

Ruby - Feature #15896

Symbol#+

06/03/2019 04:47 PM - zeus (Zeus 81)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	
Description I've made this today <pre>class Symbol @@cache_add = Hash.new { h1,k1 h1[k1] = Hash.new { h2,k2 h2[k2] = : "#{k1}#{k2}" }} def +(s) @@cache_add[self][s] end end :abc + :def # => :abcdef</pre> I thought it was so awesome it should become standard.	

History

#1 - 06/03/2019 05:26 PM - Eregon (Benoit Daloze)

FWIW, there is no need for a cache, since Symbols are interned (and an explicit cache would leak memory).

I think Symbols are generally different than Strings, so #+ feels a bit out of place for me here.
Can you show a realistic use case?

#2 - 06/03/2019 05:34 PM - jeremyevans0 (Jeremy Evans)

I am against adding this to core. Symbols represent identifiers, not data/text. Strings should be used for data/text. I don't think we should add methods to Symbol that further blur the line between String and Symbol.

I would rather see :#{s1}#{s2}" for explicitly combining two symbols in the few cases that need to do so. That would be slower than the cached approach as it currently results in 3 string allocations, but I don't think it is likely to be a bottleneck.

One issue with the cached approach you used is that the cache doesn't use a weak map, so dynamic symbols that use + would never get garbage collected.

#3 - 06/03/2019 06:43 PM - shevegen (Robert A. Heiler)

I agree with benoit and jeremy, also in regards to the use case description; even speed/cache considerations aside, matz possibly may consider the proposal to not fit into/to class Symbol by default. It is actually already awesome that class Symbol can be modified like this :) - I am not sure if it is required to add '+' to class Symbol as a method, by default. I can understand it somewhat to some extent (people love Symbols ever since the old pickaxe, I guess), but I think what benoit wrote is quite true - it may be better to be conservative here and keep Symbols and Strings distinct as separate entities. (See also what matz wrote about the origin of Symbols for ruby).

#4 - 07/29/2019 07:44 AM - ko1 (Koichi Sasada)

- Status changed from Open to Rejected

Ut seems difficult to introduce.
Please reopen it with your opinion if it should be discussed more.