

Ruby - Bug #13154

URI::Generic, == operator, normalize!

01/24/2017 11:45 AM - deman_killer (Dmitry Sk)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:		Backport: 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN
Description This two URI means the same: <pre>uri1 = URI::parse '/1/2/3' uri2 = URI::parse 'file:///1/2/3'</pre> but uri1 == uri2 return false it is cause of normalize! implementation that compare scheme this way: <pre>if scheme && scheme != scheme.downcase set_scheme(self.scheme.downcase) end</pre> But nil scheme actually equivalent to file		

History

#1 - 01/24/2017 12:11 PM - phluid61 (Matthew Kerwin)

Dmitry Sk wrote:

But nil scheme actually equivalent to file

Are you sure? I thought a URI::Generic object with a nil scheme is a [relative reference](#). See: [URI::Generic#merge](#) and [#relative?](#)

Related: <https://github.com/phluid61/file-uri>

#2 - 07/19/2021 07:53 AM - hsbt (Hiroshi SHIBATA)

- Tracker changed from Misc to Bug

- Description updated

- Backport set to 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN

#3 - 07/23/2021 07:15 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Rejected

I agree with @phluid61, plain path provided to URI.parse should not be treated as implicit file scheme. Such a change would break backwards compatibility for the methods mentioned, and I think URI.parse('http://foo/bar').merge(URI.parse('/1/2/3')) should result in #<URI::HTTP http://foo/1/2/3> and not #<URI::File file:///1/2/3>.

If you disagree, please submit an issue to <https://github.com/ruby/uri/issues> and @akira (akira yamada) can decide if the behavior should be changed.