

Ruby master - Bug #14355

clone issue

01/14/2018 07:27 PM - Winklerbugs (Bart Winkler)

Status: Rejected	
Priority: Normal	
Assignee:	
Target version:	
ruby -v:	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

```
#!/usr/bin/env ruby
aMutantSpeak = nil,nil],[nil,nil
aDogSpeak = aMutantSpeak.clone
aCatSpeak = aMutantSpeak.clone
aDogSpeak[0][0] = "W"
aCatSpeak[0][0] = "m"
aCatSpeak[0][1] = "e"
aDogSpeak[0][1] = "O"
aCatSpeak[1][0] = "o"
aDogSpeak[1][0] = "O"
aDogSpeak[1][1] = "F"
aCatSpeak[1][1] = "w"
print "aMutantSpeak ", aMutantSpeak
puts
print "aDogSpeak ", aDogSpeak
puts
print "aCatSpeak ", aCatSpeak

#all three arrays are now "m","O"],["O","w"
```

This is what I wanted:

aMutantSpeak = nil,nil],[nil,nil

aDogSpeak = "W","O"],["O","F"

aCatSpeak = "m","e"],["o","w"

I think this is what clone should do.

History

#1 - 01/14/2018 08:12 PM - jeremyevans0 (Jeremy Evans)

- ruby -v deleted (1.9.3 & 2.2.2)
- Status changed from Open to Rejected

As documented, Object#clone and Object#dup do shallow copies by default, so this behavior is expected. Create your own class and override initialize_copy (or initialize_dup and initialize_clone) as appropriate for the behavior you want.

One quick alternative for making a deep copy is using Marshal.load(Marshal.dump(obj)), but note that not all objects can be marshalled.

Files

MutantSpeak.rb	402 Bytes	01/14/2018	Winklerbugs (Bart Winkler)
----------------	-----------	------------	----------------------------