

## Ruby trunk - Feature #13726

### PATCH: Windows builds - fractional second file times

07/07/2017 02:27 AM - MSP-Greg (Greg L)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	usa (Usaku NAKAMURA)	
<b>Target version:</b>		
<b>Description</b>		
<p>At present, I believe Windows builds have integer seconds resolution on File times.</p> <p>Attached are three patches that seem to correct the issue. The numbered patches are best applied in order.</p> <p>The c source patches were created by kubo (Kubo Takehiro); I created the spec patch, along with some testing.</p> <p>All related tests (run parallel) pass on my system for MinGW builds and x64-mswin64_140 builds. I did some checks with Appveyor, and there may be issues with the logger tests, due to the fact that the tests are very time sensitive (I ran -j3 on a quad system, Appveyor runs -j2 on a dual core).</p> <p>I believe file time values are stored with 100 ns resolution, but times have 100 <math>\mu</math>s resolution. This was the reason for some of the changes to the tests, as time appears to be truncated (floor), as opposed to round.</p> <p>I would appreciate if consideration could be given to adding this (or similar) code. Thank you.</p>		
<b>Related issues:</b>		
Related to Ruby trunk - Misc #13702: TZ checking in rb_w32_fstati64()		<b>Closed</b>
Related to Ruby trunk - Feature #13731: inode for Windows on ReFS		<b>Closed</b>

#### Associated revisions

##### Revision ce7c1c05 - 12/04/2017 10:48 AM - usa (Usaku NAKAMURA)

support nanosec file timestamp on newer Windows

Support nanosec file timestamp on Windows 8 or later.  
Original patches are written by kubo (Kubo Takehiro).  
Windows 7 and earlier also supports nanosec file timestamp, but it's too accurate than system time. so, this feature is disabled on such versions.  
[Feature #13726]

this change also includes [Misc #13702]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@61013 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 61013 - 12/04/2017 10:48 AM - usa (Usaku NAKAMURA)

support nanosec file timestamp on newer Windows

Support nanosec file timestamp on Windows 8 or later.  
Original patches are written by kubo (Kubo Takehiro).  
Windows 7 and earlier also supports nanosec file timestamp, but it's too accurate than system time. so, this feature is disabled on such versions.  
[Feature #13726]

this change also includes [Misc #13702]

##### Revision 61013 - 12/04/2017 10:48 AM - usa (Usaku NAKAMURA)

support nanosec file timestamp on newer Windows

Support nanosec file timestamp on Windows 8 or later.  
Original patches are written by kubo (Kubo Takehiro).  
Windows 7 and earlier also supports nanosec file timestamp, but it's too accurate than system time. so, this feature is disabled on such versions.  
[Feature #13726]

this change also includes [Misc #13702]

**Revision 61013 - 12/04/2017 10:48 AM - usa (Usaku NAKAMURA)**

support nanosec file timestamp on newer Windows

Support nanosec file timestamp on Windows 8 or later.  
Original patches are written by kubo (Kubo Takehiro).  
Windows 7 and earlier also supports nanosec file timestamp, but it's too accurate than system time. so, this feature is disabled on such versions.  
[Feature #13726]

this change also includes [Misc #13702]

**Revision f1728527 - 12/10/2017 11:28 PM - usa (Usaku NAKAMURA)**

mention about [Feature #13726] and [Feature #13731]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@61100 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 61100 - 12/10/2017 11:28 PM - usa (Usaku NAKAMURA)**

mention about [Feature #13726] and [Feature #13731]

**Revision 61100 - 12/10/2017 11:28 PM - usa (Usaku NAKAMURA)**

mention about [Feature #13726] and [Feature #13731]

**Revision 61100 - 12/10/2017 11:28 PM - usa (Usaku NAKAMURA)**

mention about [Feature #13726] and [Feature #13731]

**History**

---

**#1 - 07/07/2017 02:12 PM - MSP-Greg (Greg L)**

- *Subject changed from Windows builds - fractional second file times to PATCH: Windows builds - fractional second file times*

**#2 - 07/07/2017 02:35 PM - MSP-Greg (Greg L)**

I was mistaken about Time.now resolution in windows.

What I believe does occur is that when windows natively writes file times, it seems to write them with 100  $\mu$ s resolution, and truncated.

But, Ruby can write with higher resolution using utime (after the patch).

**#3 - 07/08/2017 05:47 AM - kubo (Takehiro Kubo)**

01-time-subsec.patch changes declarations in include/ruby/win32.h.  
IMO, it could be in ruby 2.5.x but could not be backported to ruby 2.4.x because of ABI incompatibility.  
It also deletes meaningless TZ checking I pointed at [#13702](#).

**#4 - 07/11/2017 02:43 PM - MSP-Greg (Greg L)**

- *File spec-rubyspec-core-file-time.patch added*

While running tests today on ruby 2.5.0dev (2017-07-11 trunk 59311) [x64-mingw32], I had an error in [spec/rubyspec/core/file/utime\\_spec](#) due to the truncation issue. So, I've modified the test code.

When looking at this issue, please replace the patch in the first message with the attached one.

**#5 - 07/18/2017 12:37 PM - usa (Usaku NAKAMURA)**

- *Assignee set to usa (Usaku NAKAMURA)*  
- *Status changed from Open to Assigned*

**#6 - 07/18/2017 12:38 PM - usa (Usaku NAKAMURA)**

- *Related to Misc #13702: TZ checking in rb\_w32\_fstati64() added*

**#7 - 07/18/2017 12:38 PM - usa (Usaku NAKAMURA)**

- *Related to Feature #13731: inode for Windows on ReFS added*

**#8 - 12/03/2017 03:05 PM - usa (Usaku NAKAMURA)**

- *Backport deleted (2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN)*

- ruby -v deleted (trunk)

- Tracker changed from Bug to Feature

**#9 - 12/03/2017 04:14 PM - usa (Usaku NAKAMURA)**

I've tested the patches, and I found that my Windows7 box often set the ctime and mtime as a little future -- some hundreds nano seconds. I can't find the reason, but it makes some troubles with test-all and test-spec...

**#10 - 12/03/2017 04:29 PM - MSP-Greg (Greg L)**

usa (Usaku NAKAMURA) wrote:

I've tested the patches, and I found that my Windows7 box often set the ctime and mtime as a little future -- some hundreds nano seconds. I can't find the reason, but it makes some troubles with test-all and test-spec...

Thanks for looking at it. I'll look over the patches I'm using in ruby-loco, as all the time tests/specs have been stable for quite a while, both locally and on Appveyor. I believe all the time patches are in the [patches/gte20500](#) folder. Disk IO on AV gets a little funny when running parallel tests/specs, so I did need to increase (or add) some deltas in any numeric comparison tests...

Thanks, Greg

EDIT: Sorry, I should have been clearer, I may have increased some of the deltas in the current test/spec patches in ruby-loco to allow for reliable parallel testing on Appveyor.

**#11 - 12/04/2017 07:21 AM - larskanis (Lars Kanis)**

[usa \(Usaku NAKAMURA\)](#) I think the reason is, that precise file times are working as expected on Windows 7, but Time.now has a precision of 1/64 seconds only. This is because Windows 7 doesn't support GetSystemTimePreciseAsFileTime(), which is required to retrieve precise Time.now: <https://github.com/ruby/ruby/blob/b7de978e450779103dfe62084ce87757c5acae7d/win32/win32.c#L4604>

So to remove these inconsistencies we could truncate file times to 1/64 precision, when GetSystemTimePreciseAsFileTime() is not available?

**#12 - 12/04/2017 08:36 AM - usa (Usaku NAKAMURA)**

[larskanis \(Lars Kanis\)](#), thank you for information.

larskanis (Lars Kanis) wrote:

So to remove these inconsistencies we could truncate file times to 1/64 precision, when GetSystemTimePreciseAsFileTime() is not available?

OK, I'll test it.

**#13 - 12/04/2017 10:48 AM - usa (Usaku NAKAMURA)**

- Status changed from Assigned to Closed

Applied in changeset [trunk|r61013](#).

---

support nanosec file timestamp on newer Windows

Support nanosec file timestamp on Windows 8 or later. Original patches are written by kubo (Kubo Takehiro). Windows 7 and earlier also supports nanosec file timestamp, but it's too accurate than system time. so, this feature is disabled on such versions. [Feature [#13726](#)]

this change also includes [Misc [#13702](#)]

**#14 - 12/06/2017 10:01 AM - kubo (Takehiro Kubo)**

[usa \(Usaku NAKAMURA\)](#) Thanks for merging the patches.

I'm afraid that File.mtime(filename).usec returns always zero just after ruby.exe starts up on Windows 8 or later.

have\_precisetime is -1 at process startup. <https://github.com/ruby/ruby/blob/5a8a270/win32/win32.c#L4574>

It is changed to zero or one in get\_systemtime(). <https://github.com/ruby/ruby/blob/5a8a270/win32/win32.c#L4586-L4589>

If File.mtime is called before get\_systemime(), the fractional second is always zero even on Windows 8 or later.

<https://github.com/ruby/ruby/blob/5a8a270/win32/win32.c#L5493>

I cannot check it by myself because I failed to compile ruby from SVN source.  
Could somebody check it?

**#15 - 12/06/2017 11:09 AM - larskanis (Lars Kanis)**

[kubo \(Takehiro Kubo\)](#) I had the same concern, but tried the latest [RubyInstaller snapshot](#). It proves that `get_systemtime()` is called somewhere as part of the initialization.

```
$ ruby --disable-gems -e "p File.mtime('README').usec"
657156
$ ruby -v
ruby 2.5.0dev (2017-12-05 trunk 61034) [x64-mingw32]
```

**#16 - 12/06/2017 11:38 AM - kubo (Takehiro Kubo)**

[larskanis \(Lars Kanis\)](#) Thank you! I worried needlessly.

**Files**

---

02-file-utime.patch	5.53 KB	07/07/2017	MSP-Greg (Greg L)
spec-rubyspec-core-file-time.patch	2.99 KB	07/07/2017	MSP-Greg (Greg L)
01-time-subsec.patch	15.2 KB	07/07/2017	MSP-Greg (Greg L)
spec-rubyspec-core-file-time.patch	3.28 KB	07/11/2017	MSP-Greg (Greg L)