

## Ruby - Bug #4255

### When on a case-insensitive filesystem, "loaded features" search in require should ignore case

01/10/2011 04:45 PM - headius (Charles Nutter)

<b>Status:</b>	Rejected	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	nahi (Hiroshi Nakamura)	
<b>Target version:</b>		
<b>ruby -v:</b>	-	
<b>Backport:</b>		
<b>Description</b>		
=begin This should not re-load the same library when running on a system with case-insensitive filenames:  require 'blah' require 'BLAH'  The resource in question has been loaded, and should not be reloaded again, even if it is attempted with a differing case.  The above case seems contrived, but we have run into many real-world cases of this problem: <ul style="list-style-type: none"><li>• win32 APIs return paths with the drive name capitalized. Many libraries that manipulate load paths use lower-case drive names. (C: versus c:)</li><li>• User code that adds load path entries at command-line or in script can easily differ in case</li><li>• Typos in path names that introduce case differences will quietly work but cause double-requires. These may or may not be detected.</li></ul> A specific case in JRuby caused a series of files to be loaded twice, ultimately resulting in an alias chain looping back on itself (!!!). This is an extremely difficult case to debug, but very easy to reproduce.  I propose that Ruby 1.9 (and ideally Ruby 1.8, since I consider this a bug and not a feature) should treat "loaded features" "\$" entries sourced from a case-insensitive filesystem with a case-insensitive comparison. =end		

#### Associated revisions

Revision 27ddbc59 - 05/22/2011 12:52 PM - Hiroshi Nakamura

- include/ruby/defines.h (CASEFOLD\_FILESYSTEM): Revert r30508. See #4255.  
Now **APPLE** is not CASEFOLD\_FILESYSTEM again.
- load.c (loaded\_feature\_path, rb\_feature\_p, load\_lock): Revert r30508.  
See #4255. Make \$LOADED\_FEATURES scanning case-sensitive again.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@31692 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### History

#1 - 01/10/2011 10:47 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin  
This issue was solved with changeset r30508.  
Charles, thank you for reporting this issue.  
Your contribution to Ruby is greatly appreciated.  
May Ruby be with you.

=end

#2 - 01/10/2011 11:41 PM - Eregon (Benoit Daloze)

=begin

On 10 January 2011 14:47, Nobuyoshi Nakada [redmine@ruby-lang.org](mailto:redmine@ruby-lang.org) wrote:

This issue was solved with changeset r30508.

- include/ruby/defines.h (CASEFOLD\_FILESYSTEM): HFS+ is case insensitive.

That is not always true, unfortunately, you can have a HFS+ case sensitive partition.

However, it seems the system partition is formatted by default as case insensitive HFS+.

So, it would be best to check case sensitivity for HFS+ instead of assuming, if that is possible.

Also, it is not optimal to determine case sensitivity at compile time.

The ideal would be to be able check the case sensitivity of the partition where the concerned file is.

But I guess that is not easy and would impact too much the performance.

=end

### #3 - 01/11/2011 04:47 AM - headius (Charles Nutter)

=begin

Benoit is right, the issue is more complicated than this.

- OS X can have either case-sensitive or insensitive HFS+
- Any OS can mount remote drives (e.g. NFS or Windows sharing) that has different case-sensitivity than the install target filesystem.

We did the above change in JRuby, based on where JRuby is installed, but we have an open issue to make it address the fact that a Ruby instance may touch many filesystems with different case sensitivity.

=end

### #4 - 01/11/2011 05:06 AM - zenspider (Ryan Davis)

=begin

On Jan 10, 2011, at 06:41 , Benoit Daloze wrote:

On 10 January 2011 14:47, Nobuyoshi Nakada [redmine@ruby-lang.org](mailto:redmine@ruby-lang.org) wrote:

This issue was solved with changeset r30508.

- include/ruby/defines.h (CASEFOLD\_FILESYSTEM): HFS+ is case insensitive.

That is not always true, unfortunately, you can have a HFS+ case sensitive partition.

However, it seems the system partition is formatted by default as case insensitive HFS+.

So, it would be best to check case sensitivity for HFS+ instead of assuming, if that is possible.

Yup. I run case-sensitive HFS+ on both my root and backup disk, but interact with case-insensitive disks from time to time.

Also, it is not optimal to determine case sensitivity at compile time.

The ideal would be to be able check the case sensitivity of the partition where the concerned file is.

But I guess that is not easy and would impact too much the performance.

Agreed that you can't determine at compile time. It could switch from run to run... But I think it would be ok to check once per filesystem. But as soon as you do that you open up another set of cross-platform issues (determining mounts, etc).

=end

### #5 - 02/08/2011 01:38 AM - headius (Charles Nutter)

=begin

So is there going to be any more work on this issue? We still have <http://jira.codehaus.org/browse/JRUBY-5320> open because of the remaining issues. If ruby-core does not plan to fix them, we won't either.

=end

**#6 - 05/18/2011 04:56 PM - nahi (Hiroshi Nakamura)**

- Category set to core

- Status changed from Closed to Open

I think this issue involuntarily interpreted and "fixed".

For 1.9, I propose reverting r30508 before 1.9.3 (it's not yet released.)

The issue Charles stated is indeed the problem on 1.8 (I only checked 1.8.7, though).

```
C:\Users\nahi\Documents\tools\ruby-1.8.7-p302-i386-mswin32\bin>ruby -v
ruby 1.8.7 (2010-08-16 patchlevel 302) [i386-mswin32]
```

```
C:\Users\nahi\Documents\tools\ruby-1.8.7-p302-i386-mswin32\bin>copy con foo.rb
p :LOADED
^Z
```

```
C:\Users\nahi\Documents\tools\ruby-1.8.7-p302-i386-mswin32\bin>ruby -I. -e "require 'foo'; require 'Foo'; require 'FOO'; require 'foo.rb'; p $LOADED_FEATURES"
:LOADED
:LOADED
:LOADED
["enumerator.so", "foo.rb", "Foo.rb", "FOO.rb"]
```

It loads 3 times, not 1 time nor 4 times. (I know what's happening inside.)

But 1.9 does not have this problem since it pushes expanded path to \$LOADED\_FEATURES.

```
C:\Users\nahi\Documents\tools\ruby-1.9.2-p0-i386-mswin32\bin>ruby -v
ruby 1.9.2p0 (2010-08-18 revision 29036) [i386-mswin32]
```

```
C:\Users\nahi\Documents\tools\ruby-1.9.2-p0-i386-mswin32\bin>copy con foo.rb
p :LOADED
^Z
```

```
C:\Users\nahi\Documents\tools\ruby-1.9.2-p0-i386-mswin32\bin>ruby -I. -e "require 'foo'; require 'Foo'; require 'FOO'; require 'foo.rb'; p $LOADED_FEATURES"
:LOADED
["enumerator.so", "C:/Users/nahe/Documents/tools/ruby-1.9.2-p0-i386-mswin32/lib/ruby/1.9.1/i386-mswin32/enc/encdb.so", "C:/Users/nahe/Documents/tools/ruby-1.9.2-p0-i386-mswin32/lib/ruby/1.9.1/i386-mswin32/enc/shift_jis.so", "C:/Users/nahe/Documents/tools/ruby-1.9.2-p0-i386-mswin32/lib/ruby/1.9.1/i386-mswin32/enc/trans/transdb.so", "C:/Users/nahe/Documents/tools/ruby-1.9.2-p0-i386-mswin32/bin/foo.rb"]
```

It loads only 1 time. See 1.9.2 revision 29036 < r30508.

Even without r30508, 1.9 only loads 1 time even if user gives various case combinations of a file to 'require'.

Since the "fix" r30508 fixed the problem which did not exist, let's revert it.

We can imagine contrived example, such as 'if rubygems inserts crafted String to \$' by itself?' but we should "fix" it at that time, based on concrete examples of the problem.

Given no objection, I'm going to revert it in a week. (Say NO now! 1.9.3 is coming :-)

Aside from this, we might need this case-insensitivity fix for 1.8.  
Is it too late?

**#7 - 05/18/2011 05:20 PM - xaviershay (Xavier Shay)**

We can imagine contrived example, such as 'if rubygems inserts crafted String to \$' by itself?'

MRI ruby tests and rubyspec both rely on being able to change \$. Not sure how much of a circular justification that is.

I worry that backporting specific fixes that have already gone into trunk to 1.8 may introduce the same performance regressions we've seen in 1.9. Would it be worth instead trying to get my patch [1] over the line (which addresses this issue) and backporting that?

[1] <http://redmine.ruby-lang.org/issues/show/3924>

**#8 - 05/18/2011 05:43 PM - nahi (Hiroshi Nakamura)**

MRI ruby tests and rubyspec both rely on being able to change \$".  
Not sure how much of a circular justification that is.

Yes, it is. Changing \$" is OK. I meant that \$.replace(\$".map(&:upcase)) does not work on case-insensitive FS as well as case-sensitive FS but I don't think these contrived examples help us how to "fix" it.

For 1.8, when we need to introduce huge perf regressions, Shyouhei would not fix this.

**#9 - 05/18/2011 06:23 PM - usa (Usaku NAKAMURA)**

- ruby -v changed from Any version to -

Hello,

In message "[\[ruby-core:36307\]](#) [Ruby 1.9 - Bug [#4255](#)][Open] When on a case-insensitive filesystem, "loaded features" search in require should ignore case"  
on May.18,2011 16:57:35, [nakahiro@gmail.com](mailto:nakahiro@gmail.com) wrote:

Since the "fix" r30508 fixed the problem which did not exist, let's revert it.  
We can imagine contrived example, such as 'if rubygems inserts crafted String to \$" by itself?' but we should "fix" it at that time, based on concrete examples of the problem.

I perfectly agree with you.

r30508 is not a bugfix. It's a new spec/feature and introduced without the deliberation.

At present, I have no certainty opinion about this spec change.  
However, to change a spec, some clear purpose and reasons should be presented.

Given no objection, I'm going to revert it in a week. (Say NO now! 1.9.3 is coming :-)

Revert it now :)

If someone has objection, he/she should affix the reason and the purpose and create a new ticket as feature request.

Aside from this, we might need this case-insensitivity fix for 1.8.  
Is it too late?

There aren't any plan to release new 1.8...

**Regards,**

U.Nakamura [usa@garbagecollect.jp](mailto:usa@garbagecollect.jp)

**#10 - 05/20/2011 01:49 PM - headius (Charles Nutter)**

If 1.9 does indeed get the proper expanded path of the file, and also *uses* that expanded path for LOADED\_FEATURES, then I agree the case-insensitivity change is unnecessary.

**#11 - 05/23/2011 11:00 AM - nahi (Hiroshi Nakamura)**

- Status changed from Open to Rejected

- Assignee set to nahi (Hiroshi Nakamura)

Thanks, guys. I reverted r30508 at r31692 and r31712 (Separated just by my commit error.)

I close this ticket as 'Rejected' for 1.9, but as I stated above, hand-weaving of \$LOADED\_FEATURES can cause a similar problem. And we know that 1.8.7 still has this problem as the reporter described, though it's too big change to call 'bug fix'...