

## Ruby master - Bug #14570

### Wired behavior of File.expand\_path() on Windows

03/02/2018 09:31 PM - larskanis (Lars Kanis)

<b>Status:</b>	Open	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>		
<b>ruby -v:</b>	ruby 2.6.0dev (2018-02-22 trunk 62518) [x64-mingw32]	<b>Backport:</b> 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

#### Description

File.realpath returns the path with the same upper/lower case characters as the input parameter. But File.expand\_path is different. It adjusts the last part of the path (and only this) to the upper/lower case characters given from the file system.

```
require "fileutils"
FileUtils.mkdir_p("TestDir/abC/Xyz")

File.expand_path("testdir") # => "C:/Users/lars/TestDir"
File.expand_path("testdir/abc") # => "C:/Users/lars/testdir/abC"
File.expand_path("testdir/abc/XYZ") # => "C:/Users/lars/testdir/abc/Xyz"

File.realpath("testdir") # => "C:/Users/lars/testdir"
File.realpath("testdir/abc") # => "C:/Users/lars/testdir/abc"
File.realpath("testdir/abc/XYZ") # => "C:/Users/lars/testdir/abc/XYZ"
```

The file system on Windows is case insensitive. So this is not really a bug, but it's inconsistent and unexpected. I would expect that File.expand\_path returns a path with upper/lowercase either entirely retrieved from the input parameter or from the file system, but not a mixture of both. I would also expect that File.realpath and File.expand\_path have the same semantics in this regard.

Same behavior on ruby 2.2.2p95 (2015-04-13 revision 50295) [x64-mingw32]

#### History

##### #1 - 03/03/2018 12:13 AM - graywolf (Gray Wolf)

If the behaviour is adjusted to reflect the filesystem, it would be nice to do the same for `__FILE__` (and I assume also `__dir__`). At the moment `__FILE__` doesn't necessarily match file system casing. Again, same as in the post above, since windows are case insensitive it's not technically a bug, but if File.realpath (expand\_path) are changed to reflect real casing, `__FILE__` should probably also be changed.