

## Ruby - Feature #8709

### Dir.glob should return sorted file list

07/31/2013 01:24 AM - tommorris (Tom Morris)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>		
<b>Description</b> On OS X, Dir.glob and Dir[] return an ordered list of files.  On Ubuntu Linux, they do not and one must manually sort them.  Returning a list of files that isn't in order fails the Principle of Least Astonishment.  I attach a unit test to demonstrate ideal behaviour.		

#### Associated revisions

---

**Revision 2f1081a451f21ca017cc9fdc585883e5c6ebf618 - 01/19/2020 05:46 AM - nobu (Nobuyoshi Nakada)**

Sort globbed results by default [Feature #8709]

Sort the results which matched single wildcard or character set in binary ascending order, unless sort: false is given. The order of an Array of pattern strings and braces are not affected.

**Revision 2f1081a451f21ca017cc9fdc585883e5c6ebf618 - 01/19/2020 05:46 AM - nobu (Nobuyoshi Nakada)**

Sort globbed results by default [Feature #8709]

Sort the results which matched single wildcard or character set in binary ascending order, unless sort: false is given. The order of an Array of pattern strings and braces are not affected.

**Revision 2f1081a4 - 01/19/2020 05:46 AM - nobu (Nobuyoshi Nakada)**

Sort globbed results by default [Feature #8709]

Sort the results which matched single wildcard or character set in binary ascending order, unless sort: false is given. The order of an Array of pattern strings and braces are not affected.

#### History

---

**#1 - 07/31/2013 02:49 AM - Anonymous**

- Status changed from Open to Rejected

Dir.glob is documented to return filenames in filesystem order:

Note that case sensitivity depends on your system, *as does the order in which the results are returned*.

**#2 - 01/11/2020 06:28 AM - bmwiedemann (Bernhard M. Wiedemann)**

- Status changed from Rejected to Open

There are two problems with unsorted glob:

1. it is different from glob in C, bash and perl that all sort by default. Even GNU make finally switched back to sorted wildcard/glob ( <https://savannah.gnu.org/bugs/index.php?52076> )
2. it causes problems for reproducible builds, so that developers have to patch an infinite number of callers such as

<https://github.com/sass/sassc-ruby/pull/178>

to be able to get identical build results on identical OSES on different machines.

### #3 - 01/11/2020 11:18 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Rejected

- Backport deleted (1.9.3: UNKNOWN, 2.0.0: UNKNOWN)

Do not update the status without a maintainer's decision.

### #4 - 01/11/2020 11:27 AM - Eregon (Benoit Daloze)

- Tracker changed from Bug to Feature

- Status changed from Rejected to Open

- ruby -v deleted (ruby 1.9.3p429 (2013-05-15) [x86\_64-linux] Brightbox)

I agree always sorting the result of Dir.glob makes sense.

Non-determinism caused by Dir.glob is very annoying and IMHO doesn't feel like Ruby.

I would also expect sorting is a low overhead compared to syscalls, so performance-wise I think it's not a big hit.

FWIW, TruffleRuby returns sorted results for Dir.glob since 2016.

hsbt (Hiroshi SHIBATA) wrote:

Do not update the status without a maintainer's decision.

How should we rediscuss this then?

It's not because the documentation mentions it we should never change it.

I'll reopen as a Feature.

### #5 - 01/11/2020 11:32 AM - hsbt (Hiroshi SHIBATA)

I have no opinion about this feature.

### #6 - 01/11/2020 11:33 AM - Eregon (Benoit Daloze)

Here are some benchmark results in the ruby repository:

```
$ ruby -e 'p Dir["**/*"].size'
12171
```

```
$ ruby -rbenchmark -e '10.times { p Benchmark.realtime { Dir["**/*"] } }'
0.017877419999422273
0.015390422999189468
0.015255956001055893
0.015021605999208987
0.015777969998453045
0.015484851002838695
0.016179073001694633
0.015210424000542844
0.015358253996964777
0.014319942998554325
```

```
$ ruby -rbenchmark -e '10.times { p Benchmark.realtime { Dir["**/*"].sort } }'
0.017600111998035572
0.017109740001615137
0.017832364999776473
0.01726310600133729
0.018130796997866128
0.01659841600121581
0.018173008000303525
0.017528833999676863
0.017515739000373287
0.01770434499849216
```

So a bit slower but we can likely optimize further if desired.

### #7 - 01/11/2020 11:35 AM - Eregon (Benoit Daloze)

I added this issue to the next meeting's agenda:

<https://bugs.ruby-lang.org/issues/16454>

**#8 - 01/11/2020 08:48 PM - bmwiedemann (Bernhard M. Wiedemann)**

The benchmark numbers above show a difference of 12%

That is probably the worst case, because usually, globs will return fewer entries (though for some strange reason I get a 20% diff on a dir with 200 entries)

and usually some processing will be performed on the returned files and that will take much longer than the sorting.

**#9 - 01/13/2020 09:28 AM - byroot (Jean Boussier)**

For what it's worth I also think it should return a sorted array, because:

- Pretty much any rubyist I know have been bitten by this at least once.
- Many experienced rubyist end up always writing `Dir[patten].sort`
- It's particularly prevalent because the "develop on OSX, deploy on Linux" combo is very popular.

If the performance impact is a concern, I think an extra keyword argument could be added: `glob( pattern, [flags], [base: path], [sort: true] )`, this way you can avoid the performance impact if you know that you don't need it.

**#10 - 01/13/2020 10:04 AM - deivid (David Rodríguez)**

I got bit by this in the past too when trying to reproduce order dependent test failures ( [https://github.com/rubygems/rubygems/pull/2626#discussion\\_r254020218](https://github.com/rubygems/rubygems/pull/2626#discussion_r254020218) ).

**#11 - 01/13/2020 06:49 PM - jhawthorn (John Hawthorn)**

One potential issue with this is that though globs which scanned directories (ex. `Dir.glob("foo/*")`) would return results in an inconsistent order, globs which used purely brace expansion (ex. `Dir.glob("foo/{a,b,c,d}")`) would return values predictably in the order listed.

Rails versions prior to 6.0 unfortunately relied on this behaviour (6.0+ in *most* cases doesn't and does sorting manually). It probably shouldn't have relied on it, but it did, and I fear other libraries or tools may have done the same.

We could possibly work around that by sorting when reading directory entries rather than sorting the full result, but that's more complicated to implement and would be hard to document as an exact behaviour developers can expect/rely upon.

**#12 - 01/14/2020 06:27 AM - naruse (Yui NARUSE)**

the Principle of Least Astonishment.

You shouldn't use "the Principle of Least Astonishment".

Without the term you need to explain why the current behavior is bad and need to change.

For example ...

the result of `Dir.glob` depends a OS and filesystem. People often wrongly write code which depends their local environment.

Though people should carefully write portable code, could we provide a guard to protect people from such pitfalls?

Many people write specs which compare the result of `Dir.glob` and an expected array, and fails.

If `Dir.glob` sort the result, people can avoid pitfalls and reduce the cost of writing such specs.

**#13 - 01/14/2020 07:27 AM - mame (Yusuke Endoh)**

Hi [@jhawthorn \(John Hawthorn\)](#), I'm unsure whether you agree with the proposal or not. Do you mean sorting the result may break Rails? Or not sorting the result may do so, i.e., are you against the change?

**#14 - 01/14/2020 10:40 AM - Eregon (Benoit Daloze)**

[@jhawthorn \(John Hawthorn\)](#) Good point, I forgot to mention this.

The sorting must respect explicit order for `{...,...}` and conceptually the same as sorting just after `readdir(3)`, not on the full result to be correct.

That's also likely more efficient, due to sorting smaller arrays.

ruby/spec already captures this, 3 specs fail if sorting is done on the returned array instead of per directory.

**#15 - 01/14/2020 10:44 AM - Eregon (Benoit Daloze)**

Even C's `glob(3)` is sorted (by default), as [@bmwiedemann](#) said:

```
$ man 3 glob
```

```
...
```

```
    GLOB_NOSORT
```

```
        Don't sort the returned pathnames. The only reason to do this is to save processing time. By default, the returned pathnames are sorted.
```

#16 - 01/16/2020 05:22 AM - nobu (Nobuyoshi Nakada)

I'm for adding NOSORT option to the second argument.

#17 - 01/16/2020 06:03 AM - matz (Yukihiro Matsumoto)

Accepted. We will add sort: false keyword option to disable sorting.

Matz.

#18 - 01/16/2020 03:12 PM - Dan0042 (Daniel DeLorme)

It's good to sort the result of Dir["\*"], but as jhawthorn pointed out the brace expansion *must* keep the same order. I have code that depends on this, and I'm sure many others also have code that depend on this, since it's the behavior found in the shell:

```
$ touch a2 a1 a0 b2 b1 b0
$ echo {a,b}?
a0 a1 a2 b0 b1 b2
$ echo {b,a}?
b0 b1 b2 a0 a1 a2
```

#19 - 01/19/2020 06:54 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset [git|2f1081a451f21ca017cc9fdc585883e5c6ebf618](#).

Sort globbed results by default [Feature [#8709](#)]

Sort the results which matched single wildcard or character set in binary ascending order, unless sort: false is given. The order of an Array of pattern strings and braces are not affected.

Files

globtest.rb	454 Bytes	07/31/2013	tommorris (Tom Morris)
-------------	-----------	------------	------------------------