Ruby master - Feature #1102

Prepend Module

02/04/2009 11:37 AM - trans (Thomas Sawyer)

Status:	Closed	
Priority:	Normal	
Assignee:	nobu (Nobuyoshi Nakada)	
Target version:	2.0.0	

Description

=begin
Currently when a module is included into a classes, it is appended to the class hierarchy (ie. the method lookup order). This of course makes sense, but there are times when it would be useful to *prepend* the module. For example:

class C
def x; "x"; end
end

module M
def x; '[' + super + ']'; end
end

class C
prepend M
end

C.new.x #=> "[x]"

One big advantage of this is being able to override methods in a safer way, rather than using alias or tricks like alias_method_chain.
=end

Related issues:

Related to Ruby master - Feature #6452: Allow extend to override class methods	**Assigned**	**05/19/2012**

Associated revisions

Revision 8ddbbb33 - 06/27/2012 07:48 AM - nobu (Nobuyoshi Nakada)

Module#prepend

- class.c (rb_prepend_module): prepend module into another module.
- eval.c (rb_mod_prepend): new method Module#prepend. [Feature #1102]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36234 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 36234 - 06/27/2012 07:48 AM - nobu (Nobuyoshi Nakada)

Module#prepend

- class.c (rb_prepend_module): prepend module into another module.
- eval.c (rb_mod_prepend): new method Module#prepend. [Feature #1102]

Revision 36234 - 06/27/2012 07:48 AM - nobu (Nobuyoshi Nakada)

Module#prepend

- class.c (rb_prepend_module): prepend module into another module.
- eval.c (rb_mod_prepend): new method Module#prepend. [Feature #1102]

Revision 36234 - 06/27/2012 07:48 AM - nobu (Nobuyoshi Nakada)

Module#prepend

- class.c (rb_prepend_module): prepend module into another module.
- eval.c (rb_mod_prepend): new method Module#prepend. [Feature #1102]

Revision 36234 - 06/27/2012 07:48 AM - nobu (Nobuyoshi Nakada)

Module#prepend

- class.c (rb_prepend_module): prepend module into another module.
- eval.c (rb_mod_prepend): new method Module#prepend. [Feature #1102]

Revision 36234 - 06/27/2012 07:48 AM - nobu (Nobuyoshi Nakada)

Module#prepend

- class.c (rb_prepend_module): prepend module into another module.
- eval.c (rb_mod_prepend): new method Module#prepend. [Feature #1102]

Revision 36234 - 06/27/2012 07:48 AM - nobu (Nobuyoshi Nakada)

Module#prepend

- class.c (rb_prepend_module): prepend module into another module.
- eval.c (rb_mod_prepend): new method Module#prepend. [Feature #1102]

History

#1 - 03/20/2009 06:53 PM - trans (Thomas Sawyer)

=begin

On Feb 7, 10:42 pm, Roger Pack rogerdp...@gmail.com wrote:

> Currently when a module is included into a classes, it is appended to the class hierarchy (ie. > the method lookup order). This of course makes sense, but there are times when it would be > useful to *prepend* the module. For example:

> I suppose one [not too useful] hack at it could be something like
>
> ```
> class Class
>   def insert_module_at_top mod
>     previous_ancestors = self.ancestors.select{|a| a.class == Module}
>     include mod
>     previous_ancestors.each{|a| include a.dup}
>   end
> end
>
> #again we have to start with a module.
>
> module Original
>   def x; '[' + super + ']'; end
> end
>
> class A
>  include Original
> end
>
> modulePrepend
>  def x; "x"; end
> end
> A.insert_module_at_topPrepend
> puts A.new.x
> => [x]
> ```
>
> Perhaps what we're saying here is we wish we could "grab" methods from classes, to be able to manipulate the hierarchy better? This is possible with RubyParser, since you can basically get back to the source of the method and thus copy it around, but not afaik with 1.9

Well, that's not really the issue here. The need is to *wrap* previously defined instance methods. If every method were defined in a module (something I have suggested in the past actually) then it would not be needed.

The utility comes from doing AOP-esque coding. Consider modules that can initialize values.

```
class X
end

module P
```

```
attr :p
def initialize
@p = []
super
end
end

class X
prepend P
end

X.new.p  => []

T.

=end
```

#2 - 09/18/2009 04:21 AM - marcandre (Marc-Andre Lafortune)

- Assignee set to matz (Yukihiro Matsumoto)

```
=begin

=end
```

#3 - 03/02/2010 04:47 AM - rogerdpack (Roger Pack)

```
=begin
+1 for Module#prepend
=end
```

#4 - 03/02/2010 10:49 AM - manveru (Michael Fellinger)

```
=begin
+1 for Module#prepend
=end
```

#5 - 03/02/2010 12:40 PM - mame (Yusuke Endoh)

- Target version changed from 2.0.0 to 3.0

```
=begin
Hi,

This ticket was also discussed in the thread from [ruby-core:25208].

Module#prepend may be very significant feature not only to implementation
but also to Ruby's OO model itself.
Don't consider it just convenient method like Array's and String's.

So, in my opinion, this feature should not be included in 1.9.x.
We should discuss it towards 2.0.

Even if it will be included in 1.9.x, we need more discussion.
Just seeing clean example, you'll find it cool.  But in fact, we must
also discuss many dirty things:

- edge semantics
  - prepend into embedded class
  - prepend into singleton class
  - collaboration with reflection
  - collaboration with future expansion (e.g., classbox)
  - etc.

- implementation
  - robustness
  - binary compatibility
  - expandability
  - maintainability
  - performance
```

I think it is difficult to discuss them without material. So, please
write a patch first if you really want.

--
Yusuke Endoh mame@tsg.ne.jp
=end

#6 - 03/04/2010 01:28 AM - mame (Yusuke Endoh)

- Status changed from Open to Feedback

=begin

=end

#7 - 10/18/2011 09:16 AM - naruse (Yui NARUSE)

- Project changed from Ruby master to CommonRuby

- Category deleted (core)

- Target version deleted (3.0)

#8 - 10/23/2011 05:21 PM - naruse (Yui NARUSE)

- Project changed from CommonRuby to Ruby master

#9 - 02/13/2012 09:42 PM - mame (Yusuke Endoh)

This feature is listed as matz's "must-have." [ruby-core:39837]
Are there any volunteers to be a facilitator, to create a prototype,
and/or, to study the semantics and implementation?
Sorry I'll have no time, and completely forgot the discussion.

```
  ___
 _|*|_
 (`_`)
 m9

I WANT YOU
```

--
Yusuke Endoh mame@tsg.ne.jp

#10 - 02/26/2012 05:00 AM - ko1 (Koichi Sasada)

- Assignee changed from matz (Yukihiro Matsumoto) to ko1 (Koichi Sasada)

- Target version set to 2.0.0

#11 - 06/26/2012 04:00 AM - ko1 (Koichi Sasada)

- Assignee changed from ko1 (Koichi Sasada) to nobu (Nobuyoshi Nakada)

Nobu will commit it.

#12 - 06/27/2012 04:48 PM - nobu (Nobuyoshi Nakada)

- Status changed from Feedback to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r36234.
Thomas, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

Module#prepend

- class.c (rb_prepend_module): prepend module into another module.
- eval.c (rb_mod_prepend): new method Module#prepend. [Feature #1102]