

Kepler - Bug #2572

Export KAR can produce several actors with the same Isid

10/26/2006 12:13 PM - Norbert Podhorszki

Status:	Resolved	Start date:	10/26/2006
Priority:	Normal	Due date:	
Assignee:	Chad Berkley	% Done:	0%
Category:	actors	Estimated time:	0.00 hour
Target version:	1.0.0rc1	Spent time:	0.00 hour
Bugzilla-Id:	2572		

Description

Start new workflow
Create two composite actors (put something into them)
Export them as KARs (one-by-one into two KARs)
--> they get different Isid in the KAR (e.g. 1 and 2), fine
Modify the first one and export again
--> it gets the second's (last) Isid (2)
Copy/paste any of them, modify the new composite and export it
--> it still gets the last used Isid (2)

New Isid is generated, apparently, if you drag new actors on canvas. Just modifying the existing composites, even copy/paste them does not help.

History

#1 - 06/15/2007 12:27 PM - Chad Berkley

This is a problem with how we store Isids in the cache. The problem is that the generated kar file does not get cached when it is created (although it could be cached later). If we store the Isid in the database upon kar creation, it will have to get a new Isid when it is inserted into the cache (because there is no way to know that the object being put into the cache is the same as the one that was created earlier since we lose control of it). Another way around this is to ask the user if they wish to get a unique Isid or not.

The heart of this is that we still need a centralized way to keep track of Isids. We are still dependent on the local machine to generate ids, which in no way guarantees they are unique. To really fix a lot of these ID issues, we need a central ID repository (or we need to use library.kepler-project.org as one).

#2 - 01/24/2008 01:20 PM - Chad Berkley

I fixed this bug by detecting the Isid collision and asking the user if he/she would like to change the id of the component they are importing. This is the best solution I could come up with until we further discuss the topics I brought up in comment [#1](#). I also fixed several other bugs that came up in the process of figuring this out. I think a couple of them are probably responsible for the corrupted cache that we would see every-once-in-a-while.

#3 - 03/27/2013 02:20 PM - Redmine Admin

Original Bugzilla ID was 2572