

Foreman - Bug #1027

Cached methods are left intact during orchestration

07/06/2011 04:50 AM - Paul Kelly

Status: Resolved	
Priority: Normal	
Assignee: Paul Kelly	
Category: Rails	
Target version:	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request:	
Description	
<p>The orchestration process injects itself into the AR lifecycle. During an update the host object's attributes are modified, a copy is then taken and then the callbacks use self and self.old for further processing</p> <p>However if an association has already been accessed then AR caches this in the object and that can mean that later references to self.operatingsystem refer to the cached value. I.E. the value before the host's attributes were changed.</p> <p>The same is true for the old object. To avoid this, the setup_clone operation should ensure that any attribute that was changed and that is an association_id, be used to reload the AR association.</p> <p>Ohad,</p> <p>can you think of any possible side effects? I am also unclear as to how this could have gone undetected for so long. It is easy to see this occur. Just change a host's operatingsystem and breakpoint in setup_clone. You will see that self.operatingsystem does not match Operatingsystem.find @attributes["operatingsystem_id"]</p>	

History

#1 - 07/06/2011 11:23 AM - Paul Kelly

- Status changed from New to Ready For Testing

This workaround is not nice and if I had the time to resolve it properly I would.
It is slightly better than the first hack

#2 - 03/15/2016 03:53 AM - Ohad Levy

- Description updated

- Status changed from Ready For Testing to Resolved

I can't seem to find any reference to an actual bug over the years, closing this as it seems either non relevant usage case, or not a bug.