  #define_method
```

#### Related issues:

|   |               |                   |
|---|---------------|-------------------|
| Related to Ruby trunk - Feature #8846: Publicize Module#include                   | <b>Closed</b> | <b>08/31/2013</b> |
| Related to Ruby trunk - Feature #12697: Why shouldn't Module meta programming...  | <b>Closed</b> |                   |
| Related to Ruby trunk - Feature #14132: Module#attr{[_reader _writer]} should ... | <b>Closed</b> |                   |
| Related to Ruby trunk - Feature #14133: Module#{define alias undef}_method sh...  | <b>Closed</b> |                   |

#### Associated revisions

##### Revision cf9d17a3 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)

vm\_method.c: make initialize methods private

- id.c (Init\_id), template/id.h.tmpl: add initialize\_{copy,clone,dup} and respond\_to\_missing?.
- vm\_method.c (rb\_method\_entry\_make): make above methods private. [Feature #6539]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38113 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 38113 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)

vm\_method.c: make initialize methods private

- id.c (Init\_id), template/id.h.tmpl: add initialize\_{copy,clone,dup} and respond\_to\_missing?.
- vm\_method.c (rb\_method\_entry\_make): make above methods private. [Feature #6539]

##### Revision 38113 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)

vm\_method.c: make initialize methods private

- id.c (Init\_id), template/id.h.tmpl: add initialize\_{copy,clone,dup} and respond\_to\_missing?.
- vm\_method.c (rb\_method\_entry\_make): make above methods private. [Feature #6539]

##### Revision 38113 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)

vm\_method.c: make initialize methods private

- id.c (Init\_id), template/id.h.tmpl: add initialize\_{copy,clone,dup} and respond\_to\_missing?.
- vm\_method.c (rb\_method\_entry\_make): make above methods private. [Feature #6539]

##### Revision 38113 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)

vm\_method.c: make initialize methods private

- id.c (Init\_id), template/id.h.tmpl: add initialize\_{copy,clone,dup} and respond\_to\_missing?.
- vm\_method.c (rb\_method\_entry\_make): make above methods private. [Feature #6539]

##### Revision 38113 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)

vm\_method.c: make initialize methods private

- id.c (Init\_id), template/id.h.tmpl: add initialize\_{copy,clone,dup} and respond\_to\_missing?.

- `vm_method.c (rb_method_entry_make)`: make above methods private. [Feature #6539]

**Revision 38113 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)**

`vm_method.c`: make initialize methods private

- `id.c (Init_id)`, `template/id.h.templ`: add `initialize_{copy,clone,dup}` and `respond_to_missing?`.
- `vm_method.c (rb_method_entry_make)`: make above methods private. [Feature #6539]

**Revision 82ec3de0 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)**

`object.c`: make `remove_instance_variable` public

- `object.c (Init_Object)`: make `remove_instance_variable` public. [Feature #6539]

git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/trunk@38114](https://ci.ruby-lang.org/ruby/trunk@38114) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 38114 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)**

`object.c`: make `remove_instance_variable` public

- `object.c (Init_Object)`: make `remove_instance_variable` public. [Feature #6539]

**Revision 38114 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)**

`object.c`: make `remove_instance_variable` public

- `object.c (Init_Object)`: make `remove_instance_variable` public. [Feature #6539]

**Revision 38114 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)**

`object.c`: make `remove_instance_variable` public

- `object.c (Init_Object)`: make `remove_instance_variable` public. [Feature #6539]

**Revision 38114 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)**

`object.c`: make `remove_instance_variable` public

- `object.c (Init_Object)`: make `remove_instance_variable` public. [Feature #6539]

**Revision 38114 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)**

`object.c`: make `remove_instance_variable` public

- `object.c (Init_Object)`: make `remove_instance_variable` public. [Feature #6539]

**Revision 38114 - 12/01/2012 09:16 AM - nobu (Nobuyoshi Nakada)**

`object.c`: make `remove_instance_variable` public

- `object.c (Init_Object)`: make `remove_instance_variable` public. [Feature #6539]

**Revision 9c1b3161 - 12/01/2012 09:17 AM - nobu (Nobuyoshi Nakada)**

`complex.c`, `time.c`: make marshal methods private

- `complex.c (Init_Complex)`, `time.c (Init_Time)`: make marshal methods private. [Feature #6539]

git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/trunk@38115](https://ci.ruby-lang.org/ruby/trunk@38115) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 38115 - 12/01/2012 09:17 AM - nobu (Nobuyoshi Nakada)**

`complex.c`, `time.c`: make marshal methods private

- `complex.c (Init_Complex)`, `time.c (Init_Time)`: make marshal methods private. [Feature #6539]

**Revision 38115 - 12/01/2012 09:17 AM - nobu (Nobuyoshi Nakada)**

`complex.c`, `time.c`: make marshal methods private

- `complex.c (Init_Complex)`, `time.c (Init_Time)`: make marshal methods private. [Feature #6539]

**Revision 38115 - 12/01/2012 09:17 AM - nobu (Nobuyoshi Nakada)**

complex.c, time.c: make marshal methods private

- complex.c (Init\_Complex), time.c (Init\_Time): make marshal methods private. [Feature #6539]

**Revision 38115 - 12/01/2012 09:17 AM - nobu (Nobuyoshi Nakada)**

complex.c, time.c: make marshal methods private

- complex.c (Init\_Complex), time.c (Init\_Time): make marshal methods private. [Feature #6539]

**Revision 38115 - 12/01/2012 09:17 AM - nobu (Nobuyoshi Nakada)**

complex.c, time.c: make marshal methods private

- complex.c (Init\_Complex), time.c (Init\_Time): make marshal methods private. [Feature #6539]

**Revision 38115 - 12/01/2012 09:17 AM - nobu (Nobuyoshi Nakada)**

complex.c, time.c: make marshal methods private

- complex.c (Init\_Complex), time.c (Init\_Time): make marshal methods private. [Feature #6539]

**Revision 4f0c9b2b - 12/01/2012 10:01 AM - nobu (Nobuyoshi Nakada)**

marshal.c: private methods

- marshal.c (w\_object, r\_object0): call private marshal methods. [Feature #6539]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38118 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 38118 - 12/01/2012 10:01 AM - nobu (Nobuyoshi Nakada)**

marshal.c: private methods

- marshal.c (w\_object, r\_object0): call private marshal methods. [Feature #6539]

**Revision 38118 - 12/01/2012 10:01 AM - nobu (Nobuyoshi Nakada)**

marshal.c: private methods

- marshal.c (w\_object, r\_object0): call private marshal methods. [Feature #6539]

**Revision 38118 - 12/01/2012 10:01 AM - nobu (Nobuyoshi Nakada)**

marshal.c: private methods

- marshal.c (w\_object, r\_object0): call private marshal methods. [Feature #6539]

**Revision 38118 - 12/01/2012 10:01 AM - nobu (Nobuyoshi Nakada)**

marshal.c: private methods

- marshal.c (w\_object, r\_object0): call private marshal methods. [Feature #6539]

**Revision 38118 - 12/01/2012 10:01 AM - nobu (Nobuyoshi Nakada)**

marshal.c: private methods

- marshal.c (w\_object, r\_object0): call private marshal methods. [Feature #6539]

**Revision 38118 - 12/01/2012 10:01 AM - nobu (Nobuyoshi Nakada)**

marshal.c: private methods

- marshal.c (w\_object, r\_object0): call private marshal methods. [Feature #6539]

**Revision ccebd9b0 - 12/01/2012 01:35 PM - nobu (Nobuyoshi Nakada)**

vm\_method.c: private

- vm\_method.c (basic\_obj\_respond\_to): call even if private. [Feature #6539]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38123 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 38123 - 12/01/2012 01:35 PM - nobu (Nobuyoshi Nakada)**

vm\_method.c: private

- vm\_method.c (basic\_obj\_respond\_to): call even if private. [Feature #6539]

**Revision 38123 - 12/01/2012 01:35 PM - nobu (Nobuyoshi Nakada)**

vm\_method.c: private

- vm\_method.c (basic\_obj\_respond\_to): call even if private. [Feature #6539]

**Revision 38123 - 12/01/2012 01:35 PM - nobu (Nobuyoshi Nakada)**

vm\_method.c: private

- vm\_method.c (basic\_obj\_respond\_to): call even if private. [Feature #6539]

**Revision 38123 - 12/01/2012 01:35 PM - nobu (Nobuyoshi Nakada)**

vm\_method.c: private

- vm\_method.c (basic\_obj\_respond\_to): call even if private. [Feature #6539]

**Revision 38123 - 12/01/2012 01:35 PM - nobu (Nobuyoshi Nakada)**

vm\_method.c: private

- vm\_method.c (basic\_obj\_respond\_to): call even if private. [Feature #6539]

**Revision 38123 - 12/01/2012 01:35 PM - nobu (Nobuyoshi Nakada)**

vm\_method.c: private

- vm\_method.c (basic\_obj\_respond\_to): call even if private. [Feature #6539]

**Revision 4d73f950 - 12/03/2012 10:10 AM - nobu (Nobuyoshi Nakada)**

random.c, rational.c: make marshal methods private

- random.c (Init\_Random), rational.c (Init\_Rational): make marshal methods private. [Feature #6539]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38157 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 38157 - 12/03/2012 10:10 AM - nobu (Nobuyoshi Nakada)**

random.c, rational.c: make marshal methods private

- random.c (Init\_Random), rational.c (Init\_Rational): make marshal methods private. [Feature #6539]

**Revision 38157 - 12/03/2012 10:10 AM - nobu (Nobuyoshi Nakada)**

random.c, rational.c: make marshal methods private

- random.c (Init\_Random), rational.c (Init\_Rational): make marshal methods private. [Feature #6539]

**Revision 38157 - 12/03/2012 10:10 AM - nobu (Nobuyoshi Nakada)**

random.c, rational.c: make marshal methods private

- random.c (Init\_Random), rational.c (Init\_Rational): make marshal methods private. [Feature #6539]

**Revision 38157 - 12/03/2012 10:10 AM - nobu (Nobuyoshi Nakada)**

random.c, rational.c: make marshal methods private

- random.c (Init\_Random), rational.c (Init\_Rational): make marshal methods private. [Feature #6539]

**Revision 38157 - 12/03/2012 10:10 AM - nobu (Nobuyoshi Nakada)**

random.c, rational.c: make marshal methods private

- random.c (Init\_Random), rational.c (Init\_Rational): make marshal methods private. [Feature #6539]

**Revision 38157 - 12/03/2012 10:10 AM - nobu (Nobuyoshi Nakada)**

random.c, rational.c: make marshal methods private

- random.c (Init\_Random), rational.c (Init\_Rational): make marshal methods private. [Feature #6539]

## History

---

### #1 - 06/06/2012 10:13 AM - marcandre (Marc-Andre Lafortune)

- Assignee set to matz (Yukihiko Matsumoto)

No comments, so I'm assigning to Matz.

There are more than 150 older issues assigned to Matz already.

### #2 - 06/13/2012 02:17 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

### #3 - 10/27/2012 01:58 PM - matz (Yukihiko Matsumoto)

- Assignee changed from matz (Yukihiko Matsumoto) to nobu (Nobuyoshi Nakada)

Concrete counter proposal:

Should be made private:

Object and descendants

#initialize\_clone

#initialize\_dup

#respond\_to\_missing?

Rational & Complex

#marshal\_dump

#marshal\_load

Time

#\_dump

.\_load

Note that Delegate#initialize\_{clone|dup} are already private

Should be made public:

Object

#remove\_instance\_variable

I think class/module operations should be done in the scope.

Matz.

### #4 - 10/27/2012 07:19 PM - mame (Yusuke Endoh)

- Target version changed from 2.0.0 to 2.6

### #5 - 12/01/2012 06:16 PM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r38113.

Marc-Andre, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

---

vm\_method.c: make initialize methods private

- id.c (Init\_id), template/id.h.tmpl: add initialize\_{copy,clone,dup} and respond\_to\_missing?.
- vm\_method.c (rb\_method\_entry\_make): make above methods private. [Feature #6539]

### #6 - 11/25/2016 09:12 AM - ko1 (Koichi Sasada)

- Description updated

### #7 - 11/25/2016 01:38 PM - shyouhei (Shyouhei Urabe)

- Related to Feature #12697: Why shouldn't Module meta programming methods be public? added

**#8 - 11/28/2017 02:41 AM - marcandre (Marc-Andre Lafortune)**

- Related to Feature #14132: `Module#attr{[_reader|_writer]}` should be public added

**#9 - 11/28/2017 03:26 AM - marcandre (Marc-Andre Lafortune)**

- Related to Feature #14133: `Module#{define|alias|undef}_method` should be made public added