

Redmine - Defect #4533

ARCondition not handling dup/clone

2010-01-07 12:39 - Felix Schäfer

Status:	Closed	Start date:	2010-01-07
Priority:	High	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:	Wont fix		
Description			
Long story short: ARCondition#clone and #dup create new objects, but all objects created this way somehow share the string in @conditions[0].			
<pre>s = ARCondition.new "#{Issue.table_name}.assigned_to_id IS NOT NULL" p = s.clone s << ["#{Issue.table_name}.start_date <= ?", Date.today] p</pre>			
Results in:			
<pre>\$ script/console Loading development environment (Rails 2.3.5) >> s = ARCondition.new "#{Issue.table_name}.assigned_to_id IS NOT NULL" => #<ARCondition:0x102bbb040 @conditions=["1=1 AND (issues.assigned_to_id IS NOT NULL)"]> >> p = s.clone => #<ARCondition:0x102baac68 @conditions=["1=1 AND (issues.assigned_to_id IS NOT NULL)"]> >> s << ["#{Issue.table_name}.start_date <= ?", Date.today] => #<ARCondition:0x102bbb040 @conditions=["1=1 AND (issues.assigned_to_id IS NOT NULL) AND (issues.start_date <= ?)", Thu, 07 Jan 2010]> >> p => #<ARCondition:0x102baac68 @conditions=["1=1 AND (issues.assigned_to_id IS NOT NULL) AND (issues.start_date <= ?)"]></pre>			
Notice p got the condition added to s but not the parameter.			
I was able to get around this by being a little more brutal, but I don't think that's how it should work:			
<pre>p = ARCondition.new s.conditions</pre>			

History

#1 - 2010-06-14 21:07 - Jean-Baptiste Barth

It seems that by default ruby just gives you a shallow copy of your objects when calling #dup or #clone. Underlying objects are just a reference to original ones. If you want deep copies, many people advice to marshal your object and then restore it, which is not cleaner than your solution in my opinion.

A solution would be to rely on ruby hooks called when #dup/clone are called:

```
def initialize_copy(source)
  super
  @conditions = source.conditions.clone
  @conditions[0] = source.conditions[0].clone
end
```

With this added in ARCondition class, ARCondition objects are really independant...

But maybe it's simpler to keep the first solution you exposed, no ?

#2 - 2013-04-12 04:34 - Filou Centrinov

"ARCondition" class ist not anymore part of Redmine. Close this issue?

#3 - 2013-04-12 09:01 - Etienne Massip

- *Status changed from New to Closed*
- *Resolution set to Wont fix*

Indeed, the class has been removed with [r8089](#).