

## Ruby trunk - Feature #12843

### Proposal to add a new method to class File in order to determine the name of the file without any suffix

10/16/2016 10:39 AM - shevegen (Robert A. Heiler)

<b>Status:</b> Open	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>Description</b>	
<p>We currently have File.basename() and File.dirname(), both of which work very well and are quite useful when working with files and directories.</p>	
<p>When you want to get the filename without extension, you can use something like this:</p>	
<pre>File.basename('foo/bar/test.rb', '.rb') File.basename('foo/bar/test.rb', '.*')</pre>	
<p>While this works very well, I was wondering if it would be possible to add a method that does precisely that for you but which requires only one argument - the path. That would allow us to omit the ',' and the second argument, which I think would be nice to have.</p>	
<p>Resulting ruby code may be a tiny bit shorter:</p>	
<pre>File.basename('foo/bar/test.rb', '.*') File.methodXY('foo/bar/test.rb')</pre>	
<p>Where .methodXY() should be the name of the new method added.</p>	
<p>I would thus like to suggest a way for a new method addition, on the class File namespace, just like .basename() and .dirname(), that will return the filename of a given path/file, but without any suffix and without any path - so, similar to File.basename() but to additionally already chop away all extname suffixes.</p>	
<p>Giving things a proper, meaningful name is often difficult. In particular when method names should ideally be short; both .dirname and .basename are quite short too.</p>	
<p>My obvious choice would be:</p>	
<pre>File.filename()</pre>	
<p>But I am not sure if this is a good name. A problem is that I can not come up with a better name.</p>	
<p>We also already have File.extname() in order to determine the "extension name" of the file.</p>	
<p><a href="https://ruby-doc.org/core-2.2.0/File.html#method-c-extname">https://ruby-doc.org/core-2.2.0/File.html#method-c-extname</a></p>	
<p>So we essentially have 3 behaviours - obtain the name of the directory via .dirname; obtain the name of the file/entry itself, via .basename; and being able to obtain the name of the extension via .extname.</p>	
<p>What we actually appear to be missing is to obtain the part of the name of a path/file/directory WITHOUT the extension and WITHOUT the leading dirname part.</p>	

If `File.filename()` is not a good name then perhaps any of the following may serve as better alternatives; some are obviously not really great names though, I add them mostly for comparison:

```
File.no_suffix()
File.corename()
File.corefile()
File.shortname()
File.short_name()
File.sname()
File.filename_without_extension() # Probably too verbose but
# probably more descriptive than the other names
```

Anyway, I guess the name of the method is one thing - if the functionality itself would be approved, then I am sure a good name can be found for the functionality anyway if none of the above are very good choices. I just thought that it should fit towards `.dirname`, `.basename` and `.extname`, hence why I put `.filename` first.

Thanks for reading!

## History

#1 - 10/16/2016 11:53 PM - nobu (Nobuyoshi Nakada)

- Description updated

Robert A. Heiler wrote:

I would thus like to suggest a way for a new method addition, on the class `File` namespace, just like `.basename()` and `.dirname()`, that will return the filename of a given path/file, but without any suffix and without any path - so, similar to `File.basename()` but to additionally already chop away all `extname` suffixes.

Maybe what is called as "stem" in GNU make document?

```
File.filename()
```

It sounds like same as `basename`.

What we actually appear to be missing is to obtain the part of the name of a path/file/directory WITHOUT the extension and WITHOUT the leading `dirname` part.

What about WITHOUT the extension but WITH the leading `dirname` part?

```
File.no_suffix()
```

It sounds like keeping the prefix.

```
File.corename()
File.corefile()
```

"core" has a different meaning.

```
File.shortname()
File.short_name()
File.sname()
```

These remind me a Windows short file name.

```
File.filename_without_extension() # Probably too verbose but
# probably more descriptive than the other names
```

No ambiguity, but, too verbose.

## #2 - 10/17/2016 08:56 AM - Eregon (Benoit Daloze)

What is your use-case?

If it's to append another extension, `Pathname#sub_ext` might interest you. Actually, it can even be used to remove the extension:

```
Pathname("dir/file.ext").sub_ext("") # => #<Pathname:dir/file>
```

In path (<https://github.com/eregon/path>), I named this operation `rm_ext`. It also keeps the `dirname` part as nobu mentions, so what you would need is

```
Path("dir/file.ext").rm_ext.basename  
# or  
Path("dir/file.ext").basename.rm_ext
```

I think it's good to separate these two concerns as they compose well.

If one wants to use raw strings, then I think `File.basename(path, ".**")` is already quite short.