

## Ruby master - Bug #6434

### Block passed implicitly via super

05/15/2012 02:58 AM - trans (Thomas Sawyer)

<b>Status:</b> Rejected	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b> 1.9.3	
<b>ruby -v:</b> ruby 1.9.3p125 (2012-02-16 revision 34643) [x86_64-linux]	<b>Backport:</b>
<b>Description</b> =begin Block is being passed via super implicitly even though the method is explicit and does not pass the block when calling super.  <pre>class H &lt; Hash   def initialize(default=nil, &amp;block)     super(default)   end end H.new(0){}</pre> ArgumentError: wrong number of arguments from (pry):3:in `initialize'  Hash#initialize doesn't allow both a default argument and a default_proc, which is the cause of this error. Which means the block is being passed up even though it should not be. =end	

### History

#### #1 - 05/15/2012 02:59 AM - trans (Thomas Sawyer)

Crap, could an admin please fix the use of RD in my post. And while I am on the subject, can we just make RD mode always on and be done with it?

#### #2 - 05/15/2012 08:22 AM - shyouhei (Shyouhei Urabe)

- Description updated

#### #3 - 05/15/2012 09:56 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

=begin  
It's a spec.

((super)) inherits the given block if it's not given directly.

You can call ((super(default, &nil))) not to pass it.  
=end

#### #4 - 05/15/2012 01:36 PM - trans (Thomas Sawyer)

I see. It's not intuitive, obviously, but I suppose it makes sense for how Ruby handles implicit yield in general. Nonetheless, wouldn't it be better if this did not apply when a explicit block argument is used?

```
# would pass block implicitly
def foo
  super
end
```

```
# would NOT pass block implicitly
def foo(&block)
  super
end
```

In any case I discovered the use of &nil as a work around prior to posting this and that does the trick. Thanks.

**#5 - 05/15/2012 05:04 PM - alexeymuranov (Alexey Muranov)**

Just a note: super without arguments is documented to pass all arguments, so to not pass a block can only be expected from super()

**#6 - 05/15/2012 10:26 PM - trans (Thomas Sawyer)**

@alexey I thought that was no longer true for 1.9.

**#7 - 05/16/2012 03:49 AM - alexeymuranov (Alexey Muranov)**

[thomas \(Thomas Fritzsche\)](#), i didn't know it could change. In fact, i do not know where to find an up to date documentation about basic Ruby keywords.