

Ruby master - Feature #7360

Adding Pathname#glob

11/15/2012 10:26 PM - aef (Alexander E. Fischer)

Status:	Closed	
Priority:	Normal	
Assignee:	akr (Akira Tanaka)	
Target version:		
Description		
<p>Currently there is only a Pathname.glob method, which allows you to find Pathname objects by a pattern including wildcard characters like '*'. I would like to be able to use this relative to a current Pathname. some_directory = Pathname.new('some_directory') Pathname.glob(some_directory + 'a*') # all children starting with "a" could then simply be: some_directory.glob('a*') # all children starting with "a" If you like the idea, please let me know. I will provide a patch then.</p>		
Related issues:		
Related to Ruby master - Feature #7378: Adding Pathname#write	Closed	06/02/2012
Blocked by Ruby master - Feature #13056: base option to Dir.glob	Closed	

Associated revisions

Revision cea5f5aa - 10/21/2017 08:34 AM - akr (Akira Tanaka)

Pathname#glob method implemented.

[ruby-core:49373] [Feature #7360] proposed by Alexander E. Fischer.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60238 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 60238 - 10/21/2017 08:34 AM - akr (Akira Tanaka)

Pathname#glob method implemented.

[ruby-core:49373] [Feature #7360] proposed by Alexander E. Fischer.

Revision 60238 - 10/21/2017 08:34 AM - akr (Akira Tanaka)

Pathname#glob method implemented.

[ruby-core:49373] [Feature #7360] proposed by Alexander E. Fischer.

Revision 60238 - 10/21/2017 08:34 AM - akr (Akira Tanaka)

Pathname#glob method implemented.

[ruby-core:49373] [Feature #7360] proposed by Alexander E. Fischer.

History

#1 - 11/24/2012 10:45 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to akr (Akira Tanaka)

- Target version set to 2.6

#2 - 09/26/2015 04:57 PM - til (Tilman Singer)

- File 19c4ae36aeea65b30e8b73c89f7ac42bd6905ed7.patch added

Here is a patch that adds the functionality as outlined by Alexander. It's using "Pathname#join" to build the glob pattern, and then delegating to "Pathname.glob".

As described in the rdoc, it will accept a single string with a glob pattern, or an array of glob patterns.

The main use case is when you want to do something based relative to a current directory that is already available as Pathname, such as Rails.root in a Rails project. With this, you can say e.g.

```
Rails.root.glob("**.rb")
```

to find all ruby files in the current project.

Link to the branch of my fork on GitHub:

<https://github.com/til/ruby/tree/add-pathname-glob-instance-method>

#3 - 10/31/2016 06:41 AM - sakuro (Sakuro OZAWA)

ping?

#4 - 10/31/2016 08:39 AM - til (Tilman Singer)

The patch still applies to latest trunk.

#5 - 11/05/2016 02:00 PM - akr (Akira Tanaka)

- Status changed from Assigned to Feedback

It seems the proposed implementation treats the receiver as a glob pattern.

```
% ruby -rpathname -e '
class Pathname
  def glob(pattern, flags = 0, &b)
    Array(pattern).flat_map do |pat|
      self.class.glob(join(pat), flags, &b)
    end
  end
end
p Pathname("/b*").glob("i*")
'
```

[#<Pathname:/boot/initrd.img-3.16.0-4-amd64>, #<Pathname:/bin/ip>]

I think it is not intentional.

#6 - 12/21/2016 08:00 AM - nobu (Nobuyoshi Nakada)

- Blocked by Feature #13056: base option to Dir.glob added

#7 - 12/21/2016 02:53 PM - shyouhei (Shyouhei Urabe)

We looked at this issue at today's developer meeting. As Akira already respond, the proposed patch has a problem. In order to reroute that we need to extend Dir.glob. Nobu opened a ticker for that. Once he finished that, we can resume this one.

#8 - 10/21/2017 08:35 AM - akr (Akira Tanaka)

- Status changed from Feedback to Closed

Applied in changeset [trunk|r60238](#).

Pathname#glob method implemented.

[ruby-core:49373] [Feature [#7360](#)] proposed by Alexander E. Fischer.

Files

19c4ae36aaea65b30e8b73c89f7ac42bd6905ed7.patch	2.84 KB	09/26/2015	til (Tilman Singer)
--	---------	------------	---------------------