

Ruby trunk - Bug #12109

Exception during class load can cause partially-loaded class

02/25/2016 07:01 AM - arrtchiu (Matt U)

Status: Rejected	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.2.4p230 (2015-12-16 revision 53155) [x86_64-darwin15]	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN
Description I noticed this bug using Rails, which lazy-loads classes (via ActiveSupport.) Here's a minimal test case: <pre>#!/usr/bin/env ruby begin # simulate loading a class from file at runtime eval %q{ class Foo def self.bar fail 'bar' end bar def self.baz puts 'baz' end end } rescue => e puts "rescued: #{e}" end # => rescued: bar Foo.baz # => undefined method `baz' for Foo:Class (NoMethodError)</pre> EXPECTED Either: <ul style="list-style-type: none">• No exception - finish adding the method to the class, or• 'undefine' the class, at least making it obvious that the class didn't load completely (or in the case of rails/activesupport, allow it to be reloaded on-demand later)	

History

#1 - 03/15/2016 06:17 AM - shyouhei (Shyouhei Urabe)

- Description updated

You can write virtually anything inside a class definition. I don't think it's practically possible to revert all operations inside of a require (that needs a full ACID properties I guess).

Then on the other hand is it a wise idea to ignore exceptions? I don't think so. Because there should be a way for a class author to bail out the creation of a class on some reason. That is not a well-doing, but should there be a way to do so at least.

I heard that chef scripts intentionally raises from inside their definitions, to abnormally quit a provisioning. That sort of things sometimes happen.

#2 - 03/16/2016 03:29 PM - marcandre (Marc-Andre Lafortune)

- Status changed from Open to Rejected

Closing as this is not a bug.

I don't see any change of the sort happening as both of your expectations make no sense. Please understand that when loading a file, ruby executes them. The statements are executed in order. The def could be changed to `define_method` which is not a keyword but an instance method of Module.

Is it rails that rescues the exception when autoloading? I can't remember. If so, patch that behavior if you really want to.