

Kepler - Bug #4320

Component Library Folders and Popup Menus

08/17/2009 12:09 PM - Aaron Aaron

| | | | |
|------------------------|-------------|------------------------|------------|
| Status: | Resolved | Start date: | 08/17/2009 |
| Priority: | Immediate | Due date: | |
| Assignee: | Aaron Aaron | % Done: | 0% |
| Category: | core | Estimated time: | 0.00 hour |
| Target version: | 2.X.Y | Spent time: | 0.00 hour |
| Bugzilla-Id: | 4320 | | |

Description

The Component Library is the most used graphical component in Kepler. It is in essence the "Control Panel" for building workflows. Because of this it is important to give the user as much control over this area as possible.

To do this a mechanism for viewing components as they appear on disk is needed. This will allow users to organize their components however they want in folders while still providing the ability to index the components within the Kepler ontologies.

Popup menus allowing the user to do basic control functions like "Move, Remove (from ontology), Delete (from disk), Create New, View LSID, View Revisions, View Documentation, Open, Annotate, etc..." should be added to the components and their containers (whether the container is an ontology/concept or a folder/kar). The popup menus are displayed from AnnotatedPTree.maybeShowPopup using subclasses of the LibraryPopup class.

History

#1 - 08/18/2009 07:45 AM - Christopher Brooks

I think this is a feature request, not a bug.
A bug is behavior of a currently implemented feature that does not work as expected.
This proposed feature looks interesting, but I question whether it should be done before 2.0.0

#2 - 08/18/2009 07:57 AM - Shawn Bowers

I think this also assumes components are fundamentally stored on disk, and if stored on disk, in a meaningful place. One could easily imagine storing actors, e.g., within an underlying DBMS, similar to the Kepler provenance framework or MyExperiment. This would make a UI based on a physical directory storage structure awkward and unnecessary. (A DBMS-based approach for storing Kepler items would also simplify much of the application.) Similarly, if the system is managing where components are stored on disk, exposing this information may not be that useful (e.g., if these are within the .kepler folder, or as in pre-installed kars).

#3 - 08/18/2009 10:43 AM - Timothy McPhillips

Addressing Christopher's comment, I think what this makes this entry "bug-ish" is that the Component Library looks like a nested folder hierarchy. In many other software products the user can reorganize objects displayed in such a widget, rename folders, add new folders, etc, so it "feels" like one should be able to do this in Kepler as well. The components in our Library even can be dragged around (because they are meant to be dragged onto the canvas), but you cannot drop a component in a different "folder".

I would love to be able to quickly organize components the way I like as Aaron suggests.

#4 - 08/18/2009 12:06 PM - Aaron Aaron

Shawn, Kepler's library is now built from KAR files stored on disk in folders referred to as "Local Repositories". This was specifically done to get away from storing things in the RDBMS as was done in Kepler 1.0. See bug 3898. By storing things in an RDBMS (and nowhere else) the complexity is greatly increased because of the need to transition all that data to new versions of the software. It also becomes much more of a black box to the user as to what is going on and has an impact on transferring data between Kepler instances. It is also awkward for many different modules to contribute to an RDBMS structure intelligently. Using KAR files on disk as the main data store for user generated content and using the RDBMS as purely an indexing mechanism for java serialized copies of the KAR contents (to increase performance) is what we should be shooting for in the 2.0 release of the core modules (and really that's where we're at now, excepting some outstanding issues).

#5 - 08/26/2009 04:18 PM - Matt Jones

Kepler used to have at least some of these functions, such as the ability to delete a component from the Library. See bug # 4047 for an example of delete functionality that used to work. Maybe some of the earlier code can be reincarnated to fix these regression bugs.

#6 - 08/26/2009 04:26 PM - Aaron Aaron

Look, it's different now, if a delete function worked when everything was stored in the database then it's probably not going to work now.

#7 - 11/24/2009 09:58 AM - Aaron Aaron

Subclasses of the LibraryPopup class can now be used to add right click menu functionality to items in the component library in concert with AnnotatedPTree.maybeShowPopup().

Closing this bug, new bugs can be opened for more specific issues related to this system.

#8 - 03/27/2013 02:26 PM - Redmine Admin

Original Bugzilla ID was 4320