

Ruby - Bug #21314

Kernel#autoload requires the wrong file (?)

05/09/2025 03:52 AM - mame (Yusuke Endoh)

Status:Open Priority:Normal Assignee: Target version: ruby -v:	Backport:3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN
Description While discussing #21154 with some people at the dev meeting, I found a weird behavior of Kernel#autoload that's hard to explain. I haven't yet looked at it carefully, but I was asked to create a ticket anyway. <pre># main.rb autoload :A, "./a" autoload :B, "./b" p A # fires the autoload of A # a.rb p :A1 B # fires the autoload of B class A end # b.rb p :B1 class B p :B2 p A # expected: requires no file (because a.rb is already being require'd), and raises a NameError # actual: the autoload of B is fired (not A!), recursively (not no-op!) p :B3 end \$ ruby main.rb :A1 :B1 :B2 :B1 # What? :B2 :A1 :A2 :A3 A :B3 b.rb:17:in '<class:B>': uninitialized constant B::A (NameError)</pre> By the way, in this case, a.rb and b.rb are considered dependent each other. We need to consider how autoload should behave in this case. If it is the user's responsibility to avoid such a situation, we would like to be able to clearly explain what kind of situation should be avoided.	
Related issues: Related to Ruby - Misc #21154: Document or change Module#autoload? Feedback	

History

- #1 - 05/09/2025 03:52 AM - mame (Yusuke Endoh)
- Related to Misc #21154: Document or change Module#autoload? added