

## Ruby master - Feature #9768

### Method that is visible only within a certain module/class

04/22/2014 09:57 AM - sawa (Tsuyoshi Sawada)

<b>Status:</b>	Assigned
<b>Priority:</b>	Normal
<b>Assignee:</b>	matz (Yukihiro Matsumoto)
<b>Target version:</b>	
<b>Description</b>	
<p>Some frameworks/libraries monkeypatch their own methods on Ruby core classes like String, Hash, Array, etc., and that is often causing problems/concerns of name conflict.</p> <p>Seeing that these custom methods are used only in the context of a certain module/class, I request for a way to define a method (foo) on a module/class (A) so that it will be visible only from within a specified module/class (B) or its descendants. The following illustrates this situation:</p> <pre>A.new.foo # =&gt; undefined  class B   A.new.foo # =&gt; defined   def bar     A.new.foo # =&gt; defined   end   def self.baz     A.new.foo # =&gt; defined   end end  class C &lt; B   A.new.foo # =&gt; defined   def bar     A.new.foo # =&gt; defined   end   def self.baz     A.new.foo # =&gt; defined   end end</pre> <p>I do not have a certain syntax for this in mind, but I think it can be made much simpler, compared to the complicated syntax of refinement.</p> <p>This is reminiscent of refinement, but they are pretty much different.</p> <p>Refinement's purpose is to let certain methods be accessible from only a certain file. A typical use case would be a library developer defining their methods to be used from within the library and making such methods inaccessible from the end user.</p> <p>On the other hand, the idea I am proposing here is for making method accessible from any file, but only within a certain module/class. A typical use case would be defining a method to be used by an end user, but only from within a context of certain module/class.</p>	

#### History

#1 - 04/25/2014 06:43 AM - shugo (Shugo Maeda)

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

Tsuyoshi Sawada wrote:

Seeing that these custom methods are used only in the context of a certain module/class, I request for a way to define a method (foo) on a module/class (A) so that it will be visible only from within a specified module/class (B) or its descendants. The following illustrates this situation:

The original proposal of Refinements was able to be used in such a situation, but refinement inheritance in class hierarchies and refinement activation for reopened module definitions were removed (see <https://bugs.ruby-lang.org/issues/4085#note-141> and other comments).

Matz said that a refinement should be activated only in a lexical scope explicitly specified, because users might be confused if methods are refined implicitly.