

Ruby - Feature #7148

Improved Tempfile w/o DelegateClass

10/12/2012 02:04 PM - Glass_saga (Masaki Matsushita)

Status:	Assigned	
Priority:	Normal	
Assignee:	Glass_saga (Masaki Matsushita)	
Target version:		
Description I propose improved Tempfile without DelegateClass(). Present Tempfile has following problems. 1. confusing inspect <pre>t = Tempfile.new("foo") #=> #<File:/tmp/foo20121012-6762-12w11to> t.is_a? File #=> false</pre> 2. #dup doesn't duplicate IO <pre>t = Tempfile.new("foo") t.dup.close t.read #=> IOError: closed stream</pre> 3. finalizer performs unlink even when it has been duplicated <pre>t = Tempfile.new("foo") path = t.path #=> "/tmp/foo20121012-7533-1q537gq" File.exist? path #=> true tt = t.dup t = nil GC.start File.exist? path #=> false</pre> I think these problems caused by using DelegateClass(). Therefore, I made a patch to resolve the problems. The patched Tempfile class is a subclass of File.		
Related issues: Related to Ruby - Bug #17144: Tempfile.open { ... } does not unlink the file		
		Closed

History

#1 - 10/12/2012 02:23 PM - Anonymous

Hi,

In message "Re: [\[ruby-core:47930\]](#) [ruby-trunk - Feature #7148][Open] Improved Tempfile w/o DelegateClass" on Fri, 12 Oct 2012 14:04:08 +0900, "Glass_saga (Masaki Matsushita)" glass.saga@gmail.com writes:

I propose improved Tempfile without DelegateClass().
Present Tempfile has following problems.

1. confusing inspect
2. #dup doesn't duplicate IO
3. finalizer performs unlink even when it has been duplicated

I have no objection about (1), but what we expect when we call Tempfile#dup might differ, for example, I'd expect it to copy the underlying temporary file. So making Tempfile a subclass of File may not be the ideal solution.

matz.

#2 - 10/16/2012 09:49 AM - Glass_saga (Masaki Matsushita)

Hello,

Yukihiro Matsumoto wrote:

I'd expect it to copy the underlying temporary file.

Is the behavior of #dup you expect like the following?

```
def dup
  dupe = self.class.new(@basename)
  IO.copy_stream(self, dupe, 0)
  dupe
end
```

I think the reason why Tempfile uses DelegateClass is that to implement Tempfile#open without it used to be difficult. To implement it as subclass of File, self must be reopened with full configuration, mode and opts. IO#reopen used not to accept them, but now it accepts after r37071.

#3 - 10/18/2012 02:35 AM - headius (Charles Nutter)

JRuby has been running with Tempfile < File for a couple years now, and have received only minor bug reports about it. It works very well, and has no delegation cost.

#4 - 11/16/2012 02:58 PM - Glass_saga (Masaki Matsushita)

Are there some reasons not to make Tempfile a subclass of File?
I think it's a better solution, even if it's not an ideal solution.

#5 - 11/24/2012 10:49 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned
- Assignee set to matz (Yukihiro Matsumoto)
- Priority changed from Normal to 3
- Target version set to 2.6

#6 - 05/27/2015 09:38 PM - normalperson (Eric Wong)

"Glass_saga (Masaki Matsushita)" glass.saga@gmail.com wrote:

Feature [#7148](#): Improved Tempfile w/o DelegateClass
<https://bugs.ruby-lang.org/issues/7148>

I would still like this for 2.3.0, just hit a snag with IO.copy_stream using Tempfile :x

Also, Charles hit a similar problem not long ago, too:
[\[ruby-core:68700\]](#) [Bug [#11015](#)]

#7 - 05/29/2015 01:21 AM - Glass_saga (Masaki Matsushita)

I think the problem you faced was resolved at r50118, wasn't it?

#8 - 05/29/2015 01:28 AM - normalperson (Eric Wong)

Sadly, r50118 actually caused my problem because I was using src_offset for IO.copy_stream

#9 - 05/29/2015 01:58 PM - headius (Charles Nutter)

It doesn't sound like anyone opposes this idea, so are we just missing implementation?

#10 - 06/12/2015 05:37 AM - nobu (Nobuyoshi Nakada)

- Description updated

#11 - 06/12/2015 06:12 AM - matz (Yukihiro Matsumoto)

It's OK for me to implement Tempfile without using DelegateClass. If JRuby does similar thing already, it might be a good idea to share implementation. Besides that, we might need to think about killing/redefining some unnecessary/invalid methods of File class, e.g. reopen. I am afraid dup is one of them.

Matz.

#12 - 06/15/2015 01:22 AM - Glass_saga (Masaki Matsushita)

- Assignee changed from matz (Yukihiro Matsumoto) to Glass_saga (Masaki Matsushita)

#13 - 03/27/2019 09:51 AM - sowieso (So Wieso)

I guess this is related:

```
tf = Tempfile.new
tf.unlink
tf2 = tf.dup
puts tf.fileno
# => 10
puts tf2.fileno
# => 11
tf2.close

print tf2.fileno
# => 11
print tf.fileno
IOError: closed stream
from /usr/local/rvm/rubies/ruby-2.6.0/lib/ruby/2.6.0/delegate.rb:349:in `fileno'
```

It closes the wrong fd. Using normal File instead of Tmpfile makes it work like usual.

Libraries that accept an IO object and duplicate it internally don't work correctly with Tempfile (in my case rubyzip). This really needs a fix.

#14 - 09/03/2020 03:05 PM - Glass_saga (Masaki Matsushita)

- Related to Bug #17144: Tempfile.open { ... } does not unlink the file added

Files

patch.diff	3.52 KB	10/12/2012	Glass_saga (Masaki Matsushita)
------------	---------	------------	--------------------------------