Ruby - Bug #7664

Keyword arguments with `**` make code targeting 1.9 / 2.0 difficult

01/07/2013 07:29 AM - wycats (Yehuda Katz)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	2.0.0dev	**Backport:**	

Description

You could imagine writing code that calls into a user-supplied hook:

log("App booted", level: :info)

It is possible for both Ruby 1.9 and Ruby 2.0 users to write such a hook:

Ruby 1.9

def log(string, options={})
level = options[:level]
puts "#{level}: #{string}"
end

Ruby 2.0

def log(string, level: nil)
puts "#{level}: #{string}"
end

So far so good. It's also possible for Ruby 2.0 users to handle arbitrary arguments:

Ruby 2.0

def log(string, **options)
pass options along to other methods without explicit keyword args support
end

However, it is not possible to *call into* methods with arbitrary arguments in a compatible way:

Ruby 1.9

args = [string] + [options]
log(*args)

Ruby 2.0

log(string, **options)

Unless I'm missing something, this makes it impossible to write code that targets both Ruby 1.9 and Ruby 2.0 if you need to send arbitrary keyword arguments to a method.

Also, because ** is new syntax, the only way to handle this problem is via eval:

if RUBY_VERSION > "2"
eval "log(string, **options)", **FILE**, **LINE**
else
args = [string] + [options]
log(*args)
end

This can work if you use class_eval to create two different versions of the method at class creation time, but it's pretty ugly even in that case.

Is there a way to make it possible to have a backwards-compatible calling signature that can target both the Ruby 1.9 and Ruby 2.0 method signatures for doing keyword arguments?

History

#1 - 01/07/2013 10:03 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

I understand your concern. But I consider this as a driving force to jump in to 2.0.
Since we have small compatibility issues between 1.9 and 2.0, I'd suggest moving forward.

Matz.

#2 - 01/07/2013 11:52 AM - nobu (Nobuyoshi Nakada)

=begin
Have you tried this in 2.0 actually?

args = [string] + [options]
log(*args)
=end