Module#(define|alias|undef|remove)_method should be made public

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

Status: Closed
Priority: Normal
Assignee: matz (Yukihiro Matsumoto)
Target version: 

Description

Modules and classes can be reopened and changed (unless frozen).

This is used in many meta programming techniques.

Currently, define_method is private, so we need to either do a class_eval, reopen the class somehow, or resort to send.

As I previously stated in #6539, I feel that the use of send should be reserved for incorrect usage of actually private methods that might change of interface or aren't meant to be called this way (e.g. respond_to_missing?, puts)

Matz has stated before that "class/module operations should be done in the scope.". Nevertheless, common usage shows that there are many cases where Rubyists prefer using a single line for this, even if it means having to call send.

Here are 95k+ examples of send :define_method in the wild:
https://github.com/search?utf8=%E2%9C%93&q=language%3Aruby+%22send+%3Adefine_method%22&type=Code

Note that many of these are very bad usage of define_method. I feel this shows that even beginner programmers will use send if privacy gets in the way. Let's not hinder advanced users in a failed attempt to protect the beginners.

Rails has good examples of use of send :define_method like
https://github.com/rails/rails/blob/master/actionview/lib/action_view/lookup_context.rb#L27
That's despite the fact for most metaprogramming they don't use define_method and instead build a string of code and do a module_eval for both performance and ease debugging (see the next lines in the example). That's not possible when the new methods must access blocks though, like in the example given.

alias_method and undef_method can be seen as special cases of define_method and should have the same privacy.
55k+ examples of send :alias_method in the wild:
https://github.com/search?utf8=%E2%9C%93&q=language%3Aruby+%22send+%3Aalias_method%22&type=Code
30k+ examples of send :undef_method in the wild:
https://github.com/search?utf8=%E2%9C%93&q=language%3Aruby+%22send+%3Aundef_method%22&type=Code
20k+ examples of send :remove_method in the wild:
https://github.com/search?utf8=%E2%9C%93&q=language%3Aruby+%22send+%3Aundef_method%22&type=Code

Related issues:

Related to Ruby master - Feature #14132: Module#attr[[:reader][:writer]] should... Closed
Related to Ruby master - Feature #6539: public and private for core methods Closed

Associated revisions

Revision 60942 - 11/29/2017 05:47 PM - marcandre (Marc-Andre Lafortune)
Make Module#(define|alias|undef|remove)_method public [#14133]

Revision 60942 - 11/29/2017 05:47 PM - marcandre (Marc-Andre Lafortune)
Make Module#(define|alias|undef|remove)_method public [#14133]

Revision 60942 - 11/29/2017 05:47 PM - marcandre (Marc-Andre Lafortune)
Make Module#(define|alias|undef|remove)_method public [#14133]

Revision ac1193d3 - 07/25/2018 06:06 PM - tenderlove
Remove obsolete comment from Module#define_method documentation
Since 2.5, Module#define_method is public. (feature #14133)
Revision 64057 - 07/25/2018 06:06 PM - tenderlovemaking (Aaron Patterson)
Remove obsolete comment from Module#define_method documentation
Since 2.5, Module#define_method is public. (feature #14133)

Revision c8f77d37 - 04/30/2019 12:55 PM - usa (Usaku NAKAMURA)
merge revision(s) 64057: [Backport #14952]
Remove obsolete comment from Module#define_method documentation
Since 2.5, Module#define_method is public. (feature #14133)

Revision 67703 - 04/30/2019 12:55 PM - usa (Usaku NAKAMURA)
merge revision(s) 64057: [Backport #14952]
Remove obsolete comment from Module#define_method documentation
Since 2.5, Module#define_method is public. (feature #14133)

History

#1 - 11/28/2017 03:26 AM - marcandre (Marc-Andre Lafortune)
- Related to Feature #14132: Module#attr{[:reader,:writer]} should be public added

#2 - 11/28/2017 03:26 AM - marcandre (Marc-Andre Lafortune)
- Related to Feature #6539: public and private for core methods added

#3 - 11/28/2017 05:54 AM - shevegen (Robert A. Heiler)

Note that many of these are very bad usage of define_method. I feel this shows that even beginner programmers will use :send if privacy gets in the way. Let's not hinder advanced users in a failed attempt to protect the beginners.

_send() is simply awesome. It's a basic OOP way - you send your message to the object, which will have a look as to what to do, without any fuzz.

The rails code looks very alien to me and not very pretty. A lot of the metaprogramming makes the code hard to read IMO.

I think that people use _send() because it is so much simpler than the alternatives. One could then also say that the alternatives may be too complex or complicated. Lots of eval-methods and it is not always easy to infer which eval method is now the right one or how to use it.

Some weird code usage is out there too:
To your suggestion itself, I have nothing against it, but I guess matz has to decide; and perhaps this may have to be for ruby 3.x rather than the ruby 2.x branch.

#4 - 11/29/2017 06:55 AM - matz (Yukihiro Matsumoto)
OK, I understand. They will be public in 2.5

Matz.

#5 - 11/29/2017 05:49 PM - marcandre (Marc-Andre Lafontune)
- Status changed from Open to Closed

Great, thank you!

Done.