

## Ruby master - Misc #13787

### The path to Ruby 3.x - would it be useful to have a separate thread here at the tracker, for discussions and issues and ideas related to ruby 3.x?

08/07/2017 09:09 PM - shevegen (Robert A. Heiler)

<b>Status:</b>	Open
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Description</b>	
<p>Hello everyone but especially so the whole ruby-core team,</p> <p>This is very long, so if you just want the short gist, please jump to the:</p> <p>TL;DR line closer to the bottom.</p> <p>Matz gave several presentations in the last ~3 years or so (and of course before that as well; but I am mostly focusing only on the 3 recent years, in particular because due to the path towards ruby 3.x).</p> <p>I lately watched the presentation matz gave in Singapore and he made some references to things that will, quite likely, go into ruby 3.x, while he also said that there is a ton of work ahead - JIT compile strategies; the (optional, I guess) type system; better concurrency and so on and so forth. These are probably the big ideas - while this is nice, and I agree with comments such as "nobody minds if ruby gets faster" (everyone loves when things get done instantly rather than when you must wait), I myself am actually even more interested in smaller things. Even when they are incompatible with ruby 2.x, but I think that matz said that backwards-compatibility breaking changes may happen in ruby 3.x but not in ruby 2.x. Which is both good and bad - good because people do not have to rewrite stuff to run in ruby 2.x; bad if there are some habits or anything that isn't that great when writing software in ruby.</p> <p>Sorry for my lengthy introduction so far - I wanted to explain the angle I am coming from here.</p> <p>Now ...</p> <p>I have wanted to make some smaller changes and suggestions towards ruby 3.x. And matz once said that the core team (and, well, matz, since he is the boss :D ) is open to ideas. In ruby 2.x, often the core team asks for use cases, which is fine - some proposals have not been thought through and others are purely theoretical; whereas others are also originating from some real use case. For ruby 3.x, it is obviously ... difficult to get a real use case out. :)</p> <p>But one can actually try to make the argument about this or that in ruby 2.x not working very well, or being confusing.</p> <p>For example, some days ago, I wanted to suggest a simplified API and usage for running external programs in ruby. We have <code>system()</code>, <code>backticks</code>, <code>IO</code>, <code>popen</code>, <code>open3</code> ... and I do not know what else. All of which works slightly differently and it is, at the least to me, quite confusing sometimes. So I wanted to suggest a simplification... but I realized that this is not really a good topic for the ruby 2.x branch because I think that matz wants to retain compatibility, so such a proposal probably has not a big chance to be implemented. Perhaps for some minor changes, but I'd actually also suggest to get rid of some of the duplicate ways, while retaining flexibility still (so I'd say... <code>system</code> and <code>backticks</code> remain, and then</p>	

perhaps only one or two more ways for more advanced use cases, via a simpler API than this strange ... open2 open3 and what not...).

Anyway.

If you think this through then it probably does not fit much into "Features" for ruby 2.x because ruby 2.x will remain backwards compatible, I think.

But for ruby 3.x, it may fit.

There are many more smaller ideas like this... for example, I'd like to see the warning/error situation be improved in ruby 3.x in particular.

I'd also like case/when objects to become full objects and be passable/convertible ad-hoc into a hash or hash-like object, and re-used in different .rb files. (The workarounds I know about are usually to use a hash, with these aliases, but I found case/when menu to be so much more readable and nicer to use, especially as you can also use regexes).

And many more ideas like that!

But they all do not really fit into the "Bug" section, neither into the "Feature" subsection... they may fit somewhat into "Misc" but it's not a good category.

So, without any further explanation and further ado, here is the summary and proposal:

TL;DR - Would it be useful to have a separate thread here at the issue tracker, for discussions and issues and ideas related to ruby 3.x?

This can be kept primarily as an ideas section for the time being; at a later time, when ruby 3.x becomes the official ruby one day, the subsection can be closed and archived for, say, 3 years or so as read-only, before it would then be deleted (when ruby 3.x is eventually used more frequently than ruby 2.x, which I guess may take a few years ... there is a LOT of inertia in general, we can see this in perl, python etc..).

The reason why I think that this would be useful is because it could potentially foster some idea discussions. Getting early ideas in may be good, because ruby 3.x, while it is still quite far away I think (2020 or beyond?), getting ideas in early may help to see that they, or a variation, is considered for inclusion. Otherwise we then may have to wait for more years because ruby 3.x may be frozen and people would have to wait for ruby 4.x - this is the primary reason why I'd like to see a separate thread.

I can not say if this is useful or not to anyone else, and I can not say if this leads to more ideas or discussion for ruby 3.x features (aside from the big ideas matz already mentioned several times before, such as better concurrency/guilds, JIT and speed improves in general or any optional type system - that's already quite a lot to work on, and all this in addition to mruby ... but I think ideas are not bad, even if it is not possible to implement several of them due to lack of time. When matz approves of an idea, who knows, perhaps someone else may see that the approved idea will be worked on).

Sorry for this lengthy note - feel free to close this at any moment in time, for any reason!

---

## History

---

**#1 - 08/08/2017 08:29 AM - naruse (Yui NARUSE)**

Just use this Redmine project.

The most important feature of bug tracker is people can check info at only one place.  
Distribute information multiple places is bad idea.

Moreover whether a feature is 2.x or 3.x is decided by Matz, not reporters.

But they all do not really fit into the "Bug" section, neither into the "Feature" subsection... they may fit somewhat into "Misc" but it's not a good category.

In practice the difference between "Bug" and "Feature" is decided by whether it may need backport or not.  
The point reflects whether the issue has backport field or not.

Therefore tickets which reporters may consider it as 3.x feature should be in Feature tracker.  
(Such big change won't be backported)