

Ruby master - Feature #5422

File.fnmatch != Dir.glob # {no,sets}

10/08/2011 07:06 AM - sunaku (Suraj Kurapati)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	3.0
Description	
Hello,	
The File.fnmatch methods do not support Dir.glob's set notation:	
<pre>Dir.glob '{.g,t}' => [".gem", "test"] File.fnmatch? '{.g,t}', 'test' => false File.fnmatch? '{.g,t}*', '.gem' => false</pre>	
Please add set notation to fnmatch() and make it equal to glob().	
Thanks for your consideration.	

Associated revisions

Revision e59d5667 - 11/04/2012 01:19 AM - nobu (Nobuyoshi Nakada)

dir.c: FNM_EXTGLOB

- dir.c (file_s_fnmatch): match with expanding braces if FNM_EXTGLOB is set. [ruby-core:40037] [Feature #5422]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37463 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37463 - 11/04/2012 01:19 AM - nobu (Nobuyoshi Nakada)

dir.c: FNM_EXTGLOB

- dir.c (file_s_fnmatch): match with expanding braces if FNM_EXTGLOB is set. [ruby-core:40037] [Feature #5422]

Revision 37463 - 11/04/2012 01:19 AM - nobu (Nobuyoshi Nakada)

dir.c: FNM_EXTGLOB

- dir.c (file_s_fnmatch): match with expanding braces if FNM_EXTGLOB is set. [ruby-core:40037] [Feature #5422]

Revision 37463 - 11/04/2012 01:19 AM - nobu (Nobuyoshi Nakada)

dir.c: FNM_EXTGLOB

- dir.c (file_s_fnmatch): match with expanding braces if FNM_EXTGLOB is set. [ruby-core:40037] [Feature #5422]

Revision 37463 - 11/04/2012 01:19 AM - nobu (Nobuyoshi Nakada)

dir.c: FNM_EXTGLOB

- dir.c (file_s_fnmatch): match with expanding braces if FNM_EXTGLOB is set. [ruby-core:40037] [Feature #5422]

Revision 37463 - 11/04/2012 01:19 AM - nobu (Nobuyoshi Nakada)

dir.c: FNM_EXTGLOB

- dir.c (file_s_fnmatch): match with expanding braces if FNM_EXTGLOB is set. [ruby-core:40037] [Feature #5422]

Revision 37463 - 11/04/2012 01:19 AM - nobu (Nobuyoshi Nakada)

dir.c: FNM_EXTGLOB

- dir.c (file_s_fnmatch): match with expanding braces if FNM_EXTGLOB is set. [ruby-core:40037] [Feature #5422]

History

#1 - 10/21/2011 02:10 AM - sunaku (Suraj Kurapati)

I suppose this issue could be filed as a feature request since nobody else has complained about it yet. Sorry for the wrong categorization. Thanks.

#2 - 10/21/2011 03:08 AM - trans (Thomas Sawyer)

I have. Obviously it's not earth shattering, but it really SHOULD be the same, so that glob functionality can be exactly emulated within code on arrays of strings. I've had need of this a couple of times.

#3 - 10/21/2011 06:46 AM - drbrain (Eric Hodel)

- Tracker changed from Bug to Feature

I have moved it to the Feature tracker for you.

#4 - 10/21/2011 12:39 PM - nobu (Nobuyoshi Nakada)

```
=begin
```

What about a new method to expand braces?

```
File.expand_brace("{.g,t}").any? {|pat| File.fnmatch?(pat, name)}
```

or a shorthand

```
File.expand_brace("{.g,t}") {|pat| File.fnmatch?(pat, name)}
```

```
=end
```

#5 - 10/21/2011 12:40 PM - nobu (Nobuyoshi Nakada)

- Target version changed from 2.0.0 to 3.0

#6 - 10/21/2011 04:01 PM - trans (Thomas Sawyer)

Why more complexity? Is there reason that fnmatch? can't do it?

#7 - 10/21/2011 07:26 PM - nobu (Nobuyoshi Nakada)

Because it comes from POSIX fnmatch() function.

<http://pubs.opengroup.org/onlinepubs/9699919799/functions/fnmatch.html>

#8 - 10/21/2011 11:02 PM - trans (Thomas Sawyer)

I see. In that case, adding a new File.glob_match? (or maybe just File.match?) would be better.

File.expand_brace won't really help here b/c it would be tied to the actual file system. Part of the usefulness of fnmatch? is that it is not.

#9 - 10/22/2011 11:29 AM - nobu (Nobuyoshi Nakada)

```
=begin
```

Thomas Sawyer wrote:

I see. In that case, adding a new File.glob_match? (or maybe just File.match?) would be better.

The former doesn't sound appropriate, because "glob" does not imply brace expansion originally.

```
$ case foo in f{oo,ar}) echo ok;; *) echo ng;; esac
```

```
ng
```

```
$ case "f{oo,ar}" in f{oo,ar}) echo ok;; *) echo ng;; esac
```

```
ok
```

File.expand_brace won't really help here b/c it would be tied to the actual file system. Part of the usefulness of fnmatch? is that it is not.

Brace expansion is a string to array conversion operation, and not used only for file names in shell scripts.

```
=end
```

#10 - 10/22/2011 01:47 PM - trans (Thomas Sawyer)

"The former doesn't sound appropriate, because "glob" does not imply brace expansion originally."

I'm only going by the functionality of Dir.glob. But, whatever name seems best.

"Brace expansion is a string to array conversion operation, and not used only for file names in shell scripts."

Okay, I mistook your example's use of * and similarity of name to File.expand_path to think that it was. File.expand_brace is fine if you think it would be useful in and of itself. But the real need is for a Dir.glob compatible match method.

#11 - 03/16/2012 03:27 AM - sunaku (Suraj Kurapati)

What if we pass a File::FNM_EXTGLOB flag to the File.fnmatch methods?

File.fnmatch? '{.g,t}*', '.gem', File::FNM_EXTGLOB

The name "extglob" comes from the Bash/Zsh option for extended globbing.

#12 - 03/18/2012 11:06 AM - trans (Thomas Sawyer)

Works for me if it works for the API.

#13 - 11/03/2012 05:05 AM - sunaku (Suraj Kurapati)

Thanks for your agreement Thomas.

Hey Ruby-core developers, we have agreed on the following API, where passing a File::FNM_EXTGLOB flag (the name "extglob" comes from the Bash/Zsh option for extended globbing) to the File.fnmatch methods enables support for Dir.glob's set notation.

File.fnmatch? '{.g,t}*', '.gem', File::FNM_EXTGLOB

Do you also agree on this API? Could you please implement this for us? If not, I'll try submitting a patch.

Thanks for your consideration.

#14 - 11/04/2012 10:19 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r37463.

Suraj, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

dir.c: FNM_EXTGLOB

- dir.c (file_s_fnmatch): match with expanding braces if FNM_EXTGLOB is set. [ruby-core:40037] [Feature [#5422](#)]