

## Ruby - Misc #17299

### DevelopersMeeting20201120Japan

10/31/2020 12:55 AM - 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> 2020/11/20 13:00-17:00	
Place/Sign-up/Agenda/Log: <a href="https://github.com/ruby/dev-meeting-log/blob/master/DevelopersMeeting20201120Japan.md">https://github.com/ruby/dev-meeting-log/blob/master/DevelopersMeeting20201120Japan.md</a>	
<ul style="list-style-type: none"><li>• Dev meeting <i>IS NOT</i> a decision-making place. All decisions should be made on 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 cannot attend, a different attendee can ask instead of you (if the attendee can understand your issue).</li><li>• We will write a log about the discussion in a file or on 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' time.</li><li>• <i>DO NOT</i> discuss them on this ticket, please.</li></ul>	
<h2>Call for agenda items</h2>	
If you have a ticket that you want Matz and other committers to discuss, please post it on 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 points 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's discuss this feature.</pre>	
<ul style="list-style-type: none"><li>• Comment deadline: 2020/11/13 (one week before the meeting)</li><li>• The format is strict. We'll use <a href="#">this script to automatically create a markdown-style agenda</a>. We may ignore comments that do not follow the format.</li><li>• Your comment is mandatory. We cannot read the entire discussion of the ticket within a limited time.</li></ul>	
<b>Related issues:</b>	
Related to Ruby - Misc #14770: [META] DevelopersMeeting	Open

## History

#1 - 10/31/2020 12:55 AM - mame (Yusuke Endoh)

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

#2 - 10/31/2020 01:53 AM - sawa (Tsuyoshi Sawada)

- Description updated

#3 - 11/02/2020 11:15 AM - hsb (Hiroshi SHIBATA)

- [Feature [#17303](#)] Make webrick to bundled gems or remove from stdlib
  - Any objection?

#4 - 11/02/2020 08:45 PM - jeremyevans0 (Jeremy Evans)

- [Feature [#17143](#)] Improve support for warning categories (jeremyevans0)
  - I researched the Python warning categories, and provided some analysis.
  - Can we decide on warning categories so they can be implemented in time for Ruby 3?

- [Bug [#10845](#)] Subclassing String (jeremyevans0)
  - In [#6087](#), matz decided that in Ruby 3.0, methods for Array subclasses should return arrays and not subclass instances.
  - The same issues that apply to Array subclasses also apply to String subclasses.
  - Do we want to make String subclass methods return strings instead of subclass instances?
  - If so, is the pull request OK?
  - Note that changing the behavior of String methods will require changes in Rails.
- [Bug [#11022](#)] opening an eigenclass does not change the class variable definition context (jeremyevans0)
  - This was discussed in the December 2019 meeting, but a decision was not made.
  - Considering [#14012](#) was rejected last meeting, can we also reject this?
- [Bug [#7844](#)] include/prepend satisfiable module dependencies are not satisfied (jeremyevans0)
  - This was discussed in the December 2019 meeting, but a decision was not made.
  - Considering the other improvements to include/prepend in 3.0, it would be a good time to fix this.
- [Bug [#11213](#)] defined?(super) ignores respond\_to\_missing? (jeremyevans0)
  - I agree with Koichi that it is best to accept the current behavior as spec.
  - Can we reject this?

#### #5 - 11/03/2020 10:13 AM - byroot (Jean Boussier)

- [Feature [#13381](#)] Expose rb\_fstring and its family to C extensions (byroot)
  - The feature itself was approved, however we're still waiting on an agreement on the naming of one function (<https://github.com/ruby/ruby/pull/3586>)
  - Both json and msgpack now call String#-@, and could benefit from large allocation reduction if they could call rb\_fstring\_cstr instead.

#### #6 - 11/03/2020 08:25 PM - marcandre (Marc-Andre Lafortune)

- [Bug [#17197](#)] Yielding arity for Hash methods (marcandre)
  - Hash#each with arity 1: confirmed?
  - #select, #keep\_if, #delete\_if, #reject and to\_h should also be changed, right?
  - #map: should only accept arity 1, right?

#### #7 - 11/04/2020 12:28 AM - mrkn (Kenta Murata)

- [Feature [#17291](#)] Optimize \_\_send\_\_ call
  - Improve the performance of \_\_send\_\_ call by the following ways
    1. Omit emitting \_\_send\_\_ calls for literal method name cases
    2. Introduce a new instruction sendsym to skip \_\_send\_\_ call for dynamic method name cases
  - I guess it is no problem to skip \_\_send\_\_ call because the following warning says "*redefining \_\_send\_\_ may cause serious problems*" for a long time
  - I want to know Matz's reaction to this

#### #8 - 11/04/2020 10:46 AM - Eregon (Benoit Daloze)

- [Feature [#16043](#)] \$LOAD\_PATH.resolve\_feature\_path should not raise (eregon)
  - Seems good to me, OK to merge?

#### #9 - 11/05/2020 10:38 AM - Eregon (Benoit Daloze)

- [Feature [#17100](#)] Ractor naming (eregon)
  - What should be the method name to send a message? send seems problematic as many have reported. We should decide soon. Many good other names are listed in the issue (notably in <https://bugs.ruby-lang.org/issues/17100#note-19> and <https://bugs.ruby-lang.org/issues/17100#note-43>).
  - Specifically, Ractor#send conflicts with Kernel#send, which is a violation of the Liskov substitution principle.
  - There has been (AFAIK) no official deprecation of Kernel#send, so it seems inadvisable to use the name for other semantics for Ractor. Deprecating Kernel#send seems a way to require 100 000s of changes in gems for for very little reasons, so it seems unfeasible in practice anyway.
  - When should send and \_\_send\_\_ be used? It seems gems prefer to use send (>10x more used than \_\_send\_\_), and only use \_\_send\_\_ when required. IMHO \_\_send\_\_ feels like Python's \_\_add\_\_, which I find unpretty, and feels like C's poor way of namespacing things.

#### #10 - 11/05/2020 11:12 AM - Eregon (Benoit Daloze)

- [Feature [#17307](#)] A way to mark C extensions as thread-safe, Ractor-safe, or unsafe (eregon)
  - I'd like feedback on this. Any idea for a good way to mark C extensions?
  - How about defining symbols like foo\_is\_thread\_safe/foo\_is\_ractor\_safe to mark C extensions?

#### #11 - 11/07/2020 02:39 PM - Eregon (Benoit Daloze)

- [Feature [#16786](#)] Light-weight scheduler for improved concurrency (eregon)
  - @matz (Yukihiro Matsumoto): The Fiber.set\_scheduler(value) API doesn't make any sense to me and is very confusing, please see my

comments there (summary: <https://bugs.ruby-lang.org/issues/16786#note-70>).

- Anything wrong with the obvious and clear `Thread.current.scheduler = value` or `Thread.current.fiber_scheduler = value` ?

**#12 - 11/09/2020 01:28 PM - zverok (Victor Shepelev)**

- [Feature [#17312](#)] New methods in Enumerable and Enumerator::Lazy: flatten, product, compact (zverok)
  - If there would be no objections, I'd prepare a PR

**#13 - 11/26/2020 02:13 PM - mame (Yusuke Endoh)**

- *Description updated*

- *Status changed from Open to Closed*