

Ruby master - Bug #4403

require command loads file twice when required symlink or file under directory which is symlink (symbolic link)

02/16/2011 11:48 PM - rupert (Robert Pankowecki)

Status: Rejected	
Priority: Normal	
Assignee: nahi (Hiroshi Nakamura)	
Target version: 1.9.3	
ruby -v: ruby 1.9.2p136 (2010-12-25 revision 30365) [i686-linux]	Backport:
Description	
<pre>=begin Consider this file structure: ~/test/symbolic > tree . -- a -- a.rb -- b -> a -- c.rb -- d.rb -> c.rb Current behavior: ruby-1.9.2-p136 :001 > require './c' => true ruby-1.9.2-p136 :002 > require './d' ./d.rb:1: warning: already initialized constant CON2 => true ruby-1.9.2-p136 :003 > require './a/a' => true ruby-1.9.2-p136 :004 > require './b/a' ./b/a.rb:1: warning: already initialized constant CON => true Expected behavior: ruby-1.9.2-p136 :001 > require './c' => true ruby-1.9.2-p136 :002 > require './d' => false ruby-1.9.2-p136 :003 > require './a/a' => true ruby-1.9.2-p136 :004 > require './b/a' => false This bug can be reproduced at Ruby 1.8.7, too. =end</pre>	

History

#1 - 02/17/2011 03:24 AM - wyhaines (Kirk Haines)

=begin

On Wed, Feb 16, 2011 at 7:48 AM, Robert Pankowecki

redmine@ruby-lang.org wrote:

Bug [#4403](#): require command loads file twice when required symlink or file under directory which is symlink (symbolic link)
<http://redmine.ruby-lang.org/issues/show/4403>

Author: Robert Pankowecki

Status: Open, Priority: Normal

ruby -v: ruby 1.9.2p136 (2010-12-25 revision 30365) [i686-linux]

Consider this file structure:

```
~/test/symbolic > tree
.
|-- a
|  -- a.rb
|-- b -> a
|-- c.rb
-- d.rb -> c.rb
```

Current behavior:

```
ruby-1.9.2-p136 :001 > require './c'
=> true
ruby-1.9.2-p136 :002 > require './d'
./d.rb:1: warning: already initialized constant CON2
=> true
ruby-1.9.2-p136 :003 > require './a/a'
=> true
ruby-1.9.2-p136 :004 > require './b/a'
./b/a.rb:1: warning: already initialized constant CON
=> true
```

Expected behavior:

```
ruby-1.9.2-p136 :001 > require './c'
=> true
ruby-1.9.2-p136 :002 > require './d'
=> false
ruby-1.9.2-p136 :003 > require './a/a'
=> true
ruby-1.9.2-p136 :004 > require './b/a'
=> false
```

This bug can be reproduced at Ruby 1.8.7, too.

The behavior that you are seeing is what is expected. Ruby is using the path of the required file to differentiate between files which have been required before, and those which have not been.

This has been well documented in the code and elsewhere for a long time.

Take a look in eval.c, searching for "rb_f_require", in a 1.8.x series source code, and in the 1.9.x code, look in load.c. The comment/documentation is identical except for the last sentence, which details a single important difference:

Ruby 1.8.6:

[This first chunk is identical]

- Ruby tries to load the library named `string`, returning
- `+true+` if successful. If the filename does not resolve to
- an absolute path, it will be searched for in the directories listed
- in `$:`. If the file has the extension ``.rb"`, it is
- loaded as a source file; if the extension is `.so", ".o"`, or
- ``.dll"`, or whatever the default shared library extension is on
- the current platform, Ruby loads the shared library as a Ruby
- extension. Otherwise, Ruby tries adding `.rb", ".so"`, and so on
- to the name. The name of the loaded feature is added to the array in
- `$@`. A feature will not be loaded if its name already

[This last sentence is what is different]

- appears in `$@`. However, the file name is not converted
- to an absolute path, so that `require 'a'; require`
- `'./a'"` will load `a.rb` twice.

Ruby 1.9.3 trunk, as of today:

[Just the part that is different.]

- The file name is converted to an absolute
- path, so `require 'a'; require './a'"` will not load

- a.rb twice.

In both cases, though, a link, which creates two different absolute paths to the the same file, will allow it to get required twice. This is expected.

Kirk Haines

=end

#2 - 06/26/2011 06:39 PM - nahi (Hiroshi Nakamura)

- Status changed from Open to Assigned
- Assignee set to nahi (Hiroshi Nakamura)
- Target version set to 1.9.3

#3 - 06/28/2011 12:33 PM - nahi (Hiroshi Nakamura)

- Category set to core
- Status changed from Assigned to Rejected

As Kirk said. 1.9 expands a path for required feature for double load detection but it does not follow symlink. I can understand that some user think it as a bug but following symlink should be handled at OS layer, not at Ruby language layer I think. I don't expect 'require "foo"' adds '/path/to/bar.rb' to \$LOADED_FEATURES.

I close this ticket as 'Not a Bug'. Please re-file a new issue for require+symlink as 'Feature' if you think it good to be added.