

## Kepler - Bug #5161

### can't always use Module Manager to change to an older patch of a suite

08/25/2010 04:27 PM - Derik Barseghian

<b>Status:</b>	Resolved	<b>Start date:</b>	08/25/2010
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	David Welker	<b>% Done:</b>	0%
<b>Category:</b>	interface	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.2.0	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	5161		
<b>Description</b>			
<p>If a module is published and a patched version of this module is published and the user has downloaded both and the user uses the MM to change to the original version of the module Kepler will restart and automatically use the newest patch of the module that's available on the local system, with no notice.</p> <p>I would expect a user would be able to intentionally change to the old patch, especially since it's visible in the MM's gui. Even if 'Automatically download new patches' is left on, they should get the prompt for an update on restart, I would think.</p> <p>Is this a known issue, compromise, or intentional choice the MM is currently making? It's not strictly necessary for 2.1, because the Import Module Dependencies feature does allow this (and that's when it's <b>really</b> necessary). Also since the IMD feature can prompt a user to do this (restart into an older patch), I think a user would think they could also use the MM to do it.</p>			

#### History

##### #1 - 09/07/2010 09:10 AM - David Welker

This is an interesting point.

The issue is this. If you a suite references ^ at the end, it means, use the latest available patch. Your question is, what if the user wants a different behavior AFTER they already have downloaded the newer patches. That is, maybe they find that the new patches don't work for them (despite the fact that patches should normally be nothing more ambitious than minor bug fixes).

The issue is definitional, actually. Module foo-2.1.^ means use the latest patch available on your system. Module foo-2.1.3 means use exactly foo-2.1.3, even if food-2.1.4 is available. If a precise patch was required, then the suite developer can and should specify a precise patch.

However, the situation we have here is where the user wants to roll back the application of the older patch. Perhaps things worked fine for them before the patch and the "bug fix" in question was actually harmful to their work. Clearly, a developer who specifies module foo-2.1.^ rather than a more precise version would not have been in a position to anticipate this.

It should be noted that this problem is solvable outside of the module manager. And that is probably how it should be solved. All the user who wants a precise version has to do is either move or delete the latest patches from <user.home>/KeplerData/kepler.modules/ or they can change modules.txt in <user.home>/KeplerData/kepler.modules/build-area.

The reason I am hesitant to try to solve this problem within the module manager is because:

(1) This should be a relatively rare problem, assuming that patching is used appropriately, users should not normally demand to roll back patches.

(2) We would actually have to change the definition of ^. Somehow, instead of referring to the latest patch available on the system, ^ now would refer sometimes to the latest patch available sometimes and actually an older patch at other times. I am not even sure how this could be made to work. Although I am probably clever enough to think of a way, it probably wouldn't be pretty. I think we would lose enough in clarity concerning the meaning of ^ that allowing the user to solve this problem through the module manager might be unwise.

(3) On the other hand, I could imagine implementing a "rollback" feature accessible through the module manager. It would be possible for the module manager to be aware of EXACTLY what state the module manager was in previously and "rollback" or "undo" back to that state. This would not involve changing the definition of ^, since what the module manager would remember is the exact list of modules involved and not the previous suite.

However, I am not sure that implementing a rollback feature for 2.2 is a priority. I am therefore tempted to close or postpone this bug, but will wait for other opinions before doing so.

##### #2 - 09/15/2010 11:57 PM - David Welker

I have actually decided to address this problem with a new feature. Now the user, in addition to selecting a particular major and minor version of a suite (the default) which will accept patches, can select a check box to select a precise suite which will not be subject to patching.

#3 - 03/27/2013 02:29 PM - Redmine Admin

Original Bugzilla ID was 5161