

Ruby - Misc #19074

DevMeeting-2022-12-01 (postponed from 11-17)

10/21/2022 11:23 AM - mame (Yusuke Endoh)

Status: Closed

Priority: Normal

Assignee:

Description

The next dev meeting

Date: 2022/11/17 13:00-17:00 (JST)

Log: <https://github.com/ruby/dev-meeting-log/blob/master/DevMeeting-2022-12-01.md>

- Dev meeting *IS NOT* a decision-making place. All decisions should be done at the bug tracker.
- Dev meeting is a place we can ask Matz, nobu, nurse and other developers directly.
- 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).
- We will write a record of the discussion in the file or to each ticket in English.
- All activities are best-effort (keep in mind that most of us are volunteer developers).
- The date, time and place of the meeting are scheduled according to when/where we can reserve Matz's time.
- *DO NOT* discuss then on this ticket, please.

Call for agenda items

If you have a ticket that you want matz and committers to discuss, please post it into this ticket in the following format:

```
* [Ticket ref] Ticket title (your name)
* Comment (A summary of the ticket, why you put this ticket here, what point should be discussed, etc.)
```

Example:

```
* [Feature #14609] `Kernel#p` without args shows the receiver (ko1)
* I feel this feature is very useful and some people say :+1: so let discuss this feature.
```

- It is recommended to add a comment by 2022/11/14. We hold a preparatory meeting to create an agenda a few days before the dev-meeting.
- The format is strict. We'll use [this script to automatically create an markdown-style agenda](#). We may ignore a comment that does not follow the format.
- 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.

Related issues:

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

Open

History

#1 - 10/21/2022 11:23 AM - mame (Yusuke Endoh)

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

#2 - 10/24/2022 10:08 AM - ioquatix (Samuel Williams)

- [Feature #19078] Introduce Fiber#storage for inherited fiber-scoped variables.
 - Updated based on feedback, is it acceptable?

#3 - 10/25/2022 10:38 AM - byroot (Jean Boussier)

- [Bug #19079] Modules included in a DelegateClass cannot override delegate methods (jonathanhefner)
 - Proposed PR: <https://github.com/ruby/delegate/pull/14#pullrequestreview-1154572034>
 - Define delegation methods in an included module instead of directly in the class.
 - This allow to call super.
 - Very minimal backward compatibility concerns.

#4 - 10/31/2022 01:49 PM - Eregon (Benoit Daloze)

- [Bug [#18899](#)] Inconsistent argument handling in IO#set_encoding ([@Eregon \(Benoit Daloze\)](#))
 - Could someone review Jeremy's fix and merge it?
 - The logic for IO set_encoding is really complicated (also following the C code for it is hard), I wish we could simplify it.

#5 - 11/05/2022 02:40 PM - tenderlovmaking (Aaron Patterson)

- [Feature [#18814](#)] Add a query method to check Ractor Queue count
 - Proposed PR: <https://github.com/ruby/ruby/pull/5973>
 - Good for monitoring
 - Sender can decide if it should be queued or not

#6 - 11/05/2022 03:10 PM - mame (Yusuke Endoh)

- [Feature [#19104](#)] Introduce the cache-based optimization for Regexp matching (mame)
 - This optimization to onigmo makes most Regexp matching linear time. In short, it will solve ReDoS.
 - There are two notable limitations: (1) there are some unoptimizable Regexp patterns (e.g., back-references. About 10% Regexp objects are not optimizable according to measurement with some real-world libraries), and (2) it consumes memory proportional to the input string (According to measurement, at worst, about 10 times to the length of the input string)
 - We want you to discuss: (1) can we merge this change (into preview3 if possible), and (2) should we provide a way to tell users if a given regexp is optimizable or not (e.g., a warning at unoptimizable regexp creation, a new API like Regexp#optimizable?, or a new opt-in Regexp flag /foo/r to raise a SyntaxError when it is not optimizable, etc.)

#7 - 11/07/2022 06:16 PM - Eregon (Benoit Daloze)

- [Feature [#11689](#)] Add methods allow us to get visibility from Method and UnboundMethod object. (eregon)
 - {Method,UnboundMethod}#{private?,protected?,public?} exist in 3.1 but are currently not defined in 3.2
 - I think we should add them back, I think there is no reason to remove them anymore, since the behavior is intuitive now. Of course we need to store and return the visibility of the original method found by .method/.instance_method, not the zsuper-resolved method (same as for owner).
 - I think it is unnecessary to wait for someone to open a ticket "private?,protected?,public? methods missing in 3.2".

#8 - 11/07/2022 06:37 PM - jeremyevans0 (Jeremy Evans)

- [Bug [#19012](#)] BasicSocket#recv* methods return an empty packet instead of nil on closed connections (jeremyevans0)
 - It seems like a good idea to be able to differentiate an empty packet versus a closed connection.
 - Is it OK to change the behavior to return nil for closed connections?
- [Bug [#19003](#)] TracePoint behavior inconsistency in 3.2.0-preview2 (jeremyevans0)
 - Behavior has changed for this case since Ruby 3.1 to fix other TracePoint bugs.
 - Do we consider the new behavior a bug, or an implementation detail?

#9 - 11/07/2022 10:02 PM - st0012 (Stan Lo)

- [Feature [#18996](#)] Proposal: Introduce new APIs to reline for changing dialog UI colours
 - After discussing with [@k0kubun \(Takashi Kokubun\)](#), we've decided to use iTerm2's settings as colour options' naming reference.
 - We also think it's better to accept colour configs at both DialogRenderInfo and completion level (which builds on top of DialogRenderInfo but doesn't expose its APIs). This will give us enough flexibility for future extension.
 - Do we think it's ready for implementation?

#10 - 11/09/2022 08:30 PM - jeremyevans0 (Jeremy Evans)

- [Bug [#19113](#)] Inconsistency in retention of compare_by_identity flag in Hash methods (jeremyevans0)
 - Is it OK to change Hash.[] to never return a compare_by_identity hash?
 - Is it OK to change Hash.ruby2_keywords_hash to not drop the compare_by_identity flag for empty hash arguments?
 - Is it OK to change Hash#compact to copy the default value/proc and compare_by_identity flag?

#11 - 11/10/2022 12:04 PM - byroot (Jean Boussier)

- [Feature [#19107](#)] Allow trailing comma in method signature (byroot)
 - I would like to make def foo(a,), def foo(a:,) and def foo(&block,) valid syntax.
 - Trailing commas are popular when breaking up a very long statement into multiple lines.
 - Method calls, array literals, hash literals, all accept trailing commas, but method definitions don't.
 - This would seem more consistent to me.
- [Feature [#18951](#)] Object#with to set and restore attributes around a block (byroot)
 - It's not clear to me what was decided when this was first discussed in a previous meeting.
 - I kinda understand that with is seen as too generic, if so would with_attr be acceptable?
 - I added a handful of real world code snippets that would be simplified by this method.

- Any other concerns?
- [Feature [#19117](#)] Include the method owner in backtraces, not just the method name (byroot)
 - from /tmp/foo.rb:4:in 'Foo::Bar#inspect' instead of just from /tmp/foo.rb:4:in 'inspect'.
 - Would make backtrace much easier to understand just by reading them without jumping to the definition every time.
 - @Dan0042 suggested to display the *receiver* rather than the *owner*.

#12 - 11/12/2022 11:14 PM - bdewater (Bart de Water)

- [Feature [#19000](#)] Data: Add "Copy with changes method" [Follow-on to [#16122](#) Data: simple immutable value object] (bdewater)
 - There seems to be consensus for the need of the feature, but not on what the method name should be. The proposals are with (C#, Java, Sorbet, value_semantics gem) and dup (with arguments, similar to Sequel datasets).
 - I think it would be valuable to have this decided and shipped with 3.2 to make Data objects ergonomic to work with.

#13 - 11/16/2022 11:09 PM - shugo (Shugo Maeda)

- [Bug [#19132](#)] ** with other keyword parameters causes "no anonymous keyword rest parameter" (shugo)
 - RBS expects no name for keyword rest parameters while test_method.rb etc expects **:*. Can we choose **: for consistency?

#14 - 11/17/2022 05:00 AM - mame (Yusuke Endoh)

This dev meeting has been postponed to 1st Dec. for various reasons.

If you have any topics you really want us to discuss, you can add them by 30th Nov. (But we already have a lot on the agenda, so I would appreciate your consideration.)

#15 - 11/17/2022 05:36 AM - duerst (Martin Dürst)

- Subject changed from DevMeeting-2022-11-17 to DevMeeting-2022-12-01 (postponed from 11-17)

#16 - 11/17/2022 09:30 AM - shugo (Shugo Maeda)

- [Feature [#19134](#)] ** is not allowed in def foo(...)
 - I believe ** should be allowed, or both * and & should be prohibited.

#17 - 11/21/2022 01:50 PM - Eregon (Benoit Daloze)

- [Bug [#19108](#)] Format routines like pack blindly treat a string as ASCII-encoded ([@Eregon \(Benoit Daloze\)](#))
 - Is it OK to make unknown pack directive an ArgumentError?
 - If not is it OK to make it a non-verbose warning (i.e., shown with default \$VERBOSE=false)?

#18 - 11/25/2022 04:26 PM - Eregon (Benoit Daloze)

- [Bug [#19150](#)] pack/unpack silently ignores unknown directives (eragon)
 - Similar to [#19108](#) just above.
 - I believe unknown directive should be ArgumentError, not silent unless VERBOSE is true.

#19 - 12/02/2022 03:16 AM - mame (Yusuke Endoh)

- Description updated

- Status changed from Open to Closed