

## Ruby 1.8 - Bug #992

### Hash becomes Array for no apparent reason.

01/09/2009 10:31 AM - loqi (Loqi Tamaroon)

<b>Status:</b>	Rejected
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	Ruby 1.8.6
<b>ruby -v:</b>	
<b>Description</b>	
<pre>=begin # Using Ruby 1.8.6 (2008-03-03 patchlevel 114) [universal-darwin9.0] # on Mac OS X 10.5.5 def is_this_a_ruby_bug?   orig_hash = { :a=&gt;[:x], :b=&gt;[:x] }   orig_hash.inject({}) do  flipped_hash, (symbol_a_or_b, array_x)       array_x.each do  symbol_x        (flipped_hash[symbol_x]   = []) &lt;&lt; symbol_a_or_b # (~1)     flipped_hash       # (~2)     end       # (~3)     # flipped_hash       # (~4)     end   end end is_this_a_ruby_bug? # The above method is supposed to return { :x =&gt; [:a, :b] } . Instead, it exhibits # bizarre behavior. On the first iteration of the inner loop, line (~2) somehow causes # flipped_hash to be recast from Hash {:x=&gt;[:a]} to Array [:x] . This causes (~1) to # raise a "Symbol as array index (TypeError)" exception on the second iteration of # array_x.each , because flipped_hash has accedentally become an Array. A workaround # is to uncomment (~4), which curiously is OUTSIDE the inner loop where the exception # would've ocured. The effect is that the 'end' at (~3) no longer terminates the inner # loop and (~4) is now executed with each iteration of array_x.each as if it were above # (~3). Somehow this fixes the problem, and the method returns the expected result. =end</pre>	

### History

#### #1 - 01/09/2009 11:07 AM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Rejected

```
=begin
No it isn't. Your "workaround" is the canonical way. See whatever documentation you like that describes inject method.
=end
```

#### #2 - 01/09/2009 11:19 AM - mernen (Daniel Luz)

```
=begin
Since Enumerable#each ignores the return value of the block, (~2) is a no-op. #each returns the original object (in this case, array_x), and as such
you're passing the second iteration of #inject array_x, not flipped_hash. (~4) is the correct position to return a value here.
=end
```

#### #3 - 01/09/2009 11:42 AM - loqi (Loqi Tamaroon)

```
=begin
Hey thanks Daniel. I missed that detail about #each . I'm new to Ruby.
=end
```