

Ruby - Bug #16111

Combining default positional and keyword arguments is resulting in a `to_hash` method call

08/19/2019 05:47 AM - mtelford (Michael Telford)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	2.6.3	Backport: 2.5: UNKNOWN, 2.6: UNKNOWN
Description <p>Combining default positional and keyword arguments is resulting in a <code>to_hash</code> method call when I don't believe it should be. It's affecting some code I am writing. The full details of the behaviour I'm describing can be seen in the attached script.</p> <p>I would like to know why the method <code>to_hash</code> is being called (only in certain circumstances) and if this is a bug to be fixed or desired behaviour (and why?).</p> <p>Thank in advance!</p>		

History

#1 - 08/19/2019 06:13 AM - mame (Yusuke Endoh)

I would like to know why the method `to_hash` is being called

A method call that accepts keywords checks if the last argument is a Hash or not. The check includes attempting to convert it by using `to_hash`. This is the behavior you are seeing.

I agree that it is very strange. It is elegantly solved if we can separate positional and keyword arguments completely ([#14183](#)), but the separation seems to be difficult because the incompatibility is painful.

#2 - 08/19/2019 08:22 AM - mtelford (Michael Telford)

Ok, thanks for the reply. That now makes more sense. I understand why the last positional argument is type-checked but surely trying to convert it using `param#to_hash` is a step too far? Doing so basically means that the last positional argument cannot respond to `:to_hash`, as in my case. It's not ideal as my problem still exists.

#3 - 08/19/2019 04:45 PM - shevegen (Robert A. Heiler)

I think that either way, resolving it may not be trivial/simple right now; if you look at mame's linked in issue above at [#14183](#), it mentions 19 (somewhat) related issues concerning keyword arguments. Jeremy also pointed out some bugs prior to this, and if I remember correctly, matz spoke about the somewhat confusing situation in a presentation a while ago. This may need to be given some time, for the time being, but the ruby core team is most likely aware of the situation.

I personally have avoided keyword arguments so far (but unrelated to the issues mentioned in the issue trackers; I mostly stick to oldschool hashes. Perhaps this may also be somewhat of a "historic" nature, in that hash-behaviour came first (the old pickaxe mentioned that, e. g. you could drop the `{}` syntax for the last arguments in a method if it was a hash), and at a later time keyword arguments were added.

If I remember correctly then matz has also mention that the situation with keyword arguments may be one of the very very few incompatibilities for ruby 3.0 (perhaps), so I guess it is not out of the question to see a change in behaviour in this regard. As mame has pointed out, one of the biggest concern, if not the biggest, is the incompatibility issue.

#4 - 08/20/2019 02:26 PM - mtelford (Michael Telford)

Ok, the situation with keyword parameters is now making more sense, despite not being ideal. Since ruby 3 should implement a fix (backwards compatible or not) I am happy for this issue to be closed.

Thanks to those who commented to bring me up to speed :-)

#5 - 08/20/2019 04:26 PM - jeremyevans0 (Jeremy Evans)
- Status changed from Open to Closed

Files

ruby-bug.rb	1.44 KB	08/19/2019	mtelford (Michael Telford)
-------------	---------	------------	----------------------------