Kepler - Bug #4091

Make it so the build system can use a "blessed" version of Ptolemy.

05/20/2009 02:44 PM - David Welker

Status:	Resolved	**Start date:**	05/20/2009
Priority:	Normal	**Due date:**	
Assignee:	David Welker	**% Done:**	0%
Category:	build system	**Estimated time:**	0.00 hour
Target version:	2.0.0	**Spent time:**	0.00 hour
Bugzilla-Id:	4091		

Description

gatekeeping mechanism where by we use a "blessed" version of Ptolemy that is very close to the trunk, but not necessarily on the trunk. This will shield us from small compilation errors in Ptolemy. Every night, a process will run that will update the "blessed" version of Ptolemy to the trunk if compilation occurs correctly and certain tests are met.

History

#1 - 05/26/2009 07:40 AM - Christopher Brooks

We should use the term "stable" to refer to revisions of Ptolemy and Kepler
that are likely to build for users. A true stable version usually requires
developer time to validate the revisions necessary. If developer time is
available, in the perfect world, there would be an acceptance test such as:
- Build Kepler and Ptolemy from a clean checkout
- Start Kepler
- Run a particular model
Other things to try:
- Build a simple model

The minimum set of acceptance tests for a stable version should be clearly
documented so that users of stable versions know what they are getting.

If developer time is not available, then we could try automatically
designating particular versions as stable, but certain types of failures
will slip through.
There are several different levels of stability of a particular Ptolemy build:
1) Does Ptolemy and Kepler build properly? A continuous build system
would help detect problems here.
Ptolemy does not have a continuous build system, Kepler's continuous build
system does not send useful email.
One idea would be to have a mailing list that included people who made
changes within the last N days and mail them the continuous build info.

If the build fails, then obviously something is wrong and the revisions
of both Kepler and Ptolemy are not stable.

2) Tests should be run and the results evaluated. Unfortunately, the tricky
part is that Kepler will work fine if some tests fail. Probably the thing
to do is to develop a few simple system tests that must pass if a particular
revision is to be considered stable. The system tests should be both
non-graphical and graphical. These tests could invoke models and test
the known good results. We need better info about running the Kepler tests.

3) UI tests would help indicate the overall stability. Common problems include
missing actors and problems building under Eclipse. Testing for these
automatically would be difficult, but are worth a shot.

#2 - 05/26/2009 10:46 AM - David Welker

I agree with you that the word "stable" is better than "blessed" and that is the the word used in one of several new build system commands. The basic infrastructure for this task is not complete. However, right now, the only testing that is done is to ensure that Kepler still compiles against the head of Ptolemy (which also must compile). If that happens, then the revision associated with the head of Ptolemy is considered to be "stable" and is designated as such so that any command that invokes "ant update" "ant get" or "ant change-to" directly or indirectly will now retrieve the new stable version of Ptolemy.

Clearly, some errors will still get through, so testing that goes beyond compilation would be desirable. For example, if configuration files in Ptolemy were to be moved and the references to those files in Kepler not updated, this would cause run time incompatibilities that would not be caught under the current approach. However, the point of this exercise is not to give Kepler trunk developers 100% stability (if developers want such stability, the should work off a published version of Kepler) but instead to greatly increase stability. Furthermore, since the "stable" revision is stored in PTII, any update to an older version of Kepler will get the corresponding version of Ptolemy. So, it is very easy to step back to working with a slightly older revision of Kepler (and Ptolemy) with a single command of the form: "ant update -Drev={$revision}."

Nonetheless, more testing would be desirable. And we now have the infrastructure that makes it very easy to add such tests when we get around to deciding exactly the nature of whatever additional tests we want. However, for now, I am going to move on to work on other tasks. This bug is therefore closed. However, I am opening a new bug that references the desirability of doing more extensive testing of Ptolemy.

#3 - 05/26/2009 06:42 PM - Christopher Brooks

I'm confused.
How do I build from the Ptolemy head?

It seem like "ant update" should actually update the tree to the head of
each svn repository, much like "svn update" would.

Maybe "ant stable" should update to the what ever version is marked
as stable?

#4 - 05/26/2009 11:10 PM - David Welker

Hi Christopher,

That was my bad. I forgot to include the names of the new commands in my comment closing the issue.

To work from ptolemy head, you type in:

ant use-ptolemy-head

To switch back to using the stable version of ptolemy, you type in:

ant use-stable-ptolemy

That is it.

As far as what is default, I tend to think that ant update should probably use the stable version of Ptolemy. Whatever is default will be what most developers actually do in practice.

Keep in mind that the difference between the head and the "stable" version of Ptolemy should only be a few revisions in most cases. Kepler tries to update the stable version to be the head every single night. Basically, Kepler is committed to working at the head of Ptolemy, but with a tiny little firewall.

Of course, it is possible to modify Ptolemy itself from Kepler. But, this is not the default case. I do not think that asking Ptolemy developers using the Kepler build (which would include Kepler developers who want to modify Ptolemy) to type in the extra command: 'ant use-ptolemy-head' is much to ask. What do you think?

#5 - 05/27/2009 11:11 AM - Christopher Brooks

Ok, good news that I can build against the ptII head.

The word "update" in "ant update" means something specific, which is that
"svn update" is being run. I think that by changing the meaning of update,
which means that "ant update" is not building against the head is a misuse
of the word "update", but it is not that serious. I think "ant update-stable"
should update against stable versions of all the repositories, including
kepler. "ant update" should update against the head of all the trees.

That is just my $0.02, do as you wish here.

Happily, when ant update is run, the message shows that the head of Ptolemy is
not being retrieved:

```
[update-modules] Updating ptolemy...
[update-modules] svn -r 53941 update /Users/cxh/src/kepler/ptolemy
[update-modules] At revision 53941.
```

The above message should be modified to be very clear when I'm running against
a version of Ptolemy that is not the head and how to build against the head by running the ant commands you describe.

Where is the documentation of the procedure for how the stable of version number of Ptolemy is determined? For example, how often is the version of Ptolemy checked? How is it updated?

The next step is to definitely include a smoke test that starts up Kepler to validate the configuration. This will catch many errors. Ptolemy has
this capability available in the about:copyright facility. Basically, if
you go the the "information about Ptolemy" link and follow the copyright page

eventually there is a list of configurations and tests.

Kepler could run the same tests by getting the configuration and calling
ptolemy.actor.gui.Configuration.check().

#6 - 05/28/2009 05:04 PM - Christopher Brooks

So every time I run "ant update", I need to also run "ant use-ptolemy-head".
The README.txt and web pages should be updated to reflect this change, I'm sure
that Ptolemy developers will stumble over it. Also, when "ant update" is run, the
message should say something about possibly needing to run
"ant use-ptolemy-head".

Nonetheless, this is pretty cool that Kepler is somewhat insulated from
the daily chaos of the Ptolemy project.

#7 - 05/28/2009 05:24 PM - Christopher Brooks

Actually, maybe "ant update-head" could replace "ant use-ptolemy-head"?
I'll never remember "ant use-ptolemy-head". This also means that
"ant update" gets the non-head version of Ptolemy and "ant use-ptolemy-head"
then gets the head. This will be a small problem for me when I'm
connected via modem later this summer.

How are "ant use-stable-ptolemy" and "ant update" different?

Below is the output of ant:

```
[exec]  use-ptolemy-head       Update Ptolemy so that the head of ptolemy is used.
     [exec]  use-stable-ptolemy     Update Ptolemy so that stablalized ptolemy revision is used.
     [exec]  validate-ptolemy       Tests the current version of Kepler against the current version of Ptolemy
 and changes the validated ptolemy revision if appropriate.
```

I see we now have "ant validate-ptolemy", which looks good, I just tried it.

I think this can be closed if:
1) ant use-stable-ptolemy is changed to ant update-head
2) ant update prints messages about using ant update-head for the
modules that are not at the head

Optionally, it would be nice if there was a way if "ant update-head"
would update from the head from all the modules, instead of how
"ant use-stable-ptolemy" just updates ptolemy.

#8 - 05/28/2009 05:44 PM - David Welker

Hi Christopher,

Good comments.

Actually, the command ant use-ptolemy-head and ant use-stable-ptolemy have been both removed. There is a new command in their place, called ant update-ptolemy. This command can be used in one of two ways:

ant update-ptolemy -Drev=head
ant update-ptolemy -Drev=stable

If one of these isn't specified, the command fails.

You have suggested using the update command and having a parameter. Adding that as an option would probably be a good idea. However, I want a separate command just to update-ptolemy, because just because you want to use the head of ptolemy that doesn't mean you want to use the head of everything else.

Of course, there is actually another way to specify that you only want to update ptolemy, and that is by using the -Dinclude option in an update. So, for example, to update just ptolemy, you could type:

ant update -Dinclude=ptolemy

If we combined this with your idea of specifying the "head" as a revision, then we could just type

ant update -Dinclude=ptolemy -Drev=head

Alternatively, you could switch back to using a stable version of ptolemy with:

ant update -Dinclude=ptolemy -Drev=stable

This is generally applicable functionality. So, for example, if you just wanted to update the core module, you could type:

```
ant update -Dinclude=ptolemy -Drev=head
```

However, I think switching between using the head of ptolemy and the stable version is common enough that it deserves its own separate, slighly easier to type command in addition to the ability to work with it in this more generic way. So, I think I want to keep the "ant update-ptolemy" command. But, if you prefer, I could remove this command and we could use "ant update" for everything.

#9 - 05/28/2009 05:47 PM - David Welker

Oh, I should add that per your suggestion, I have added an informative message when you are updating ptolemy:

```
Updating ptolemy...
[update-modules] Using the stable version of ptolemy. Use 'ant update-ptolemy -Drev=head' to work from the head.
[update-modules] svn -r 54012 update /Users/davidwelker/kepler.modules/ptolemy
[update-modules] U    /Users/davidwelker/kepler.modules/ptolemy/doc/makefile
[update-modules] U    /Users/davidwelker/kepler.modules/ptolemy/ptolemy/configs/test/allConfigs.tcl
....
```

It looks better outside of Bugzilla, which has these very nasty limits on character length in terms of how long a line can be. (But, don't even get me started on ranting about Bugzilla! =))

#10 - 05/28/2009 05:55 PM - Christopher Brooks

Thanks for adding the message, that will help.

I think if we had "ant update-head", then I'd be happy. The other commands
are too long, I won't remember them. If "ant update-ptolemy" also updated
the other modules to their heads, then that would be ok, as it would be
the same as "ant update-head".

You wrote:

> ant update-ptolemy -Drev=head
> ant update-ptolemy -Drev=stable
>
> If one of these isn't specified, the command fails.

To many options in the above, how about:

```
ant update       - updates everything to stable, if present, otherwise to head.
ant update-head  - updates everything to head
```

#11 - 05/28/2009 06:10 PM - David Welker

How about:

```
ant update -Drev=head
```

Updates everything, including ptolemy, to the head?

#12 - 06/12/2009 01:30 PM - David Welker

This functionality is now fully implemented.

#13 - 06/12/2009 01:30 PM - David Welker

Forgot to close the bug with the last comment.

#14 - 03/27/2013 02:25 PM - Redmine Admin

Original Bugzilla ID was 4091