

Ruby - Bug #21384

const_added is triggered twice when using autoload

05/29/2025 11:02 PM - petekinnecom (Pete Kinnecom)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:		Backport: 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

As the title says, I've noticed that const_added is invoked twice when using autoload. I'm wondering if this behavior is intended or perhaps a bug. Here's a small script to reproduce what I'm seeing:

```
require "tmpdir"

dir = Dir.mktmpdir

File.write(
  File.join(dir, "const.rb"),
  "class Const; end"
)

def Object.const_added(const_name)
  super.tap { puts "const_added: #{const_name}" }
end

$LOAD_PATH << dir

puts "before autoload call"
autoload :Const, "const"
puts "after autoload call"
puts Const

# Produces output:
#
# => before autoload call
# => const_added: Const
# => after autoload call
# => const_added: Const
# => Const
```

I'm seeing the behavior using the following docker containers ruby:3.2 and ruby:3.5-rc when running like so: docker run -v ./:/home ruby:3.5-rc ruby /home/script.rb. Thanks.

History

#1 - 05/30/2025 03:30 PM - fxn (Xavier Noria)

It is intended.

Reason is, constants includes the constant after a call to autoload.

#2 - 05/30/2025 03:32 PM - fxn (Xavier Noria)

Let me add that I'd prefer that Ruby does not consider autoloads as constants in the API, I'd like autoloads to become constants if realized (autoloading can err, and even if succeeds, the owner may not be the receiver of the autoload call).

But that is the way it works.

#3 - 05/30/2025 11:56 PM - petekinnecom (Pete Kinnecom)

I see, thank you for the clarification!

#4 - 06/03/2025 07:56 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

#5 - 06/05/2025 11:00 AM - mame (Yusuke Endoh)

Briefly discussed at the dev meeting. [@matz \(Yukihiro Matsumoto\)](#) said that it was not intentional to fire the hook twice. But changing it now would be a compatibility concern, so he decided to keep the current behavior.

Conceptually, we may consider this behavior as:

- When autoload is set, the constant is (virtually) defined (and `const_added` fires)
- When the constant is actually defined by the firing of autoload, the constant is conceptually deleted once and defined again (and `const_added` fires again)