

Ruby - Feature #7361

Adding Pathname#touch

11/15/2012 10:30 PM - aef (Alexander E. Fischer)

Status:	Rejected		
Priority:	Normal		
Assignee:	akr (Akira Tanaka)		
Target version:			
Description			
Pathname has an #mkdir method to create a directory at the path, but does not have a #touch method to create an empty file at that path.			
There were numerous cases where I had to fallback to using			
new_file = Pathname.new('location/for/new/file')			
FileUtils.touch(new_file)			
instead of simply being able to use:			
new_file.touch			
I would like to add this method. If you like it, let me know. I will provide a patch then.			
Related issues:			
Related to Ruby - Feature #7378: Adding Pathname#write		Closed	06/02/2012
Related to Ruby - Feature #17295: Feature: Create a directory and file with P...		Rejected	

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 - 01/22/2015 02:39 PM - Salzig (Ben Rexin)

Any way i can help?

#3 - 01/24/2015 09:20 PM - djberg96 (Daniel Berger)

I'm a little surprise there isn't an option for this in FileUtils. Something like FileUtils.touch('/location/for/new', :force => true). I'd be in favor of adding that, or an option to FileUtils.mkpath.

#4 - 04/07/2016 03:35 PM - dsisnero (Dominic Sisneros)

Any update on this

```
class Pathname
  def touch(options={})
    FileUtils.touch(@path, options)
  end
end

Pathname('test.pdf').touch(mtime: 1.hour.ago)

"#mkpath" almost works but doesn't have the mtime option
```

#5 - 04/13/2016 06:50 AM - akr (Akira Tanaka)

- Status changed from Assigned to Rejected

I don't like the method name "touch".

"touch" command has multiple features:
create empty file, update mtime.
So, the intent of touch method is ambiguous.

#6 - 07/19/2016 07:25 PM - najamelan (Naja Melan)

I prefer something like (warning, untested code):

```
def touch( subPath = '', **options )

  path = @path
  ret  = self

  if directory?

    ret  = join subPath
    path = ret.to_path

  end

  FileUtils.touch( path, **options )

  ret

end
```

lets you do:

```
mydir.touch 'somefile'
```

I think touch is a good name because everybody already knows it and it's what people will try out spontaneously before even looking in the docs why it doesn't work. Joel Spolsky [explains quite well](#) why this is a good thing.

#7 - 07/20/2016 12:32 AM - shyuhei (Shyouhei Urabe)

We don't always blindly import broken designs; for instance Ruby do not have creat(2) equivalent method. "Everybody knows it" is (definitely a good property but) not enough.

When it comes to touch, there already is FileUtils.touch so your urgent needs are already satisfied I believe. Here on Pathname let us consider a clearer design.

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

- Related to Feature #17295: Feature: Create a directory and file with Pathname#touch added