

Kepler - Bug #3947

Create a module manager

04/06/2009 11:46 AM - Chad Berkley

Status:	Resolved	Start date:	04/06/2009
Priority:	Immediate	Due date:	
Assignee:	David Welker	% Done:	0%
Category:	core	Estimated time:	0.00 hour
Target version:	2.0.0	Spent time:	0.00 hour
Bugzilla-Id:	3947		
Description			
<p>Future releases of Kepler will consist of only "standard" modules which include only the basic functionality of kepler. Further functionality will be added on in a dynamic plugin-type fashion. This will be managed within kepler by the module manager. The module manager will be responsible for the following:</p> <ul style="list-style-type: none">• Allowing a user to identify a module that needs to be loaded into the system. Modules should be able to reside locally on the users machine, or in the Kepler repository on the web.• Dynamically loading the resources within the module and managing any errors• Allowing the user to remove a module from the system• Allowing the user to upgrade a module to a new version• This functionality should be integrated into the kepler UI. Ideally, the UI would show a list of available modules with a short description that are available on the repository, or allow the user to browse for a module on the local drive.• Encapsulating a module into a kar file would be nice since kepler already uses kar files to load other types of resources. This is not a strict requirement, however.• A module should contain metadata on the creator, version, version(s) of kepler it is compatible with, etc. The module manager should not allow the loading of modules onto a version of kepler with which the module is not compatible. <p>This is an initial list of possible functionality. Please add other ideas, options to this bug.</p>			
Related issues:			
Blocked by Kepler - Bug #4009: kepler 2.0 must provide everything needed for ...		Resolved	04/21/2009

History

#1 - 04/09/2009 03:20 PM - Chad Berkley

Need to deal with caching issues with switching modules. Some objects should only be used with one module or suite and should not load when another module/suite is active. This could be achieved by tagging cache items that should only be used with a specific module. items not tagged can be loaded no matter which module is active, but tagged items are only loaded if its module is active.

#2 - 04/09/2009 04:02 PM - David Welker

Here is the current status of the module manager.

It is currently under the tools menu. It allows scientists or developers to retrieve published modules (which are located at <https://code.kepler-project.org/code/kepler/tags/published/modules/>). There are two panes. In the left pane there is a tree in which modules and suites can be selected. The right pane is essentially a virtual modules.txt file. By default, only published suites, which are a set of modules that a particular development team intends to work together are visible. Multiple suites can be selected and ordered, and they will be merged just as they would be if multiple suites were in a modules.txt file. Clearly, selecting multiple suites is adventurous and in general no guarantee of compatibility can be made. Only individually published suites (which may themselves referenced in one or more other suites) are known and certified by a particular development team to be compatible. The most common use case for a scientist will be to select a single published suite with perhaps zero or more other modules that on feels are likely compatible with that suite (perhaps that other module is nothing more than a suite of actors).

By default, individual published modules are not visible in the tree on the right. However, clicking the modules check box renders individual modules visible. They can be moved to the right hand pane just as suites can (when suites are moved to the right, they appear with a * in front of their name. When individual modules are moved, they appear without a *. Priorities are in play too, with suites or modules higher in the list taking priority over suites and modules lower in the list. Overall, just like a modules.txt file.

However, with an appropriate caveat. When you select a particular suite or module, it is the latest published revision that is retrieved and used, not the trunk version of that suite. Further, you can control precisely which revisions are used by clicking on the revisions check box, which will make revisions visible. In that case, you can either select a particular revision, or the latest revision (by selecting the module with the highest revision number or the unqualified root).

Before clicking on restart, you can see precisely how a particular selection of modules and suites will be resolved by clicking on preview. When you finally do click on restart, all of the modules (which are zipped) are retrieved using https and unzipped. When these processes take a noticeable amount of time, status bars will appear letting you know what is occurring.) After the appropriate modules are retrieved and unzipped, Kepler will

automatically restart itself and you will have the functionality of those modules (assuming compatibility between the modules).

#3 - 04/21/2009 10:32 AM - Christopher Brooks

I just tried Tools -> Module Manager, and it is pretty slick!
Is there anyway to avoid downloading modules?
Ideally, I'd like to be able to use the code in the svn trunk.
Consider the use case of the developer who downloads the svn tree
and then has no internet connectivity.
Being able to use the svn trunk would also help developers developing modules.
This functionality could be post 2.0.0.

#4 - 06/12/2009 01:23 PM - David Welker

The module manager now works adequately for the purposes of the Sans Parks and 2.0 releases. I am closing this bug.

#5 - 03/27/2013 02:25 PM - Redmine Admin

Original Bugzilla ID was 3947