

Status:	Closed
Priority:	Normal
Assignee:	

Description

The next dev meeting

Date: 2023/11/30 13:00-17:00 (JST)

Log: <https://github.com/ruby/dev-meeting-log/tree/master/2023>

- 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 2023/11/27. 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 - 11/10/2023 10:07 AM - mame (Yusuke Endoh)

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

#2 - 11/10/2023 01:34 PM - peterzhu2118 (Peter Zhu)

- [Feature #19993] Optionally free all memory at exit (HParker & peterzhu2118)
 - Environment variable RB_FREE_ON_SHUTDOWN to free all memory during shutdown.
 - Useful for using memory checkers (e.g. Valgrind, ASAN) to find memory leaks.
 - Has found 8 memory leaks in Ruby.
 - The feature is incomplete, and is currently experimental (emits an experimental warning when enabled).

#3 - 11/13/2023 01:16 AM - hsb (Hiroshi SHIBATA)

- [Feature #11322] OpenUri: RuntimeError: HTTP redirection loop
 - <https://github.com/ruby/open-uri/pull/18>

- Is max_redirects okay for this?

#4 - 11/16/2023 08:14 AM - byroot (Jean Boussier)

- [Feature #19998] Emit deprecation warnings when the old (non-Typed) Data_XXX API is used (byroot)
 - The new API is available since 2010, and the old API is officially deprecated since 2015.
 - Yet some relatively popular gem maintainers didn't know about it (<https://github.com/mudge/re2/pull/116#pullrequestreview-1725029520>)
 - In most cases the transition is rather straightforward.
 - Getting rid of the old API would allow to reclaim 8B in TypedData.
 - Could we start emitting deprecation warnings in 3.3 or 3.4 to at least surface the remaining issues?
 - I'm happy to help any high profile gem migrate.

#5 - 11/22/2023 05:32 AM - hsbt (Hiroshi SHIBATA)

- [Misc #20015] Privacy policy for ruby-lang.org
 - I would like to review <https://github.com/ruby/www.ruby-lang.org/pull/3144>

#6 - 11/22/2023 03:22 PM - tagomoris (Satoshi Tagomori)

- [Feature #20005] Add C API to return symbols of native extensions resolved from features
 - The new C API for extensions, to resolve&return symbols of other extensions
 - <https://github.com/ruby/ruby/pull/8991> (tests are not written yet)
 - This provides a better way to check whether a dependency extension is correctly loaded previously, or not (than call-and-crash trial)
 - #19744 (or similar namespace/package-like features) requires extensions to access symbols of other extensions via this (or similar) API
 - So if this gets merged into Ruby earlier, extension maintainers can start the work to support namespace/package features earlier

#7 - 11/22/2023 09:02 PM - jeremyevans0 (Jeremy Evans)

- [Bug #20008] f(**kw, &block) calls block.to_proc before kw.to_hash (jeremyevans0)
 - I think this is a bug as it goes against expected evaluation order.
 - Is fixing this using a new VM instruction and the peephole optimizer acceptable?
- [Bug #20012] Fix keyword splat passing as regular argument (jeremyevans0)
 - Should we fix this for Ruby 3.3, or wait until Ruby 3.4 (my preference would be 3.3)?
 - Should we backport this fix to Ruby 3.2 or 3.1 (my preference would be both)?
- [Bug #18886] Struct aref and aset don't trigger any tracepoints. (jeremyevans0)
 - I implemented support using the same approach used for attr_{reader,writer}.
 - Is the pull request acceptable?
- [Bug #19779] eval "return" at top level raises LocalJumpError (jeremyevans0)
 - I developed a pull request to fix this.
 - Is the pull request acceptable?
- [Feature #20011] Reduce implicit array allocations on caller side of method calling (jeremyevans0)
 - Some of these optimizations have the same theoretical issues as f(*a, &lvar) and f(*a, &@ivar) currently.
 - I think the benefit of the optimizations exceeds the costs (unexpected behavior in pathological cases).
 - Are these optimizations acceptable?
 - If not acceptable, should we fix the theoretical issues with f(*a, &lvar) and f(*a, &@ivar), at the cost of performance?

#8 - 11/24/2023 06:25 PM - mdalessio (Mike Dalessio)

- [Misc #19980] Is the Ruby 3.3 ABI frozen? (flavorjones)
 - Precompiled native gems (nokogiri, sqlite3, etc.) have always released support for a new version of Ruby many days or weeks after Ruby's release, slowing adoption of Ruby in some cases.
 - rake-compiler-dock v1.4.0.rc1 has been released with support for Ruby 3.3.0-preview3, primarily for testing by gem maintainers.
 - I would like to ship a final release of rake-compiler-dock as early as possible to allow gem maintainers to release native gems that support Ruby 3.3 before the 3.3.0 final release. When can I do this safely?

#9 - 11/27/2023 08:28 PM - kddnewton (Kevin Newton)

- [Feature #20024] SyntaxError subclasses (kddnewton)
 - Many tools rely on the specific error message coming from syntax errors.
 - Many specs require a specific error message for syntax errors.
 - It seems like the error message is hiding a "type" for syntax errors. Could we subclass SyntaxError to make this interface easier on the consumer? It would allow us to change the error message in the future, and also allow a nicer interface for tooling.

#10 - 11/28/2023 01:50 AM - k0kubun (Takashi Kokubun)

- [Feature #18980] Re-reconsider numbered parameters: it as a default block parameter (k0kubun)
 - Reminder to resume the discussion from the previous dev-meeting. I'll update here once I finish my action item.

#11 - 11/28/2023 01:32 PM - kddnewton (Kevin Newton)

- [Bug [#19877](#)] Non intuitive behavior of syntax only applied to literal value (tompng)
 - if (1; cond1..cond2); end becomes a flip-flop, because 1 and () are dropped
 - (1; (2; 3; (4; /(?!<a>)/))) =~ s assigns to a, because integers and () are dropped
 - This behavior seems unintentional and confusing. You have to know which values will be dropped in order to know if it's going to become a flip-flop/named capture. No tooling parsers do this (I checked prism, parser, ruby_parser, lib-ruby-parser, and syntax tree). It appears that some implementations do (JRuby/TruffleRuby) but not others (Natalie, artichoke).
 - There has previously been discussion of adding these nodes back into the tree for universal_parser. That would probably mean this behavior would change, since they would now be in the parse tree.
 - I would like to propose requiring that to be a named capture assignment that it be the only part of the expression, and for a flip-flop the same except that it follows and/or like it currently does.

#12 - 11/28/2023 08:16 PM - zverok (Victor Shepelev)

- [Bug [#19392](#)] Endless method vs and/or
 - Not only and/or are affected, but also trailing if/unless: [#19392#note-7](#)
 - When met with the behavior, it is very hard for users to diagnose the problem: [#19731](#) which is a serious impediment in feature usage;
 - With all due respect, I don't think that closing the issue was justified (the provided justification was apparently assuming the ticket requires change of and/or priorities, not endless method behavior);
 - After its closing, me and [@Eregon \(Benoit Daloze\)](#) provided detailed justifications about its importance ([#19392#note-10](#), [#19392#note-13](#));
 - The patch was **provided** by [@yui-knk \(Kaneko Yuichiro\)](#) [#19392#note-12](#) - <https://github.com/ruby/ruby/pull/8054>. As far as I can see, it was approved by [@nobu \(Nobuyoshi Nakada\)](#).

#13 - 11/30/2023 08:18 AM - mame (Yusuke Endoh)

[@matz \(Yukihiro Matsumoto\)](#) was absent today. We had a light discussion with the committers who were present, but will maybe hold the meeting again on 7th Dec. if matz can make it.

#14 - 12/06/2023 10:20 PM - kddnewton (Kevin Newton)

- [Bug [#20044](#)] Add runtime flag and environment variable for prism
 - We would like to add a runtime flag and environment variable to activate prism
 - We would like to mark it as experimental, so that it could be removed easily in the future
 - We would like to get this in for 3.3 so that we can test it on a stable branch in our CI
 - Is the naming/API okay?

#15 - 12/07/2023 10:47 AM - mame (Yusuke Endoh)

- Description updated

- Status changed from Open to Closed