Ruby - Feature #12138

Support `Kernel#load_with_env(filename, cbase: SomeMod, cref: someMod, binding: SomeMod) # => obj`

03/02/2016 06:08 PM - josh.cheek (Josh Cheek)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	

Description

Ruby's require and load methods currently only return true, which means that any code being loaded must affect the global namespace, or have the global namespace affected for it to interact with.

This means that to load data, like a gemspec, you have to read the file in and eval it:

- Rubygems
- Bundler

To load a plugin or configuration, you have to modify a global object:

- Railties hook into subclassing
- RSpec tests have to be declared to a global object
- Minitest, too
- Rake extends main to store results on a global object

It's why gems have to be built the way they do, to avoid colliding with other code in the global namespace (fwiw, I've seen a computer segfault when the person unwittingly write their own Queue class). Which is also why we can't load multiple versions of a gem in the same namespace.

History

#1 - 03/04/2016 07:04 AM - shevegen (Robert A. Heiler)

+1 for the spirit of the suggestion

Perhaps ruby 3.x will also feature some more advanced import-like system
fitting to ruby. :) Manipulating environments or objects-in-environments
would be nice.

I have no particular strong feeling in favour or disfavour of the syntax
but it seems a bit hard to remember, the
(filename, cbase: SomeMod, cref: someMod, binding: SomeMod)

#2 - 03/05/2016 04:42 PM - josh.cheek (Josh Cheek)

Robert A. Heiler wrote:

> +1 for the spirit of the suggestion
>
> Perhaps ruby 3.x will also feature some more advanced import-like system
> fitting to ruby. :) Manipulating environments or objects-in-environments
> would be nice.
>
> I have no particular strong feeling in favour or disfavour of the syntax
> but it seems a bit hard to remember, the
> (filename, cbase: SomeMod, cref: someMod, binding: SomeMod)

Aye, I have no preference either, I chose those names just to make it clear what things I want to be able to set. Ie RSpec.describe ... would be fine if I could choose what object RSpec evaluates to in that file. Or toplevel method defs would be fine if I could choose what object they get defined on. By giving it a binding to evaluate within, I can choose what its self should be, and provide it with local variables. I don't particularly care how this is achieved.

#3 - 07/19/2016 06:09 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Feedback

I understand the need for load() with context, but this proposal has

- a bad name, "env" is ambiguous
- the proposed context (cbase, cref, binding) is not proven sufficient

So you (or someone) have to improve the proposal.

Matz.

#4 - 09/29/2016 11:44 PM - josh.cheek (Josh Cheek) *2/2*

Yukihiro Matsumoto wrote:

> I understand the need for load() with context, but this proposal has
>
> - a bad name, "env" is ambiguous
> - the proposed context (cbase, cref, binding) is not proven sufficient
>
> So you (or someone) have to improve the proposal.
>
> Matz.

I don't understand what "proven sufficient" entails. Is there a metric I could use to know it was achieved?