Ruby master - Feature #17500
Move RubyVM::* to ExperimentalFeatures

01/01/2021 12:43 PM - Eregon (Benoit Daloze)

Status: Rejected
Priority: Normal
Assignee:
Target version: 3.1

Description
RubyVM is a trap:

- Users will think from the name it's some official blessed API when it is meant as experimental unstable MRI-specific APIs (the complete opposite).
- CRuby developers seem to see it as an experimental module to put constants and methods which are not stable yet.
- About half the things under RubyVM is MRI-specific and the other half not, which is confusing and problematic for other Ruby implementations.

I think the best way to solve this is to move everything under RubyVM to ExperimentalFeatures, and deprecate RubyVM. That achieves multiple important things:

- it makes it clear such constants/methods are experimental (documentation is not enough, https://bugs.ruby-lang.org/issues/17490#note-6)
- it makes it possible for other Ruby implementations to implement such API, if it makes sense for compatibility
- it avoids the common pitfall of CRuby developers thinking an API is MRI-specific when in fact it's not, and such an API often end up being used in gems and so other Ruby implementations must implement it for compatibility.

In my opinion, keeping the current status quo is irresponsible for compatibility between Rubies. Users end up using e.g. RubyVM::AbstractSyntaxTree in gems (often missing the fact it's experimental), which is something other Rubies can technically implement, but currently are prevented to since RubyVM is supposed to be MRI-specific. That's just one example, I think more than half of what has been under RubyVM is not MRI-specific.

So rather than guessing if some new experimental API is implementation-specific, let's add all new experimental APIs under ExperimentalFeatures.

I believe this is better for everyone.

It means there is no module for MRI-specific features. That's on purpose, every method in MRI is eventually relied upon by some gem or program, so it should be in ExperimentalFeatures if experimental and somewhere as a normal API otherwise.

Related issues:
Related to Ruby master - Feature #15743: RubyVM should be renamed to CRuby  Closed
Related to Ruby master - Feature #15752: A dedicated module for experimental ... Feedback

History
#1 - 01/01/2021 12:43 PM - Eregon (Benoit Daloze)
- Related to Feature #15743: RubyVM should be renamed to CRuby added

#2 - 01/01/2021 12:43 PM - Eregon (Benoit Daloze)
- Related to Feature #15752: A dedicated module for experimental features added

#3 - 01/01/2021 12:44 PM - Eregon (Benoit Daloze)
- Description updated

#4 - 01/01/2021 12:45 PM - Eregon (Benoit Daloze)
- Description updated

#5 - 01/01/2021 12:56 PM - Eregon (Benoit Daloze)

Migration plan:

- In Ruby 3.1: define ExperimentalFeatures, add the same constants & methods as RubyVM under it, and deprecate the RubyVM constant.
  User code can do ExperimentalFeatures = RubyVM unless defined?(ExperimentalFeatures) to work seamlessly with Ruby<~3.0 and Ruby>=3.1.

03/11/2022
In Ruby 3.2: remove the RubyVM constant.

#6 - 01/01/2021 11:23 PM - k0kubun (Takashi Kokubun)

it is meant as experimental unstable MRI-specific API

I think it means only MRI-specific API, while some of it may be experimental and/or unstable. For example, RubyVM::InstructionSequence will continue to be MRI-specific, but I don't think it's experimental anymore (although unstable). So ExperimentalFeatures does not explain its nature. CRuby or MRI would be the accurate name, if we were to change that part.

#7 - 01/02/2021 11:12 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

I am not opposed to renaming RubyVM module for the reason Eregon (Benoit Daloze) stated. But I am strongly against renaming it to ExperimentalFeatures or whatever means experimental. Users are forced to rewrite their code when the feature becomes out of the experimental phase. It's not good.

In addition, the features under RubyVM might be reorganized to CRuby specific ones and ones available on multiple VMs. For example, RubyVM::MJIT is certainly CRuby specific. RubyVM::AbstractSyntaxTree might not.

I am OK with renaming RubyVM to CRubyVM for example. I am not fully satisfied with renaming it to MRI for ambiguity.

Matz.

#8 - 01/28/2021 08:56 PM - Eregon (Benoit Daloze)

Thank you for both of your replies. I took a while to respond, because I wasn't sure how to phrase this.

Essentially, it seems this main point was lost:

Nobody knows if some API will always be MRI-specific.

For instance, RubyVM::InstructionSequence might make sense to implement on some other Ruby implementation which also uses bytecode.

So given that reasoning, I think there should be no MRI-specific module, because that can only hurt compatibility with other Ruby implementations.

Users are forced to rewrite their code when the feature becomes out of the experimental phase.

But so what should happen when the feature gets out of experimental phase, and that feature was defined under RubyVM? Either it moves, and users need to adapt anyway, or it stays in RubyVM, but then RubyVM loses its meaning of being experimental (even more).

RubyVM feels like a place where people put not-yet-ready APIs, whether it's MRI specific or not. But this just doesn't work, people use RubyVM even in gems, and what was once thought as experimental APIs is forced to remain there forever due to wide usage (e.g., RubyVM::InstructionSequence).

So I think the whole concept of RubyVM being an experimental module is already broken.

I think a good way to get out of this mess is to no longer add new APIs under RubyVM. Every core Ruby API should have enough thought to have a proper name (i.e., not under RubyVM), or start under some Experimental module, where it's clear to everyone it won't stay there.

I wanted to clarify the meaning of the RubyVM module (#15743), but clearly there is no easy answer there. If we actually renamed to CRubyVM (the main advantage would be it's clearer it's MRI-specific), then eventually there would be some API there that was thought as MRI-specific but actually is not, and such API might become widely used in gems.

And then what should other Ruby implementations do? Define a CRubyVM module too? (weird of course) Forever being unable to support that gem?

I think the main point that is missed about RubyVM is that it's experimental stuff that might change in incompatible ways. That's why I think highlighting that it's experimental is more important than highlighting it's MRI-specific. After all, if it's experimental APIs, it's already clear it's not portable and not stable, and cannot be relied on to be generally available.

If we don't rename RubyVM, I think it's clear what will happen. Some RubyVM API there will become widely used, other Ruby implementations will have no choice but to define RubyVM (not the end of the world). But then nobody would know what RubyVM means anymore. And then RubyVM will no longer be a place for experimental features. And it won't be MRI-specific anymore. It will be "just another constant defined in the Ruby core library".

So, to get ahead of this, how about taking the step to add experimental API under a Experimental/ExperimentalFeatures module? (#15752)