

Ruby - Bug #12257

io/wait: wait_readable/writable working different than IO.select when descriptor not open

04/07/2016 12:49 PM - chucke (Tiago Cardoso)

Status:	Rejected	Backport: 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:		
Description		
Coming from this pending issue: https://github.com/net-ssh/net-ssh/pull/303 Gist of it: <pre>f = IO.popen("/bin/sh", "r+") f.close_write f.wait_writable #=> IOError: not opened for writing IO.select(nil, [f], nil, 10) #=> no error</pre> The way I see it, it should have the same behaviour, i.e. either IO.select should complain, or wait_writable should just return nil.		

History

#1 - 06/26/2019 08:12 PM - jeremyevans0 (Jeremy Evans)

I don't think this is a bug. It makes sense for IO#wait_writable to raise an exception, because the specific IO instance will never be writable after close_write. As the documentation for IO.select indicates, it calls select(2), and returns arrays showing which IO objects you can take action on. I don't think that it would make sense for IO.select to go through all input or output read and write arrays and check if the file descriptors are still open for the appropriate direction. Even if it could, raising in that case would not be helpful, as you would lose information on which other file descriptors you could take action on.

One thing that may be helpful would be the addition of IO#read_closed? and IO#write_closed?, as currently I'm not sure how you can determine if an IO instance is closed in a given direction. However, that would be a feature request, not a bug fix.

#2 - 07/24/2019 07:02 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Rejected