

Kepler - Bug #4740

Create Ontology_Catalog sql table in the CORE persistent database

02/04/2010 08:56 PM - Aaron Aaron

Status:	New	Start date:	02/04/2010
Priority:	Normal	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:	4740		

Description

I ran into our old friend, the "Cannot Find Ontology" error. After much time trying to debug through the OntologyConfiguration and OntologyCatalog classes I discovered that the way the owl files are being accessed is very fragile and thus causes this error (which I was finally able to reproduce by ant clean-cache). The issue is how the absolute paths are being created to find the owl files. I would suggest we create a database table in the persistent core database that has four columns: ID, NAME, PATH, and LIBRARY. The PATH is the fully qualified absolute path to the OWL file on the system that Kepler is running on (so we don't have to keep trying to figure out what the absolute path is in the Java code every time we want to get to the owl file).

The new catalog system would work like this:

- 1.) Default owl files (that are shipped with Kepler) are defined in Java code and inserted into the table the first time Kepler is started using the fully qualified absolute path for whatever system Kepler is running on
- 2.) Tagging (or other modules) can then add ontologies to the sql table (thus not having to overwrite the ontology_catalog.xml file directly as is done now)
- 3.) The user can then edit the the catalog through a gui interface (as opposed to editing the ontology_catalog.xml file as is done now)

History

#1 - 02/04/2010 09:15 PM - Shawn Bowers

I think the work Sean Riddle has done in the tagging module could help with this since he developed a gui to dynamically load and manage these ontology files.

#2 - 02/10/2010 03:11 PM - Aaron Aaron

push to 2.X.Y

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

Original Bugzilla ID was 4740