

Ruby trunk - Feature #13789

Dir - methods

08/07/2017 09:31 PM - opti (Andreas Opti)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
Description		
If having (new) Dir.entries, Dir.chidren, I would also add: Dir.entries (includes all), Dir.files Dir.dirs Dir.symlinks Dir.pipes (Dir.something_more?) Dir.file?(filename); Dir.symlink[s]?(*symlinks), etc # give one or more names To be discussed if more names are useful, perhaps using Dir.file_s_, Dir.pipe_s_ => return count of given files, that are of that type. Also (a new) Dir.cd should return nil (no exception), if cd isn't sucessful, else the new path. thanks Opti		
Related issues:		
Related to Ruby trunk - Feature #11302: Dir.entries and Dir.foreach without [...		Closed

History

#1 - 08/10/2017 09:19 PM - shevegen (Robert A. Heiler)

I am having difficulties understanding your proposal.

If I understand it correctly, you propose to add one or more methods that would return, e. g., all **files** or all **directories** or all **symlinks** or all **pipes**, yes?

Then you also suggest **Dir.cd** ...

I have nothing at all against the proposal as such, but I believe that the ruby core team may prefer if things could be split up a bit, e. g. a separate proposal for Dir.cd.

The names Dir.file_s_ Dir.pipe_s_ are pretty bad so I do not think that these have any realistic chance to be implemented.

To be honest, I think that the current proposal is too convoluted.

It would be better if you would split it up IMO.

The ruby core team has shown to be amenable to suggestions; see File.empty? and Dir.empty? and similar useful methods.

I am not completely sure that the names that you picked would be accepted though; in particular "Dir.dirs" is orthogonal to "Dir.files", since you use the plural form of "file" in the latter, but "dirs" is not the plural form of "directory".

I am not among the ruby core team so I can not say anything there, but my gut feeling is that you may have a better chance if you would split the proposals up and make an individual case for these. There is already Dir.chdir(), but note that if Dir.cd() behaviour is different, then matz may possibly not accept it due to orthogonal behaviour and potential confusion for ruby users. (I could not explain to anyone else why Dir.chdir() would then behave differently to Dir.cd() as I can not think of a legit reason for that. It may be ok if it is an alias, I don't mind that; I myself mostly use wrapper methods, such as "def cd(i); Dir.chdir if File.directory?(i); end"

or something like that, and then directly use "cd something").

You have a point in regards to `Dir.entries`, and using shortcuts such as `Dir.files` etc... - while this is already doable via `.select/.reject` filters, considering that I also do use this a lot, I would not mind more support within ruby itself. (I use `Dir[]` almost exclusively though so I can not even say whether I would use `Dir.files`; for some reason, I found `Dir[]` to be superior to `Dir.entries` ... hard to explain why for me ... note that I am not entirely sure whether `Dir.files` etc.. would be a good addition as such, even though I am also the first to agree that it would be useful to be had. Perhaps let's wait for a bit and see what other people have to say about your proposal as well.)

#2 - 08/14/2017 04:07 PM - Eregon (Benoit Daloze)

`Dir.children(dir)` behaving as `Dir.entries(dir)` - `['.', '..']` would be nice. I have wished many times `Dir` would have such a method, and the `./..` are almost never useful.

#3 - 08/31/2017 08:48 AM - matz (Yukihiko Matsumoto)

- Status changed from Open to Rejected

Rejected. You can simply filter `Dir` entries. If you have concrete use-case of the method, reopen the issue, please.

Matz.

#4 - 09/14/2017 09:05 AM - Eregon (Benoit Daloze)

- Related to Feature #11302: Dir.entries and Dir.foreach without [".", ".."] added