Kepler - Bug #4795

RExpression & cache cleaning

02/12/2010 04:15 PM - Oliver Soong

Status:	New	**Start date:**	02/12/2010
Priority:	Normal	**Due date:**	
Assignee:	Chad Berkley	**% Done:**	0%
Category:	core	**Estimated time:**	0.00 hour
Target version:	Unspecified	**Spent time:**	0.00 hour
Bugzilla-Id:	4795		

Description

My .kepler cache bloats pretty quickly because of RExpression's temporary files. Can we have RExpression clear it's cache folder on initialize? This way, if we need to inspect those temporary files after Kepler closes, we still can, but we'll inhibit cache bloat. I suggest doing this automatically because, while I might know what's safe to delete, I've been operating under the assumption that end users aren't expected to learn the internal structure and dependencies of .kepler.

History

#1 - 02/12/2010 04:22 PM - Matt Jones

I've noticed this too, and it seems like a good idea to me.

If the RExpression actor put these files into the provenance DB, then they 1) wouldn't need to be on disk to be cleaned up, and 2) would be part of kar run archives when they are saved. Deleting runs would then also delete the provenance record, which is enabled with a UI, so it would be more intuitive for users. Do we already do this with RExpression to enable including them in reports? Can we eliminate the temporary storage altogether, or just delete the cached files from disk as Oliver indicates? Does ImageJ and other actors use the disk version to display the files, and if so, could it be made to use the provenance db version?

In general I think its better to write results to provenance and then be able to access those results from provenance. This centralizes result managment, and avoids the messy actor-by-actor management of outputs. Thoughts?

#2 - 02/12/2010 04:36 PM - ben leinfelder

There are a few kinds of files generated by RExpression:
1. complex data transfer from one RExpression actor to another
2. temporary files for long script input (not complex data, just long data)
3. graphics output (whether used for reporting or not)

The first 2 are usually not needed unless you're doing some hardcore debugging. The graphics could be saved in provenance on their own, but I think that is redundant when reporting is enabled since the images are grabbed and put in provenance as tokens for use in the reports.

Basically anything here could be wiped clean on initialize, and you'd only be able to access the last one that ran (before it was blown away):
/Users/leinfelder/.kepler/cache-2.0.0/modules/r/tpc09-plant-dynamics-woody_1266009177777

#3 - 02/12/2010 05:14 PM - Oliver Soong

We could do away with the .sav files. Ben, try this in R:

```
str <- paste("_data.frame_:", rawToChar(serialize(iris, NULL, TRUE)), sep = "")
data <- unserialize(charToRaw(substr(str, 14, nchar(str))))
```

I'd want to make sure that escaping is done properly and we might bump into string length problems.

#4 - 02/12/2010 05:20 PM - ben leinfelder

Dan H. already had some code in there for managing "too long of strings" for the script input - long numeric arrays can get long.

So you're saying just pass a huge StringToken with the dataframe (or other obj) in it?

I would be concerned with string length and memory. But could be cool!

#5 - 02/12/2010 05:29 PM - Oliver Soong

Zactly.

#6 - 03/27/2013 02:28 PM - Redmine Admin

Original Bugzilla ID was 4795