

Ruby - Bug #12688

Thread unsafety in autoload

08/19/2016 05:13 AM - headius (Charles Nutter)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:		
ruby -v:	2.3.0	Backport: 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

I need clarification here. I expected, based on Ruby's assertion that autoloads are thread-safe, that the following code would never error. Instead, it gets a couple iterations in and raises NameError:

```
loop do
  class Foo
    autoload :Bar, 'bar.rb'
  end

  go = false
  threads = (1..50).map {Thread.new { 1 until go; print '.'; Foo.const_get(:Bar) }}
  go = true
  threads.each(&:join)
  puts

  self.class.send :remove_const, :Foo
end
```

And the output with Ruby 2.3.0:

```
$ ruby23 -I. autoload_breaker.rb
.....
.....autoload_breaker.rb:7:in `const_get': uninitiali
zed constant Foo::Bar (NameError)
Did you mean?  Foo::Bar
  from autoload_breaker.rb:7:in `block (3 levels) in <main>'
```

Is there something wrong with my script? Is my expectation incorrect?

History

#1 - 08/19/2016 05:14 AM - headius (Charles Nutter)

Oh, sorry...the source of bar.rb is trivial:

```
class Foo
  Bar = 1
end
```

#2 - 08/22/2016 07:23 AM - h.shirosaki (Hiroshi Shirosaki)

Charles Nutter wrote:

I need clarification here. I expected, based on Ruby's assertion that autoloads are thread-safe, that the following code would never error. Instead, it gets a couple iterations in and raises NameError:

```
loop do
  class Foo
    autoload :Bar, 'bar.rb'
  end

  go = false
  threads = (1..50).map {Thread.new { 1 until go; print '.'; Foo.const_get(:Bar) }}
  go = true
  threads.each(&:join)
```

```
puts

self.class.send :remove_const, :Foo
end
```

Is there something wrong with my script? Is my expectation incorrect?

\$.pop would be needed to clear bar.rb in loaded features.
I don't get NameError after adding \$.pop.

#3 - 08/23/2016 12:20 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

I have the same expectations as you, Charles, and as so I'd also expect this to be a bug.

#4 - 09/22/2016 05:39 PM - headius (Charles Nutter)

\$.pop would be needed to clear bar.rb in loaded features.
I don't get NameError after adding \$.pop.

But why does it work for a while and then stop working? There still seems to be a threading issue here.

Here's my modified script, with the confirmation dots moved after the constant lookup and Thread.abort_on_exception = true:

```
Thread.abort_on_exception = true

loop do
  class Foo
    autoload :Bar, 'bar.rb'
  end

  go = false
  threads = (1..50).map {Thread.new { 1 until go; Foo.const_get(:Bar); print '.' }}
  go = true
  threads.each(&:join)
  puts

  self.class.send :remove_const, :Foo
end
```

It successfully runs for less than 50 dots, and then one of the threads errors out. I don't think it should.

```
$ ruby23 -I. blah.rb
.....
blah.rb:9:in `const_get': uninitialized constant Foo::Bar (NameError)
Did you mean?  Foo::Bar
from blah.rb:9:in `block (3 levels) in <main>'
```

Note also this is not even getting to a second iteration; the first iteration of the outer loop fails. If I add \$.pop after remove_const, it does run for longer...but it still produces a NameError for me.

```
$ ruby23 -I. blah.rb
.....
.....
<about 100 rows omitted>
.....
.....
.....
blah.rb:9:in `const_get': uninitialized constant Foo::Bar (NameError)
Did you mean?  Foo::Bar
from blah.rb:9:in `block (3 levels) in <main>'
```

#5 - 09/23/2016 01:00 PM - headius (Charles Nutter)

It successfully runs for less than 50 dots, and then one of the threads errors out. I don't think it should.

Ok, I can't count...it does run through one full iteration and then probably fails because the autoload doesn't actually load anything. But then it still fails a hundred iterations later, so something's still not right.

#6 - 11/25/2016 08:20 AM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned
- Assignee set to ko1 (Koichi Sasada)

#7 - 11/25/2016 01:35 PM - shyouhei (Shyouhei Urabe)

We looked at this issue in today's developer meeting and had 2 feelings in common.

1. The autoload should not render NameError. It definitely is a bug that must be fixed. Ko1 is assigned.
2. Besides, we want to discourage people from doing something like 1 until go. JRuby+Truffle might optimize this out (as far as I understand). Also generally speaking, sharing local variables across threads is something difficult to do properly.

#8 - 01/26/2017 10:38 AM - ko1 (Koichi Sasada)

- Status changed from Assigned to Feedback

I can't reproduce headius's issue. It shows 50 dots and stop at next iteration because autoload is failed.

Inserting `$.pop` Shirosaki san suggested, I don't get any exception.

I tried on current trunk.

```
$LOAD_PATH.unshift __dir__

Thread.abort_on_exception = true

loop do
  class Foo
    autoload :Bar, 'bar.rb'
  end

  go = false
  threads = (1..50).map {Thread.new { 1 until go; Foo.const_get(:Bar); print '!' }}
  go = true
  threads.each(&:join)
  puts

  self.class.send :remove_const, :Foo
  $.pop
end
```

#9 - 07/25/2019 05:48 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Feedback to Closed