Hash#slice does not keep compare_by_identity on the results

03/28/2021 12:25 PM - kachick (Kenichi Kamiya)

Status: Closed
Priority: Normal
Assignee:
Target version:

ruby -v:
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin20]

Backport:
2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN

Description

$ ruby -v
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin20]

str1 = +'str'
str2 = +'str'
hash = {a: :a, b: :b, c: :c}.compare_by_identity
hash[str1] = 1
hash[str2] = 2
p hash.values_at(str1, str2) #=> [1, 2]
p hash.except.compare_by_identity? #=> true
p hash.slice.compare_by_identity? #=> false
p hash.slice(str1, str2) #=> {:a=>:a, :b=>:b, :c=>:c}
p hash.slice(str1, str2) #=> {"str"=>2}

Is this an intentional behavior?
I would expect Hash#slice keeps compare_by_identity behaviors.

Associated revisions

Revision 95f8ffa5 - 07/15/2021 05:04 PM - jeremyevans (Jeremy Evans)

Copy hash compare_by_identity setting in more cases

This makes the compare_by_identity setting always copied
for the following methods:

- except
- merge
- reject
- select
- slice
- transform_values

Some of these methods did not copy the setting, or only
copied the setting if the receiver was not empty.

Fixes [Bug #17757]
Co-authored-by: Kenichi Kamiya kachick1@gmail.com

History

#1 - 03/28/2021 07:33 PM - marcandre (Marc-Andre Lafortune)

I should have looked into this when I checked #16996.

Looks like we have behavior that is all over the place, and the same method sometimes returns a hash with different comparability depending if the
receiver is empty or not...

ruby 3.1.0dev (2021-03-28T18:18:08Z master lcodecb4349) [x86_64-darwin18]
slice with arg: false
slice without arg: false
except with arg: true but false for {}
extcept without arg: true but false for {}
dup: true
select: true but false for {} # in 3.0: false
reject: true but false for {} # in 3.0: false
merge: true but false for {}
transform_values: true but false for {}
transform_keys: false
to_h: false
to_h without block: true

Script:

def check(method, label = method, *args, &block)
  ci1, ci2 = [{a: 1}, {}].map do |h|
    h.compare_by_identity
  h2 = h.send(method, &block)
  h2.compare_by_identity?
  end
  s = "#{label}: #{ci1}"
  s << " but #{ci2} for {}" if ci2 != ci1
  puts s
end

check(:slice, 'slice with arg', :a)
check(:slice, 'slice without arg')
check(:except, 'except with arg', :a)
check(:except, 'except without arg')
%i[dup select reject merge transform_values transform_keys to_h].each do |method|
  check(method) { |k, v| [k, v] }
end
check(:to_h, 'to_h without block')

It's probably clear that the result should always be consistent for empty hash or not.

I feel the flag should be conserved for all these examples, except to_h […] which should create an array without comparison by identity (as it does currently).

#2 - 07/01/2021 06:00 PM - jeremyevans0 (Jeremy Evans)
I've submitted a pull request that builds on kachick (Kenichi Kamiya)’s PR and fixes the remaining cases found by marcandre (Marc-Andre Lafortune): https://github.com/ruby/ruby/pull/4616

#3 - 07/01/2021 08:16 PM - marcandre (Marc-Andre Lafortune)
Thanks for the PR jeremyevans0 (Jeremy Evans), looks pretty good.

On the other hand, I'd like to revise what I wrote and suggest that transform_keys remain untouched. It's the only one of these methods (with to_h[...]) that changes the keys so has a reason to interpret them differently.

Matz, do you agree?

#4 - 07/15/2021 05:55 AM - matz (Yukihiro Matsumoto)
Yes, we must keep compare_by_identity status for those methods. Accepted.

Matz.

#5 - 07/15/2021 05:05 PM - jeremyevans (Jeremy Evans)
- Status changed from Open to Closed

Copied hash compare_by_identity setting in more cases

This makes the compare_by_identity setting always copied for the following methods:
- except
- merge
- reject
- select
- slice
- transform_values

08/06/2021
Some of these methods did not copy the setting, or only copied the setting if the receiver was not empty.

Fixes [Bug #17757]

Co-authored-by: Kenichi Kamiya kachick1@gmail.com