

## Ruby - Misc #18346

### DevelopersMeeting20211209Japan

11/18/2021 02:32 PM - mame (Yusuke Endoh)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Description</b>		
<h2>The next dev meeting</h2>		
<b>Date:</b> 2021/12/09 13:00-17:00 (JST)		
Place/Sign-up/Agenda/Log: <a href="https://github.com/ruby/dev-meeting-log/blob/master/DevelopersMeeting20211209Japan.md">https://github.com/ruby/dev-meeting-log/blob/master/DevelopersMeeting20211209Japan.md</a>		
<ul style="list-style-type: none"><li>• Dev meeting <i>IS NOT</i> a decision-making place. All decisions should be done at the bug tracker.</li><li>• Dev meeting is a place we can ask Matz, nobu, nurse and other developers directly.</li><li>• Matz is a very busy person. Take this opportunity to ask him. If you can not attend, other attendees can ask instead of you (if attendees can understand your issue).</li><li>• We will write a log about the discussion to a file or to each ticket in English.</li><li>• All activities are best-effort (keep in mind that most of us are volunteer developers).</li><li>• The date, time and place are scheduled according to when/where we can reserve Matz's time.</li><li>• <i>DO NOT</i> discuss then on this ticket, please.</li></ul>		
<h2>Call for agenda items</h2>		
If you have a ticket that you want matz and committers to discuss, please post it into this ticket in the following format:		
<pre>* [Ticket ref] Ticket title (your name)   * Comment (A summary of the ticket, why you put this ticket here, what point should be discussed , etc.)</pre>		
Example:		
<pre>* [Feature #14609] `Kernel#p` without args shows the receiver (kol)   * I feel this feature is very useful and some people say :+1: so let discuss this feature.</pre>		
<ul style="list-style-type: none"><li>• It is recommended to add a comment by 2021/12/06. We hold a preparatory meeting to create an agenda a few days before the dev-meeting.</li><li>• The format is strict. We'll use <a href="#">this script to automatically create an markdown-style agenda</a>. We may ignore a comment that does not follow the format.</li><li>• Your comment is mandatory. We cannot read all discussion of the ticket in a limited time. We appreciate it if you could write a short summary and update from a previous discussion.</li></ul>		
<b>Related issues:</b>		
Related to Ruby - Misc #14770: [META] DevelopersMeeting		Open

## History

### #1 - 11/18/2021 02:32 PM - mame (Yusuke Endoh)

- Related to Misc #14770: [META] DevelopersMeeting added

### #2 - 11/23/2021 06:22 AM - k0kubun (Takashi Kokubun)

- [Feature #18349] Let --jit enable YJIT on supported platforms (k0kubun)
  - Any concerns? Can we do this from Ruby 3.1?

### #3 - 11/25/2021 04:32 PM - Eregon (Benoit Daloze)

- [Feature #16038] Provide a public WeakMap that compares by equality rather than by identity (eragon)
  - This is necessary to have access to a weak-keys map on TruffleRuby/JRuby. There are many use cases:, notably [this PR in Rails](#) and all the reasons WeakHashMap is used in Java code.

#### #4 - 11/29/2021 04:59 PM - jeremyevans0 (Jeremy Evans)

- [Feature [#18351](#)] Support anonymous rest and keyword rest argument forwarding (jeremyevans0)
  - I think this is a natural addition after the addition of anonymous block forwarding in [#11256](#).
  - Is it OK to support this syntax? If so, is the patch acceptable?
- [Feature [#16663](#)] Add block or filtered forms of Kernel#caller to allow early bail-out (jeremyevans0)
  - Two possible method names are find\_caller and each\_caller.
  - find\_caller optimizes for the case where the first matching frame will be returned.
  - each\_caller is more general, always iterating unless there is a non-local exit (break/return).
  - Are either of these names acceptable? If so, which method name should be used?
- [Feature [#12084](#)] Class#instance (jeremyevans0)
  - This should be simple to implement, since a singleton class keeps a reference to the instance.
  - Are we OK adding this feature? If so, is the method name acceptable, or do we want singleton\_instance?
  - If called on a non-singleton class, should TypeError be raised?
  - For TrueClass, FalseClass, and NilClass, should it return true, false, and nil?
- [Bug [#14891](#)] Pathname#join has different behaviour to File.join (jeremyevans0)
  - I agree with [@znz \(Kazuhiko NISHIYAMA\)](#) that this behavior is expected.
  - Can this be closed?
- [Bug [#14434](#)] IO#reopen fails after EPIPE (jeremyevans0)
  - I'm not sure whether this behavior is a bug.
  - The behavior has existed since Ruby 1.9 (no failure in 1.8.7).
  - Can we decide whether this behavior is expected or a bug?

#### #5 - 11/30/2021 02:37 PM - Eregon (Benoit Daloze)

- [Feature [#18370](#)] Call Exception#full\_message to print exceptions reaching the top-level (eregon)
  - Improve consistency in how exceptions are shown, and makes it easier to evolve exception formatting.
  - OK to do?

#### #6 - 11/30/2021 03:47 PM - peterzhu2118 (Peter Zhu)

- [Feature [#18364](#)] Add GC.stat\_pool for Variable Width Allocation (peterzhu2118)
  - This feature will allow accessing stats for memory pools through Ruby. This will be especially useful for the Variable Width Allocation feature.
  - Should we extend GC.stat to return stats for memory pools or create a new method such as GC.stat\_pool?
  - Is GC.stat\_pool a good name?

#### #7 - 12/03/2021 11:19 PM - byroot (Jean Boussier)

- [Feature [#17881](#)] Add a Module#const\_added callback
  - Would allow the Zeitwerk autoloader to no longer use tracepoint.
  - The main problem with Zeitwerk using tracepoint is that since tracepoint callbacks are not "reentrant" Zeitwerk is partially broken when using a debugger, e.g. <https://github.com/ruby/debug/issues/408>
  - This cause lots of confusion for users.
  - Could alternatively be Module#module\_defined, to only be called for Module/Class

#### #8 - 12/04/2021 08:31 AM - byroot (Jean Boussier)

- [Feature [#15912](#)] Allow some reentrancy during TracePoint event
  - Alternative to Module#const\_added for fixing the Zeitwerk and debuggers combinaison.
  - [@ko1 \(Koichi Sasada\)](#) proposed Tracepoint#reopen(&block) so that debuggers can yield to the user with that.
  - Potentially with a list of reopened events: tp.reopen(:class) { "do stuff" }

#### #9 - 12/07/2021 04:26 AM - shugo (Shugo Maeda)

- [Feature [#12737](#)] Module#defined\_refinements
  - Is the name Module#refinements acceptable?
    - It's consistent with Module#constants.
    - Other candidates: configured\_refinements, contained\_refinements, defined\_refinements
  - Is it OK to return an Array instead of Hash?
    - The target class of a refinement can be obtained by another new method Refinement#refined\_class.
  - I'd like to add Module#refinements and Refinement#refined\_class after the release of Ruby 3.1.
- [Feature [#14332](#)] Module.used\_refinements to list refinement modules
  - eregon suggested to add it soon, but It may be too late to add it in Ruby 3.1.

#### #10 - 12/07/2021 06:07 AM - nobu (Nobuyoshi Nakada)

- [Feature [#18033](#)] <https://bugs.ruby-lang.org/issues/18033>
  - Time.at("2021-12-09") instead of Time.new.
  - Other new methods: Time.try\_convert or Time.iso.
- [Feature [#18190](#)] Split Random::Formatter from securerandom
  - Already [ruby/securerandom#7](#) has been merged.
- [Feature [#18183](#)] make SecureRandom.choose public

#### #11 - 12/07/2021 06:43 AM - sawa (Tsuyoshi Sawada)

- [Feature [#18331](#)] Kernel.#Time (sawa)
  - Introduce Kernel.#Time {instead of/in addition to} extending Time.new as in [#18033](#).
  - Similar methods like Kernel#.Integer are used to parse strings, and this is a good fit for Time.

#### #12 - 12/09/2021 12:30 PM - mame (Yusuke Endoh)

- Description updated

- Status changed from Open to Closed