# Ruby master - Misc #16130

## [Discussion / Ideas] Finding a good name for the concept of/behind guilds - primarily the NAME

08/27/2019 11:59 AM - shevegen (Robert A. Heiler)

| | | |
|---|---|---|
| **Status:** | Open | |
| **Priority:** | Normal | |
| **Assignee:** | | |

**Description**

In recent presentions this year, and perhaps prior to that as well if I
remember correctly, matz mentioned that the core team (and matz) may be
looking for a good name to the concept of/behind guilds.

The thread here, this issue, is PRIMARILY confined with this, the name -
if you have any good suggestion for names in this context, or discussions
that may relate to this secondarily, please feel free to chime in and
comment. This is primarily meant to give some ideas, possibly.

Since koichi is also heavily involved here (matz pointed out that koichi
likes the name guilds, and came up with the idea/proposal/implementation),
I think this should be considered too. And of course how ruby users may
want to use/view the concept behind guilds, and actually use them in
their own code - the best idea is not great if nobody is using the
concept. Like with refinements ... great idea but I found the API
somewhat strange. :D (May also be because subclassing is so easy with
"Foo < Bar"; ideally we could have something like this with refinements
too, but this is for another proposal or discussion - this here is
about the name for the concept behind guilds.)

Anyway.

I'll give my opinion too, on the names, but I will decouple this from
the initial suggestion here, and reply to my own issue.

Note that this here really is primarily concerned with finding a good
NAME, which is not trivial, since names may have different meanings
in different contexts.

Furthermore, some links for those who may be curious - some of which
are old, and I really just randomly linked these in:

http://www.atdot.net/~ko1/activities/2016_rubykaigi.pdf
https://mensfeld.pl/2016/11/getting-ready-for-new-concurrency-in-ruby-3-with-guilds/
https://olivierlacan.com/posts/concurrency-in-ruby-3-with-guilds/

### History

**#1 - 08/27/2019 11:59 AM - shevegen (Robert A. Heiler)**

(1) The name "Guilds":

To me the name guild implies something that is related to one another.

In medieval days, the concept of a guild was mostly used to refer e. g.
to the "guild of mastersmiths", that is people of a certain profession.
They may also agree to not undercut their own profession and profits,
e. g. by not working too cheaply, and such.

There is another concept that I know of guilds, that comes from ecology.

For example, within the bird family, there are different guilds that can
use different resources/niches, depending on the shape of their beak.
Birds with a larger, heavier beak, may often wish to crack hard(er),
stronger nuts, whereas birds with a smaller beak, may specialize on
other plant seeds - this is also a concept of guilds, just from ecology.

As for ruby, I think the name guild does not reveal a lot of intent or intention. It could mean a lot. If we are to assume that guilds relate to resources related to threads, and sharing data between threads, then perhaps this may make some sense, but I think the fundamental problem with the name itself here, if there is one, is that it does not carry a whole lot of a "specific description" related to the main function at hand.

I think this is the primary reason why another name may be better than guilds. But ... which name, then? Often suggesting another name is not necessarily better.

Guilds relate to concurrency in ruby, but I think the name guilds per se is not that good. I hope koichi does not mind this comment, it is just an opinion, and people have lots of opinions. :)

Guilds are assumed to allow communication with one another using "channels" (Guild::Channel) - at the least that was proposed at one point. I think the name "Channel" is semi-ok, but it should be pointed out that we have similar names, such as "shared mutex/semiphore". So in this case, we would have to be able to explain what the difference is between channels, mutex and semaphores... and fibers ... and couroutines... and threads ... and ... well.

In my opinion, no matter which name is chosen, I think it would be best to stick with the same name for the same or very closely related concepts. Otherwise it can become quite confusing if we use different terminology.

I also assume that part of the name choice comes from the fact that objects can be copied and moved between guilds, so this is where the "relatedness" aspect comes in.

Note that in the URL given above, koichi views Guilds as implementation in terms of existing Thread and Fibers. So this may also be a reason why the name was chosen; but even there I am still not sure if the name is great. Actually, I wonder if fibers, as names, could be removed, and everything be put into Threads as-is ... but I guess this may not be easy to do at this point ... ;)

A good resource to read was:

https://olivierlacan.com/posts/concurrency-in-ruby-3-with-guilds/

However, it is a bit complicated, and ideally this could all be much, much simpler ... I don't think I was the only one to notice that this is not trivial. The GIL is not trivial either. :\

Part of the problem is with the API, I think such as:

```
Guild::Channel.
transfer_membership(object)
```

(I may have made up that API, but how many can easily remember that API and what it does? And how it should be used, even more so when we already have Threads, Mutex and Fibers? And that is just one ...)

(2) The name "Isolate" / "Isolates":

Matz suggested this name, and I think it carries a better intent than guilds, purely from the name part alone. People may have an easier time associating with what "isolates" do in ruby (if they understand the concept).

I am not sure if "isolates" fully capture the intent behind the name guilds, though; for example, sharing state? That does not seem to be fully addressed by isolates, or perhaps this comes simply from another point of view. With the name "Isolates" we put into focus the idea that individual "threads" may run in a way that they are completely unhindered by other threads or events that may be ongoing in these threads. Because they may be "isolated from one another".

I think one problem with the name "isolates" as such, though, is that it is a bit clunkier to use overall, at the least for me. "Guilds" as a name sounds a bit better, even though "isolates" as a name may carry a slightly more specific intent for its purpose.

(3) The name "Fibers" / "AutoFibers":

Matz mentioned this too during presentations. Perhaps this is used in other languages.

To me personally, it does not tell me that much. I still don't know what fibers really are. :D

You could say the same thing about threads, but oddly enough I have adjusted to threads as a name fairly well. No clue how others think about threads.

(4) The name "Coroutines" / "Rutines":

I tried to coin the latter word to ruby ... Go has goroutines or something lke that, so ruby may have Rutines. ;) But the name "Rutines" is not good; goroutines sounds better because of the word "Go". (By the way, names of programming languages are REALLY strange if you think about it; "Go" means to walk, right? So that is weird too ... and Rust, that is also a strange name ... C, C++, D ... perl, ruby, python ... why name something after an animal. But I digress ... ;))

Coroutines may be used in other programming languages, so perhaps it fits. I have no real opinion on that, though.

(5) The name "Mutex" / "Semaphore":

I think Mutex came from "mutually exclusive", so perhaps this is where matz came from, when the name "isolates" was proposed, which seem to make sense or relate to that concept. At the least you can read this often how difficult it is to share state between threads.

I have no idea about where the name "semaphore" came from, but I have heard it before somewhere. Lots of words here.

(6) The name "Threads":

Matz also mentioned this. Personally I like Threads. I used them back in the days when the pickaxe showed example, to download different URLs at the "same time", and it was easy to work with.

My personal favourite choice would be to group this all under "Threads", no matter which name is chosen, simply because I think it makes the most sense.

But this may not be ideal either, for example, "Threads::Guilds" would be longer to type than e. g. just "Guilds" (toplevel) or just "Fibers", rather than "Threads::Fibers".

The reason why I like the name Threads the most is possibly because I have already adjusted to threads, and have used them in the past, and found them quite simple.

I have not really used mutexes, fibers etc... and for a reason I am not completely sure, I think the whole concept here is quite difficult to understand. Much harder than Threads.

Many parts of ruby are much, much easier to understand (Array, String, Hash, even Enumerators and Enumerable). I don't know if I am the only one who has a problem with this here, but if you look at the API examples from:

https://olivierlacan.com/posts/concurrency-in-ruby-3-with-guilds/

Then I may not be the only one who thinks this is a bit confusing and complex.

Ideally, no matter what name is chosen, the simpler this could be, the simpler for ruby users to use, the better. Names are important but I think no matter which name is ultimately choosen, if it is simple then people can use it. If the usage is too difficult then only a few may use it.

Perhaps there could be an experimental API before ruby 3.0, no matter the name used, so people can experiment with it and see which make the most sense. And, if wanted, past that point, there could also be a poll (although I am a bit sceptical of polls in general, people are not always creatures of logic; but perhaps it may help to see which names are better, and

then pick the best name from e. g. the top three or something, or at the
least narrow it down. The reason why I think a specific API may be important,
on top of it being ideally simple, is because people need to somehow use
code as well, in order to refer to the idea through code - no matter if the
name is Guilds or Isolates or any other name. This is why I think an experimental
API that is encouraged by the core team, may be helpful.).

Anyway, feel free to make any comment related to the name(s) itself. API
usage is very important too, no matter who proposed which name, IMO.

Olivier also proposed this, on the blog entry:

> I'd encourage the Ruby core team to release this new feature under an
> opt-in experimental flag so that the Ruby community can participate in
> testing.

I think this could ideally be done with ruby 2.7 this year; an experimental
flag to try out some APIs or so, to see how people may want to use it. The
"final" name can be finalized in 2020, for example, perhaps a bit before
3.0 is released.

PS: I think when a name has been chosen, or at the latest when ruby 3.0
is released, this issue here can be closed.

It is only meant to give ideas and thoughts behind the name(s) really, so
past that point it is no longer that important.

#### #2 - 08/27/2019 01:27 PM - sawa (Tsuyoshi Sawada)

If Fiber composes Thread, then Thread composes Yarn, or Rope.

#### #3 - 08/27/2019 04:00 PM - shan (Shannon Skipper)

A Braid of Threads also makes sense in that metaphor. Twine is another option similar to Rope or Yarn.

I do like the name Guild since it's a thing than can be joined to cooperate for a common purpose.

What would the terminology equivalent to "transferring membership from one Guild to another" be for Braid, Yarn, Rope or Twine? I guess "transferring membership from one Braid/Yarn/Rope/Twine to another" could still work?