

Ruby - Feature #17295

Feature: Create a directory and file with Pathname#touch

10/30/2020 03:06 PM - schneems (Richard Schneeman)

Status:	Rejected	
Priority:	Normal	
Assignee:	akr (Akira Tanaka)	
Target version:		
Description		
Right now if a developer wants to create a file and is not sure if the path exists yet or not they must:		
<pre>Pathname.new("/a/b/c/d.txt").tap { p p.dirname.mkpath; FileUtils.touch(p)}</pre>		
After this patch a developer can instead call:		
<pre>Pathname.new("/a/b/c/d.txt").touch</pre>		
An alternative name for this behavior could be mkfile but I think it is confusing to have a mkfile and a mkpath where one creates a directory and one creates a file.		
Diff:		
<pre>\$ git diff master diff --git a/ext/pathname/lib/pathname.rb b/ext/pathname/lib/pathname.rb index e6fb90277d..2ed02a6633 100644 --- a/ext/pathname/lib/pathname.rb +++ b/ext/pathname/lib/pathname.rb @@ -585,6 +585,27 @@ def mkpath nil end + # Creates a file and the full path to the file including any intermediate directories that don't yet + # exist. + # + # Example: + # + # Dir.exist?("/a/b/c") # => false + # + # p = Pathname.new("/a/b/c/d.txt") + # p.file? => false + # p.touch + # p.file? => true + # + # Dir.exist?("/a/b/c") # => true + def touch + require 'fileutils' + dirname.mkpath + + FileUtils.touch(self) + self + end + + # Recursively deletes a directory, including all directories beneath it. + # + # See FileUtils.rm_r diff --git a/test/pathname/test_pathname.rb b/test/pathname/test_pathname.rb index 43cef4849f..3c518cc3da 100644 --- a/test/pathname/test_pathname.rb +++ b/test/pathname/test_pathname.rb @@ -1394,6 +1394,14 @@ def test_mkpath } end</pre>		

```

+ def test_touch
+   with_tmpchdir('rubytest-pathname') { |dir|
+     Pathname("a/b/c/d.txt").touch
+     assert_file.directory?("a/b/c")
+     assert_file.file?("a/b/c/d.txt")
+   }
+ end
+
+ def test_rmtree
+   with_tmpchdir('rubytest-pathname') { |dir|
+     Pathname("a/b/c/d").mkpath

```

Github link: <https://github.com/ruby/ruby/pull/3706>

Related issues:

Related to Ruby - Feature #7361: Adding Pathname#touch

Rejected

History

#1 - 08/30/2021 06:51 AM - hsb (Hiroshi SHIBATA)

- Status changed from Open to Assigned

- Assignee set to akr (Akira Tanaka)

#2 - 09/01/2021 01:23 PM - Dan0042 (Daniel DeLorme)

I agree having Pathname#touch would be nice, but the issue of making sure the parent dir exists is not limited to touch.

I often have code such as `path.tap{ |p| p.dirname.mkpath }.open("a"){ ... }`

So I think here it would be nice to have something like Pathname#ensure_parent_dir_exists (but with a shorter name) that can be used in various situations:

```

path.ensure_parent_dir_exists.touch
path.ensure_parent_dir_exists.open('w'){...}
path.ensure_parent_dir_exists.write('w', str)
source.rename(dest.ensure_parent_dir_exists)

```

#3 - 09/02/2021 04:06 PM - schneems (Richard Schneeman)

For the example you gave:

```
path.tap{ |p| p.dirname.mkpath }.open("a"){ ... }
```

It looks like you want to ensure a file is created in a directory that exists. I actually think that would be a good use case for the proposed touch. It could be shorter as:

```
path.touch.open("a") { ... }
```

I see two cases:

- Want to mkdir -p the parent and the path points to a file: This proposed touch interface would accommodate that.
- Want to mkdir -p the parent and the path points to a dir: Then the dev can use mkpath. I can't think of a situation you would want to have the parent dir created, but not the full path.

I'm not opposed to adding a specialized method that creates the parent dir, but I think that should be a separate proposal. I also think the name would need to be both specific and short-ish:

```

path.ensure_parent_dir_exists.open('w'){...}
path.tap{|p|p.dirname.mkpath}.open('w'){...} # Same length if you remove whitespace

```

I think that adding a touch that also does mkdir -p of the parent dir buys us the same functionality (if there's some cases I've not considered, that would be good to put into the separate proposal.

Back to this proposal, we could add a touch that only creates the file without a mkdir -p. But I don't know why someone would ever want to touch a file that doesn't exist. If they're wanting an error there are other ways to get it. We could also make it configurable touch(skip_mkpath: true), however someone can still use the regular FileUtils.touch if they want:

```

touch(skip_mkpath: true)
tap{|p|FileUtils.touch(p)}

```

#4 - 09/14/2021 06:24 AM - knu (Akinori MURASHI)

Shouldn't this method take keyword arguments that FileUtils.touch accepts?

#5 - 09/15/2021 07:24 PM - schneems (Richard Schneeman)

Shouldn't this method take keyword arguments that FileUtils.touch accepts?

I looked into it. Of the existing pathnames that delegate to FileUtils, only one supports kwargs and it does not support all of them, just one:

```
def mkpath(mode: nil)
```

This was added by nobu 16 days ago <https://github.com/ruby/ruby/commit/2dd26bed86f721ed1982d00c3a0bd5ed37568e96>.

I explored what it would look like to support all kwargs and wrote it up. It ended up being a little involved:

<https://gist.github.com/schneems/681a42ee54aa91a2185f49556469b319>.

I am fine merging this and adding kwarg support as people see fit. Or if the rest of core wants it in I can add support for all the kwargs that I've described. I want to get some feedback before implementing such a change.

Pending an agreeable implementation what do you think of the opportunity to add such an interface?

#6 - 09/16/2021 01:17 AM - Dan0042 (Daniel DeLorme)

In the end I agree that touch is enough and ensure_parent_dir_exists is unnecessary (even with a shorter name). Even though creating the file via "touch" is kinda redundant before open('a') it's not really a problem either.

schneems (Richard Schneeman) wrote in [#note-5](#):

I explored what it would look like to support all kwargs and wrote it up. It ended up being a little involved:

<https://gist.github.com/schneems/681a42ee54aa91a2185f49556469b319>.

The nocreate option is intended to update the timestamp on an existing file. It's like "noop if file doesn't exist". So in the case the file doesn't exist, IMHO it shouldn't create the directories either.

#7 - 09/28/2021 01:20 AM - schneems (Richard Schneeman)

For what it's worth this idea isn't my favorite. I would LOVE to have a mktmpdir that returns a pathname instead of a string:

- <https://bugs.ruby-lang.org/issues/17297>

Also, this would be handy in cases:

- <https://bugs.ruby-lang.org/issues/17296>

For this touch feature, it's a nice-to-have. What do you think about adding touch() that just touches a file, and a kwarg that enables directory creation:

```
touch() # Just touches the file
touch(mkpath: true) # Touches and creates
```

#8 - 10/03/2024 04:06 AM - mame (Yusuke Endoh)

- Related to Feature #7361: Adding Pathname#touch added

#9 - 10/04/2024 02:07 AM - hsb (Hiroshi SHIBATA)

- Status changed from Assigned to Rejected

Pathname#touch is rejected at [this reason](#).

I asked this proposal to [@akr \(Akira Tanaka\)](#) again, he is still negative for Pathname#touch.