Remove verbose warning on treating keyword splat as positional argument in Ruby 2.6

02/13/2020 09:55 PM - jeremyevans0 (Jeremy Evans)

Status: Closed
Priority: Normal
Assignee:
Target version:
ruby -v:
ruby 2.6.5p114 (2019-10-01 revision 67812) [x86_64-openbsd]
Backport:
2.5: DONTNEED, 2.6: REQUIRED, 2.7: DONTNEED

Description
Ruby 2.6 will issue a verbose mode warning if treating a keyword splat as a positional argument if the method does not accept rest arguments or keyword arguments. This is different from Ruby 2.5, 2.7, and the master branch.

```ruby
$ ruby25 -ve 'def a(b); b end; hash = {bar: 1}; p a(**hash)'
```

```
ruby 2.5.7p206 (2019-10-01 revision 67816) [x86_64-openbsd]
{:bar=>1}
```

```ruby
$ ruby26 -ve 'def a(b); b end; hash = {bar: 1}; p a(**hash)'
```

```
ruby 2.6.5p114 (2019-10-01 revision 67812) [x86_64-openbsd]
-e:1: warning: in `a': the last argument was passed as a single Hash
-e:1: warning: although a splat keyword arguments here
{:bar=>1}
```

```ruby
$ ruby27 -ve 'def a(b); b end; hash = {bar: 1}; p a(**hash)'
```

```
ruby 2.7.0p0 (2019-12-25 revision 647ee6f091) [x86_64-openbsd]
{:bar=>1}
```

My guess for the reason for the Ruby 2.6 behavior is it was done before we agreed on the semantics of keyword argument separation. If we were going to drop keyword argument to positional argument conversion for methods that did not accept keyword arguments, it is a reasonable warning. However, since we convert keyword arguments to positional arguments for methods that do not accept keywords, the 2.6 verbose mode warning doesn't make sense and I think should be removed. See attached diff for the removal of the warning and updating of related test.

History

#1 - 02/19/2020 07:31 PM - Dan0042 (Daniel DeLorme)
This is really interesting. It means that mixing double-splat with positional-only methods was already discouraged.

It feels kinda weird to be more strict about keyword/positional separation in general, but to become more more lenient for this case. Another possibility would be to re-add the warning in 2.7 and trunk. The semantics of keyword-to-positional conversion were basically reverse engineered from the 2.6 behavior, so we could say the correct behavior is "however it worked in 2.6" and this part was simply overlooked.

Also I find this gives an interesting alternative view on keyword arguments: the braceless hash syntax can be used either for a positional hash or keywords, but a double splat is specifically for keywords and shouldn't be used with a positional-only method.

I don't really have a strong opinion either way.

#2 - 02/21/2020 09:39 AM - Eregon (Benoit Daloze)
As mentioned in https://bugs.ruby-lang.org/issues/16511#note-17 it seems very unlikely this triggers in practice, because indeed who would bother with an extra ** when the method doesn't take keyword arguments? Maybe for the case the user thinks the method takes keyword arguments but it doesn't (yet, maybe the next gem version does)?

Anyway, agreed with Jeremy, no point to warn about something which no longer warns in more recent versions. Warning ** but not foo: 1 as a special case seems to just make things more complicated.

#3 - 02/26/2020 01:11 AM - jeremyevans0 (Jeremy Evans)
Looking through the repository history, the patch attached to this issue is mostly implemented by 6424d316b993ecceb6f583ae476096274e304788, so that commit could be backported instead, though I'm not sure it applies without conflicts.

Files

remove-verbose-warning-keyword-splat-to-positional.patch 2.1 KB 02/13/2020 jeremyevans0 (Jeremy Evans)