Ruby master - Feature #10018

Consider adding Sub-Includes as in include Foo::bar

07/09/2014 11:48 AM - shevegen (Robert A. Heiler)

Status:	Feedback	
Priority:	Normal	
Assignee:		
Target version:		

Description

On IRC someone asked a nice question:

> <arup_r> If I write class Foo; include Enumerable; end This code include all methods from Enumerable. But suppose I want to include #all?, #any? .. What's the way to include only required methods ?

This prompted me to write this.

What if ruby would allow us to include just singular methods?

Something like:

```
include Enumerable::any?
```

And so forth, so that we could cherry-pick what we want.

This is already possible in some projects, like facets.

But I think they solve it by splitting up the whole
stuff into individual files, and then including
this.

I think it would be nicer if include itself would
support a scope-mechanic.

History

#1 - 07/09/2014 11:49 AM - shevegen (Robert A. Heiler)

Oops sorry!

This is not a bug, it is a feature request.

I am not sure how to move it to features now, sorry. :(

#2 - 07/09/2014 01:55 PM - nobu (Nobuyoshi Nakada)

- Tracker changed from Bug to Feature

- Description updated

- Status changed from Open to Feedback

You can achieve it by "method transplanting".

```
class Foo
  define_method(:any?, Enumerable.instance_method(:any?))
end
```

Wrapper method like Module#mix may help you.
http://www.slideshare.net/yukihiro_matz/rubyconf-2010-keynote-by-matz

It has been removed finally, you can construct

#3 - 07/10/2014 12:49 AM - matz (Yukihiro Matsumoto)

And method transplanting only works when the method does not call super nor other Enumerable methods.
Automatic resolving is possible, but it would make inclusion far more complex.

Matz.