

## Ruby trunk - Feature #11415

### autoload with a Proc

08/05/2015 02:28 AM - matthewd (Matthew Draper)

<b>Status:</b>	Open
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	
<b>Description</b>	
autoload currently allows you to run arbitrary code when a constant is referenced, as long as that code is written to a file.	
I propose extending autoload to also accept a Proc in place of the string filename.	
Usage:	
<pre>autoload :Foo, -&gt; { ::Object.const_set :Foo, 42 } autoload :Bar, -&gt; { require "bar_1"; require "bar_2" } autoload :Baz, -&gt; { with_special_lock { require "baz" } }</pre>	
No built-in concurrency protection is provided: if a second thread encounters a constant reference before the first one sets the constant, the second thread will also enter the Proc.	
Under simple usage, the proc will still call require at some point -- then require's locking will take effect. A more advanced user can apply their own locking.	
I believe this is sufficient to allow the Rails dependency loader to use autoload instead of const_missing, and fix the "nested constants" problem.	
The implementation is quite small, and it doesn't seem to make the dangers of autoload any worse than they already are. Hopefully, it can provide an easier place for users to experiment in ways to save autoload by making it more generally safe.	
<i>Alternatives</i>	
For Rails's purposes, nested constants could also be solved by a missing_constant (or similar) macro:	
<pre>module Foo; end module Bar; end  Foo::Bar # =&gt; returns ::Bar  module Foo; missing_constant :Bar; end module Bar  Foo::Bar # =&gt; invokes Foo.const_missing(:Bar)</pre>	
More directly, instead of passing a Proc to autoload, a caller can write the code to a tempfile, and then specify that tempfile's path.	

### History

#### #1 - 10/21/2015 12:53 PM - matz (Yukihiro Matsumoto)

I hate autoload. autoload is one of things I regret. I'm not positive about enhancing something I hate.  
-- matz in <https://speakerdeck.com/skade/the-dark-side-of-matz-1>

How about using const\_missing?

Matz.

#### #2 - 10/21/2015 08:38 PM - normalperson (Eric Wong)

[matz@ruby-lang.org](mailto:matz@ruby-lang.org) wrote:

I hate autoload. autoload is one of things I regret. I'm not positive about enhancing something I hate.  
-- matz in <https://speakerdeck.com/skade/the-dark-side-of-matz-1>

Would you like autoload if Ruby didn't have threads? :)

Anyways, I think autoload is somewhat useful for improving startup times; but the interaction with threads is currently dangerous.

How about using `const_missing`?

Using that would be even more fragile, I think.

## Files

---

autoload-proc.patch	5.83 KB	08/05/2015	matthewd (Matthew Draper)
---------------------	---------	------------	---------------------------