

## Ruby trunk - Feature #13731

### inode for Windows on ReFS

07/08/2017 06:15 AM - kubo (Takehiro Kubo)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	usa (Usaku NAKAMURA)
<b>Target version:</b>	
<b>Description</b>	
<p><a href="#">#11216</a> Windows inode BY_HANDLE_FILE_INFORMATION FileIndexHigh/Low ReFS ReFS 64-bit 128-bit</p> <p>In <a href="https://msdn.microsoft.com/en-us/library/windows/desktop/aa363788(v=vs.85).aspx">https://msdn.microsoft.com/en-us/library/windows/desktop/aa363788(v=vs.85).aspx</a> The ReFS file system, introduced with Windows Server 2012, includes 128-bit file identifiers. To retrieve the 128-bit file identifier use the GetFileInformationByHandleEx function with FileIdInfo to retrieve the FILE_ID_INFO structure. The 64-bit identifier in this structure is not guaranteed to be unique on ReFS.</p> <pre>128-bit struct stati64 #13726 stat st_ino 128-bit (st_ino 64-bit) ruby 2.5 ABI</pre>	
<b>Related issues:</b>	
Related to Ruby trunk - Feature #13726: PATCH: Windows builds - fractional se...	<b>Closed</b>
Related to Ruby trunk - Misc #13702: TZ checking in rb_w32_fstati64()	<b>Closed</b>

#### Associated revisions

##### Revision 8e7560bb - 12/04/2017 03:33 AM - usa (Usaku NAKAMURA)

support File.identical? on ReFS

- file.c (rb\_file\_identical\_p): move Windows dependent code to win32/win32.c.
- win32/win32.c (rb\_w32\_file\_identical\_p): support ReFS.  
see [Feature #13731] [ruby-dev:50166]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@61005 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 61005 - 12/04/2017 03:33 AM - usa (Usaku NAKAMURA)

support File.identical? on ReFS

- file.c (rb\_file\_identical\_p): move Windows dependent code to win32/win32.c.
- win32/win32.c (rb\_w32\_file\_identical\_p): support ReFS.  
see [Feature #13731] [ruby-dev:50166]

##### Revision 61005 - 12/04/2017 03:33 AM - usa (Usaku NAKAMURA)

support File.identical? on ReFS

- file.c (rb\_file\_identical\_p): move Windows dependent code to win32/win32.c.
- win32/win32.c (rb\_w32\_file\_identical\_p): support ReFS.  
see [Feature #13731] [ruby-dev:50166]

##### Revision 61005 - 12/04/2017 03:33 AM - usa (Usaku NAKAMURA)

support File.identical? on ReFS

- file.c (rb\_file\_identical\_p): move Windows dependent code to win32/win32.c.
- win32/win32.c (rb\_w32\_file\_identical\_p): support ReFS.  
see [Feature #13731] [ruby-dev:50166]

#### Revision 91de3d65 - 12/10/2017 05:26 PM - usa (Usaku NAKAMURA)

support 128bit ino on Windows (if available)

- win32/win32.c, include/ruby/win32.h (stati128, rb\_{,u,l,ul}stati128): rename from stati64ns, change the type of st\_ino to 64bit and added st\_inohigh.
- dir.c, file.c (stat, lstat): follow above changes.
- file.c (rb\_stat\_ino): support 128bit ino.
- win32/win32.c (rb\_{,u,l,ul}stati128): ditto.  
[Feature #13731]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@61096 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 61096 - 12/10/2017 05:26 PM - usa (Usaku NAKAMURA)

support 128bit ino on Windows (if available)

- win32/win32.c, include/ruby/win32.h (stati128, rb\_{,u,l,ul}stati128): rename from stati64ns, change the type of st\_ino to 64bit and added st\_inohigh.
- dir.c, file.c (stat, lstat): follow above changes.
- file.c (rb\_stat\_ino): support 128bit ino.
- win32/win32.c (rb\_{,u,l,ul}stati128): ditto.  
[Feature #13731]

#### Revision 61096 - 12/10/2017 05:26 PM - usa (Usaku NAKAMURA)

support 128bit ino on Windows (if available)

- win32/win32.c, include/ruby/win32.h (stati128, rb\_{,u,l,ul}stati128): rename from stati64ns, change the type of st\_ino to 64bit and added st\_inohigh.
- dir.c, file.c (stat, lstat): follow above changes.
- file.c (rb\_stat\_ino): support 128bit ino.
- win32/win32.c (rb\_{,u,l,ul}stati128): ditto.  
[Feature #13731]

#### Revision 61096 - 12/10/2017 05:26 PM - usa (Usaku NAKAMURA)

support 128bit ino on Windows (if available)

- win32/win32.c, include/ruby/win32.h (stati128, rb\_{,u,l,ul}stati128): rename from stati64ns, change the type of st\_ino to 64bit and added st\_inohigh.
- dir.c, file.c (stat, lstat): follow above changes.
- file.c (rb\_stat\_ino): support 128bit ino.
- win32/win32.c (rb\_{,u,l,ul}stati128): ditto.  
[Feature #13731]

#### 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 - 07/14/2017 09:01 AM - naruse (Yui NARUSE)**

- Backport deleted (2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN)
- ruby -v deleted (trunk)
- Assignee set to usa (Usaku NAKAMURA)
- Status changed from Open to Assigned
- Tracker changed from Bug to Feature

**#2 - 07/18/2017 12:36 PM - usa (Usaku NAKAMURA)**

2.5

**#3 - 07/18/2017 12:38 PM - usa (Usaku NAKAMURA)**

- Related to Feature #13726: PATCH: Windows builds - fractional second file times added

**#4 - 07/18/2017 12:38 PM - usa (Usaku NAKAMURA)**

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

**#5 - 09/26/2017 05:12 PM - naruse (Yui NARUSE)**

- Target version set to 2.5

**#6 - 12/04/2017 03:33 AM - usa (Usaku NAKAMURA)**

- Status changed from Assigned to Closed

Applied in changeset [trunk|r61005](#).

---

support File.identical? on ReFS

- file.c (rb\_file\_identical\_p): move Windows dependent code to win32/win32.c.
- win32/win32.c (rb\_w32\_file\_identical\_p): support ReFS. see [Feature #13731] [ruby-dev:50166]

**#7 - 12/04/2017 03:36 AM - usa (Usaku NAKAMURA)**

- Status changed from Closed to Assigned

stat [Feature #13726] File.identical? 2.5

**#8 - 12/04/2017 08:55 AM - naruse (Yui NARUSE)**

- Target version deleted (2.5)

**#9 - 12/07/2017 11:37 AM - kubo (Takehiro Kubo)**

2.5

ReFS FILE\_ID\_INFO 128 nFileIndexHigh/Low 64

FILE\_ID\_INFO 128 (32)

- 0x00000000\_00000001\_00000000\_00000600
- 0x00000000\_00000002\_00000000\_00000600

nFileIndexHigh/Low

- 0x30000000\_00001600
- 0x30000000\_00002600

128 xor 64 64

#10 - 12/10/2017 05:27 PM - usa (Usaku NAKAMURA)

- Status changed from Assigned to Closed

Applied in changeset [trunk|r61096](#).

support 128bit ino on Windows (if available)

- win32/win32.c, include/ruby/win32.h (stati128, rb\_{u,l,ul}stati128): rename from stati64ns, change the type of st\_ino to 64bit and added st\_inohigh.
- dir.c, file.c (stat, lstat): follow above changes.
- file.c (rb\_stat\_ino): support 128bit ino.
- win32/win32.c (rb\_{u,l,ul}stati128): ditto. [Feature #13731]

#11 - 12/10/2017 11:24 PM - usa (Usaku NAKAMURA)

ReFS

#12 - 12/11/2017 11:26 AM - kubo (Takehiro Kubo)

ReFS

stati128\_handle()

```
st->st_ino = *((unsigned __int64 *)&fii);
st->st_inohigh = *((__int64 *)&fii + 1);
```

&fii &ffi.FileId

#13 - 12/11/2017 05:32 PM - usa (Usaku NAKAMURA)

kubo (Takehiro Kubo) wrote:

stati128\_handle()

```
st->st_ino = *((unsigned __int64 *)&fii);
st->st_inohigh = *((__int64 *)&fii + 1);
```

&fii &ffi.FileId

r61121

#14 - 12/11/2017 07:22 PM - MSP-Greg (Greg L)

Usaka (and Kubo),

Thank you, with 61121 ruby-loco MinGW build is passing. I'll see what happens with a local Win7 build later on.

You're embarrassed? Don't be. How about me, as all I can do re c source issues is say 'something broke'...

Thanks again, Greg

**#15 - 12/12/2017 11:16 AM - kubo (Takehiro Kubo)**

ReFS NTFS File.stat(filename).ino GetFileInformationByHandleEx() FILE\_ID\_128 warning

```
compiling win32/win32.c
win32/win32.c: In function 'get_ino':
win32/win32.c:5445:11: warning: unused variable 'err' [-Wunused-variable]
    DWORD err;
    ^~~
```

**#16 - 12/12/2017 11:58 AM - usa (Usaku NAKAMURA)**