

## Ruby trunk - Bug #10222

### require\_relative and require should be compatible with each other when symlinks are used

09/10/2014 12:53 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b> 2.3.1, 2.1.2p95	<b>Backport:</b> 2.3: REQUIRED, 2.4: DONE, 2.5: DONTNEED
<b>Description</b> <p>Not sure if this should be considered a bug or a feature request since I don't know whether the current behavior is intended or not.</p> <p>Recently I got a report for my gem rails-web-console related to require_relative causing trouble with symlinked dirs:</p> <p><a href="https://github.com/rosenfeld/active_record_migrations/issues/6">https://github.com/rosenfeld/active_record_migrations/issues/6</a></p> <p>Dmitry was able to replicate the issue using vanilla Ruby:</p> <pre>mkdir a ln -s a b echo "require_relative 'b'" &gt; a/a.rb echo "p 'b loaded'" &gt; a/b.rb echo "\$: &lt;&lt; File.expand_path('../b', __FILE__); require 'a'; require 'b'" &gt; c.rb ruby c.rb</pre> <p>Notice how "b loaded" is printed twice but if you replace require_relative with require it's just loaded once.</p> <p>Shouldn't Ruby always expand the loaded files before appending them to the \$LOADED_FEATURES and avoid this kind of error? I don't think require_relative should behave differently than a regular require in such cases.</p> <p>Any thoughts?</p>	
<b>Related issues:</b> <p>Related to Ruby trunk - Bug #14372: Memory leak in require with Pathnames in ... <b>Closed</b></p>	

#### Associated revisions

##### Revision b6d3927e - 09/21/2017 07:29 AM - nobu (Nobuyoshi Nakada)

load.c: real path to load

- load.c (rb\_construct\_expanded\_load\_path): expand load paths to real paths to get rid of duplicate loading from symbolic-linked directories. [Feature #10222]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@59984 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 59984 - 09/21/2017 07:29 AM - nobu (Nobuyoshi Nakada)

load.c: real path to load

- load.c (rb\_construct\_expanded\_load\_path): expand load paths to real paths to get rid of duplicate loading from symbolic-linked directories. [Feature #10222]

##### Revision 59984 - 09/21/2017 07:29 AM - nobu (Nobuyoshi Nakada)

load.c: real path to load

- load.c (rb\_construct\_expanded\_load\_path): expand load paths to real paths to get rid of duplicate loading from symbolic-linked directories. [Feature #10222]

##### Revision 59984 - 09/21/2017 07:29 AM - nobu (Nobuyoshi Nakada)

load.c: real path to load

- load.c (rb\_construct\_expanded\_load\_path): expand load paths to real paths to get rid of duplicate loading from symbolic-linked directories.

[Feature #10222]

#### Revision eaba9da1 - 02/16/2018 04:25 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 59983,59984: [Backport #10222] [Backport #14372] [Backport #14424]

```
file.c: rb_check_realpath
```

```
* file.c (rb_check_realpath): returns real path which has no
symbolic links. similar to rb_realpath except for returning
Qnil if any parts did not exist.
```

```
load.c: real path to load
```

```
* load.c (rb_construct_expanded_load_path): expand load paths to
real paths to get rid of duplicate loading from symbolic-linked
directories. [Feature #10222]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_4@62440 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 62440 - 02/16/2018 04:25 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 59983,59984: [Backport #10222] [Backport #14372] [Backport #14424]

```
file.c: rb_check_realpath
```

```
* file.c (rb_check_realpath): returns real path which has no
symbolic links. similar to rb_realpath except for returning
Qnil if any parts did not exist.
```

```
load.c: real path to load
```

```
* load.c (rb_construct_expanded_load_path): expand load paths to
real paths to get rid of duplicate loading from symbolic-linked
directories. [Feature #10222]
```

## History

---

### #1 - 09/10/2014 02:23 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

I can't change the title myself. Could someone with privileges please change it to something like: "require\_relative and require should be compatible with each other when symlinks are used".

I think this would make it easier to be searchable if others are experiencing the same issue. The key change is to add the "symlinks" word to the title so that the connection is made clear.

### #2 - 09/14/2014 09:56 AM - shevegen (Robert A. Heiler)

Curious behaviour indeed, there may be a reason why symlinks were assumed to behave differently.

### #3 - 08/19/2016 06:58 AM - abolshakov (Tema Bolshakov)

- ruby -v changed from 2.1.2p95 to 2.3.1

### #4 - 08/19/2016 06:59 AM - abolshakov (Tema Bolshakov)

- ruby -v changed from 2.3.1 to 2.3.1, 2.1.2p95

### #5 - 10/11/2016 11:07 AM - shyouhei (Shyouhei Urabe)

- Subject changed from require\_relative and require should be compatible with each other to require\_relative and require should be compatible with each other when symlinks are used

We looked at this issue in developer meeting today.

The ultimate reason why require and require\_relative behaves differently is that while require\_relative infers its argument's realpath every time, require doesn't.

This was by design; because require is called many times, we wanted to completely avoid disk access for 2nd and later calls to require with identical arguments.

But I believe the reported behaviour is a bug to be fixed. In order to do so a meeting attendee suggested to push both symlink-resolved and unresolved paths at once to \$LOADED\_FEATURES on the first call.

### #6 - 10/11/2016 01:19 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

This could make it harder for auto-reloaders to unload a required file when `require_relative` is used... Doesn't seem like a great solution to this bug to me... Ruby could cache internally the real path when using "require" so that the second call would avoid any disk access...

#### #7 - 10/11/2016 03:39 PM - nobu (Nobuyoshi Nakada)

Shyouhei Urabe wrote:

In order to do so a meeting attendee suggested to push both symlink-resolved and unresolved paths at once to `$LOADED_FEATURES` on the first call.

I think this explanation differs from that we discussed accurately. IIRC, it should be `vm->expanded_load_path`, not `$LOADED_FEATURES`. `$LOADED_FEATURES` won't be doubled, but will have realpaths only.

#### #8 - 10/11/2016 05:13 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Nobu, if a symlinked path would be removed from `$LOADED_FEATURES` from Ruby code, will all related internal references to the same file be cleared too?

#### #9 - 11/01/2016 06:44 AM - shyouhei (Shyouhei Urabe)

Today I learned that [PHP caches realpath](#), and causes troubles when people use symlink to deploy scripts.

The situation is not exactly the identical to ours, but we should avoid their footsteps.

#### #10 - 09/13/2017 02:45 PM - nobu (Nobuyoshi Nakada)

Would you expect just directory names get expanded?  
Or basename too?  
For instance, 'b loaded' should be printed twice or just once,?

```
mkdir a
echo "p 'b loaded'" > a/b.rb
ln -s b.rb a/c.rb
ruby -I./a -ra -rb -e''
```

#### #11 - 09/13/2017 03:27 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

I'd expect b.rb and c.rb to be handled like separate files, so "b loaded" would be printed twice.

#### #12 - 09/13/2017 03:29 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

I see, what you mean, if we simply add the expanded filenames to `LOADED_PATH` then that file would be loaded just once. Do you have any use cases where someone would symlink Ruby code for good usage?

#### #13 - 09/13/2017 11:35 PM - matsuda (Akira Matsuda)

Here's an actual use case that we saw in Rails: <https://github.com/rails/rails/pull/29638#issuecomment-321335175>  
The reporter says that it happened in Jenkins, but I guess the same situation may happen in any case where we put the .rb files under a symlinked directory, for instance Capistrano.

#### #14 - 09/13/2017 11:38 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Akira, in those cases was the basename different among the real path and the symlink?

#### #15 - 09/21/2017 12:26 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset [trunk|r59984](#).

---

load.c: real path to load

- load.c (`rb_construct_expanded_load_path`): expand load paths to real paths to get rid of duplicate loading from symbolic-linked directories. [Feature [#10222](#)]

#### #16 - 01/30/2018 08:27 AM - nobu (Nobuyoshi Nakada)

- Related to Bug #14372: Memory leak in require with Pathnames in the `$LOAD_PATH` in 2.3/2.4 added

#### #17 - 01/31/2018 04:24 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.3: REQUIRED, 2.4: REQUIRED, 2.5: DONTNEED

**#18 - 01/31/2018 04:26 AM - nagachika (Tomoyuki Chikanaga)**

I've filled Backport field according to the request at [#14424](#).

**#19 - 02/16/2018 04:25 PM - nagachika (Tomoyuki Chikanaga)**

- Backport changed from 2.3: *REQUIRED*, 2.4: *REQUIRED*, 2.5: *DONTNEED* to 2.3: *REQUIRED*, 2.4: *DONE*, 2.5: *DONTNEED*

ruby\_2\_4 r62440 merged revision(s) 59983,59984.