

## Ruby - Misc #21281

### DevMeeting-2025-05-08

04/23/2025 09:56 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> 2025/05/08 13:00-17:00 (JST) Log: <a href="https://github.com/ruby/dev-meeting-log/blob/master/2025/DevMeeting-2025-05-08.md">https://github.com/ruby/dev-meeting-log/blob/master/2025/DevMeeting-2025-05-08.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 record of the discussion in the 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 of the meeting 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 (ko1) * 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 2025/05/05. 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	<b>Open</b>

## History

### #1 - 04/23/2025 09:56 AM - mame (Yusuke Endoh)

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

### #2 - 04/23/2025 11:15 AM - byroot (Jean Boussier)

- [Feature #15408] Actually deprecate ObjectSpace.\_id2ref (byroot)
  - Matz agreed to deprecate \_id2ref for Ruby 2.7 but that was never implemented.
  - [I have a PR open to finally do it](#), but should the deprecation mention the removal timeline, is there one?
  - The main user is drb but there is a PR open to solve it: <https://github.com/ruby/drb/pull/31>

### #3 - 05/01/2025 06:21 AM - tagomoris (Satoshi Tagomori)

- [Feature #21311] Namespace on read (tagomoris)
  - I'm finishing the namespace branch to be ready for merge (disabled by default)  
<https://github.com/tagomoris/ruby/tree/namespace-on-read-classex>

- Feature [#21311](#)
  - <https://github.com/ruby/ruby/pull/13226>
- Current status:
  - Default (Namespace disabled, YJIT/ZJIT disabled): make check and make exam passed on my macOS (not checked on CI)
  - Namespace enabled: make check passed but make exam failed (the known cause and I'll update the implementation soon)
- I hope it'll be merged without further more rebases :P

#### #4 - 05/03/2025 07:43 AM - jeremyevans0 (Jeremy Evans)

- [Feature [#21274](#)] Show performance warnings for easily avoidable unnecessary implicit splat allocations (jeremyevans0)
  - This warning only shows method calls that are allocating unnecessarily, where allocation can be avoided by using a local variable.
  - This warning found unnecessary allocations in the standard library, rubygems, Rails, and Sequel.
  - We are currently only using performance warnings for objects with too many shapes.
  - Is this OK to commit?
- [Feature [#21287](#)] Remove SortedSet autoload and set/sorted\_set (jeremyevans0)
  - Set is now a core class, and the SortedSet autoload that was in set.rb is now in core.
  - This autoload only works if the sorted\_set gem is installed.
  - I don't think we should have this autoload, can we remove it?
  - If so, do we want to deprecate the autoload before removing it?
- [Bug [#21302](#)] Remove or Fix Set#to\_h (jeremyevans0)
  - The method was not present in stdlib Set, and not previously approved for addition.
  - However, psych started using it.
  - The method is not backwards compatible with the previous Set#to\_h (implemented by Enumerable#to\_h).
  - I recommend we remove it and fix psych, and have a pull request for that.
  - If we want to keep it, I have a pull request to fix it, and add tests and documentation.
  - Do we want to keep Set#to\_h and have a return a true-valued hash if not given a block?
- [Bug [#21280](#)] StringIO#set\_encoding warns when backed by chilled string literal (jeremyevans0)
  - This warning is bogus, as most code using this does not need changes when string literals are frozen.
  - Issuing bogus warnings prompts for unnecessary code changes and should be avoided.
  - StringIO#set\_encoding already does not set the encoding if the string is frozen, since [\[Bug #11827\]](#)
  - I recommend we stop setting the underlying encoding for non-frozen strings to avoid the warning.
  - This could be done for only chilled strings, or for all strings if behavior should be consistent.
  - Should we make this change, and if so, for chilled strings or for both chilled and unfrozen strings?

#### #5 - 05/03/2025 06:25 PM - mame (Yusuke Endoh)

- [Feature [#21307](#)] A way to strictly validate time input (mame)
  - I want to confirm that the current tolerant behavior of Time.new is intentional, and if so, want a strict variant of Time.new.

#### #6 - 05/04/2025 01:26 AM - jhawthorn (John Hawthorn)

- [Feature [#21267](#)] respond\_to check in Class#allocate is inconsistent
  - Class#allocate does a slow respond\_to check to try to prevent uninitialized values via method(:allocate).bind\_call on MatchData, Refinement, Module, Complex, Rational
  - The respond\_to check is easy to trick and bypass. Even if it was more strict, it does not seem to provide additional safety as there are [other ways to construct uninitialized objects](#).
  - Is it OK to remove?

#### #7 - 05/07/2025 05:01 AM - hsb (Hiroshi SHIBATA)

- [Feature [#21258](#)] Retire CGI library from Ruby 3.5
  - Can I remove CGI library without CGI::Util ?
  - What's the preferred migration process?
    1. We keep only cgi/escape (C impl)feature in Ruby.
    2. We keep cgi/util with cgi/escape. The current CGI library is removed and depend cgi-util gem.
    3. We migrate cgi/escape to other class/module. The current CGI library and cgi/escape are removed.
    4. We provide cgi-util gem for migration with deprecated warning at Ruby 3.5. In next year, we will remove cgi-util gem.

#### #8 - 05/09/2025 10:31 AM - Eregon (Benoit Daloze)

- Description updated

#### #9 - 06/05/2025 11:04 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed