Ruby master - Feature #7835

autoload will be dead

02/12/2013 10:26 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	

Description

A while ago matz declared his intention to remove the autoload feature from the Ruby language. It was sent to the ruby-core mailing list but no ticket has been created for it.

I'm creating the ticket now so that we could discuss the idea here.

Posted by Yukihiro Matsumoto on 2011-11-19 08:12

Hi,

Today, I talked with NaHi about enhancing const_missing to enable
autoload-like feature with nested modules. But autoload itself has
fundamental flaw under multi-thread environment. I should have remove
autoload when I added threads to the language (threads came a few
months after autoload).

So I hereby declare the future deprecation of autoload. Ruby will
keep autoload for a while, since 2.0 should keep compatibility to 1.9.
But you don't expect it will survive further future, e.g. 3.0.

I strongly discourage the use of autoload in any standard libraries.

```
                  matz.
```

Related issues:

Related to Ruby master - Feature #5653: "I strongly discourage the use of aut...	**Closed**

History

#1 - 02/12/2013 10:26 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

+1 for removing autoload. It not only has troubles with thread safety, which I believe is fixable (although I confess I haven't thought much about it).

There is a more serious issue in my opinion that is unfixable with current specs.

For a while I've been experiencing a subtle bug in my Rails application that only happened on development mode (when autoload is enabled) and only for the first request. Recently I decided to understand what was happening and Xavier Noria showed me what was happening.

The problem is that I had some custom classes that relied on ActionView::Helpers::NumberHelper. Since all code in number_helper.rb was self-contained and it had no extra dependencies on ActionView, I just used it directly and my unit tests passed and my application always worked except for the first request when my custom class was used by the controller.

The problem is that once Rails is loaded it will declare an auto-load dependency on ActionView. Then when I "require 'action_view/helpers/number_helper'" that file will be something like:

module ActionView
module Helpers
module NumberHelper

But since Helpers is set up to use autoload, once "module Helpers" is found, it will load "action_view/helpers.rb". And this file will then require "action_view/helpers/number_helper.rb" which causes a circular dependency problem.

Here is a complete Ruby-only code exemplifying the issue:

./test.rb:

```
autoload :A, 'a'
require 'a/b'
```

./lib/a.rb:

```
require 'a/b'
```

./lib/a/b.rb:

```
module A
module B
end
end
```

```
ruby -I lib test.rb
```

This kind of bug may be hard to track, so I agree with Matz that autoload should be dead.

#2 - 02/12/2013 10:32 PM - mame (Yusuke Endoh)

- Status changed from Open to Rejected

Matz created #5653 by himself. Please reply into that ticket :-)

--
Yusuke Endoh mame@tsg.ne.jp

#3 - 02/12/2013 10:37 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Thanks, Yusuke! I tried to search it on Redmine but I couldn't easily find it nor I could filter for issues created by matz, for instance...

#4 - 11/23/2018 08:23 AM - matz (Yukihiro Matsumoto)

- Related to Feature #5653: "I strongly discourage the use of autoload in any standard libraries" (Re: autoload will be dead) added