

Kepler - Bug #5396

rename ptolemy-8.1.0 module for the kepler 2.2 release

05/05/2011 04:44 PM - Derik Barseghian

Status:	Resolved	Start date:	05/05/2011
Priority:	Normal	Due date:	
Assignee:	David Welker	% Done:	0%
Category:	general	Estimated time:	0.00 hour
Target version:	2.2.0	Spent time:	0.00 hour
Bugzilla-Id:	5396		
Description			
Christopher mentioned that the next ptolemy release might be called 8.1, and so he'd prefer kepler-2.2 name the included ptolemy module ptolemy-8.0.2 (why 8.0.2 and not 8.0.1, Christopher?).			

History

#1 - 05/05/2011 05:35 PM - Derik Barseghian

Besides probably breaking our rules of what constitutes a patch-level change (perhaps we can make an exception for ptolemy since it's the bottom-most module in our stack and thus a bit unique?), another thing to keep in mind is that we won't be able to patch ptolemy-8.0.0 for kepler 2.0 or 2.1 if we release this as 8.0.1 (no numbers left between). I don't really expect anyone will want to do this, I just want to mention it.

#2 - 05/05/2011 05:41 PM - David Welker

This is not acceptable. Modules where the "patch" version number indicates a patch and will be treated as such by the PatchManager.

There really is no good reason to keep our ptolemy module versioned in a manner that is in sync with Ptolemy, at least absent some sort of agreement to modularize Ptolemy itself. The module ptolemy is not the same as a release of Ptolemy. If someone wants a release of Ptolemy, they shouldn't come to the Kepler project to get it, they should go to the Ptolemy Project at <http://ptolemy.eecs.berkeley.edu/> not to the Kepler Project at <http://www.kepler-project.org/>.

If we want to rename the ptolemy module in a manner to avoid it being confused with an actual release of Ptolemy, we should consider renaming that module to ptolemy-2.2. We could then include a prominent disclosure that the ptolemy module is NOT a release of Ptolemy itself.

In any case, renaming the ptolemy module (which again, is not the same thing as Ptolemy) to 8.0.2 or 8.0.1 is not technically feasible.

#3 - 05/06/2011 09:46 AM - Christopher Brooks

The pttl development tree has a version number of 8.1.devel, see `ptll/ptolemy/kernel/attributes/VersionAttribute.java`

Traditionally, after branching the pttl tree in preparation for a release, I bump the version number of the trunk to X.1.devel and the release branch is X.0.alpha.

The reasoning for the X.0.alpha -> X.0.beta -> X.0.1 sequence is lost in the sands of time. I remember I tried X.0.alpha -> X.0.beta -> X.0 and there were problems. I don't remember what the problems were. It will probably come to me if I try it, or it could have been an artifact of installer software that is no longer being used.

The reason to go with 8.0.2 instead of 8.0.1 is because 8.0.1 has already been released as the current Ptolemy II 8.0 release.

It is unfortunate that the module system has a technical limitation. `VersionAttribute.java` is based on the JNLP versioning system which allows for arbitrary lengths of version names.

Kepler-2.1 was presumably based on a version of the pttl tree that was not cleaned. Who knows what shipped? The files were not

formatted, no checks were made of the licensing or whether there were broken links.

The notion of using a different version number for the version of ptII that is shipped with Kepler will work fine until there are problems. Tcl/Tk had similar issues, where Tk had a lower version number than Tcl, but required Tcl to run. The confusion about the version numbers consumed quite a bit of email list discussion.

Another issue is that ptII has its version number encoded in the VersionAttribute. If you have a module named ptolemy-2.0.0, then the VersionAttribute cannot be set to 2.0, as Ptolemy II 2.0 was released in 2002 with VersionAttribute set to 2.0. A side issue is that the "ptolemy" module is misnamed. The name of the product is Ptolemy II. The svn module is ptII. The ptolemy cvs repository is Ptolemy Classic, which is a separate repository. By rights, the Kepler "ptolemy" module should be renamed to "ptII". However, this is a detail and probably not worth addressing.

If there are technical reasons that you can't ship a ptII version of 8.0.2, then I'd be happy to clean the tree and create a version 8.1.alpha branch from the ptII head. It would take a two weeks to get a real 8.1.0 tree out. I'm traveling next week. I could clean the tree today and get a 8.1.alpha branch out by the end of the weekend.

In summary, my primary concern is that Kepler not ship with a version number of Ptolemy II that is the same as a version of Ptolemy II that has already shipped. This means that the Ptolemy II version number should be greater than 8.0.1.

My secondary concern is that Kepler should not artificially bump the Ptolemy II version number up. We were thinking of shipping a Ptolemy II 8.1, so discussing this is good. Kepler **could** ship with a Ptolemy II 8.1 version number if I clean the tree and the VersionAttribute is set to 8.1.

My tertiary concern is that I would prefer that Kepler ship with cleaned Ptolemy code. This requires a little coordination and planning. I can usually clean a tree in a day. This concern is actually the one that really matters the most because an uncleaned Ptolemy II tree likely has broken links and could have licensing issues.

#4 - 05/06/2011 12:04 PM - Derik Barseghian

Just to note: Kepler 2.1 uses the same version of ptolemy that Kepler 2.0 did (and we didn't create an installer for 2.1, so the user didn't have to download this twice). Kepler refers to this as module "ptolemy-8.0.0" and it's the ptII/branches/rel-8-0-beta-2 branch at r58234.

#5 - 05/24/2011 05:25 PM - David Welker

For now, I am proceeding on my understanding that Kepler modules are independent entities and therefore we should not couple Kepler and Ptolemy release cycles. My view is that "ptolemy-8.1" module should be renamed to be "engine-2.2" since Ptolemy is the "engine" upon which Kepler currently runs. This would avoid any confusion about whether Kepler is a way one gets official releases of Ptolemy.

I think it would be a mistake to couple Ptolemy and Kepler release cycles, but ultimately this is a leadership team decision regarding the fundamental definition of what Kepler is and what its relationship to Ptolemy is. Is Kepler an independent project or is it a mere extension or subproject of Ptolemy? My understanding has always been that Kepler is an independent project, but some other people may have different views.

If we are an independent project, we should not couple Kepler and Ptolemy release cycles or management.

#6 - 05/25/2011 04:44 PM - David Welker

Per discussion with Bertram, renaming ptolemy-8.1 to ptolemy-kepler-2.2 which mean "the Ptolemy version used for Kepler 2.2."

Closing.

#7 - 03/27/2013 02:30 PM - Redmine Admin

Original Bugzilla ID was 5396