Kepler - Bug #3898

update .kepler instead of removing it between versions

03/17/2009 09:59 AM - Christopher Brooks

Status:	Resolved		**Start date:**	03/17/2009
Priority:	Immediate		**Due date:**	
Assignee:	Aaron Aaron		**% Done:**	0%
Category:	actors		**Estimated time:**	0.00 hour
Target version:	2.0.0		**Spent time:**	0.00 hour
Bugzilla-Id:	3898			

Description

Thanks to Qiao Huijie for pointing this out.
When switching between Kepler-1.0.0 and the Subversion development tree,
the ~/.kepler directory must be removed.
If it is not removed, then no actors are present in the actor browser pane.

I'm not sure why there is an incompatibility, but the code should detect
this incompatibility and prompt the user.

In addition, upon startup, the code should check to see if the actor browser
pane is empty and, if it is, prompt the user to delete their ~/.kepler
directory

Also, the developer instructions should be updated to mention removing
the ~/.kepler directory. I'll update the instructions.

Related issues:

Blocked by Kepler - Bug #4337: 1.0 -> 2.0 upgrade issues tracking bug	**Resolved**	**08/26/2009**
Blocks Kepler - Bug #4702: KAR to module conversion utility	**Resolved**	**01/27/2010**

History

#1 - 03/18/2009 12:02 PM - Christopher Brooks

I updated the Build System instructions and the Eclipse instructions
to include running "ant clean-cache". This seems much better than
trying to describe how to remove ~/.kepler.

Also, at one point I did see a message on stdout about removing ~/.kepler,
so some error checking is happening. To close this bug, the follow
acceptance criteria should be met:
1) Run Kepler 1.0, verify that the actors are present.
2) Install Kepler from the developement tree, verify that the actors are present.
3) Do step 1 again.
If there is an error with step 2, then a dialog should be displayed that
suggests running "ant clean-cache", or, ideally, clicking on a button that
would clean the cache and restart Kepler.

David Welker wrote:

> Just to chime in. I am not sure if deleting your .kepler directory is
> the solution to your problem or not. But, if it is, the best way to do
> it is by typing:
>
> ant clean-cache
>
> That command works across platforms and will delete your .kepler directory.
>
> And no, deleting the .kepler directory will not cause Kepler to
> "misbehave." (Although, sometimes the .kepler directory does mask some
> problems that exist, and it may sometimes that deleting .kepler caused a
> problem. But deleting .kepler doesn't cause such problems, it merely
> unmasks them.)

[Qiao Hui-jie] wrote:

Thanks for that.

Are you referring to the file present not in the Kepler folder, but
separately under Users?
I can see it present, but I have Kepler 1.0.0 installed. Will deletion
of that affect the working of Kepler 1.0.0 in any way?

I am asking since all previous workflows have been done in Kepler
1.0.0 and I don't want Kepler 1.0.0 to begin misbehaving in any way.

Thanks!

#2 - 03/18/2009 12:21 PM - Matt Jones

Although in development mode it is fine to remove .kepler either manually or with clean-cache, there would be problems for users to regularly remove
this. When a user selects 'Save to Library...' for an actor from the canvas, Kepler creates a new KAR file and puts it into the cache which is then
loaded into the Library pane. This allows users to create new actors from composites or specialized atomics very easily. If the .kepler directory is
removed, these would be lost, and if the cache contains the only copy of the kar (ie., if they didn't 'Upload to repository' as well), then their actor is
lost. So, in general, we should be creating a system in which new versions of kepler have no need to delete the .kepler directory when upgrading.
Using the development tree should be handled similarly. The hard part, of course, is going backwards (e.g., from trunk to 1.0.0), in that 1.0.0 may not
be able to execute new actors from a newer version. It would be nice if the next version of kepler could recognize based on actor metadata which
actors it can handle, and ignore those that it can't. This would allow switching among different versions of Kepler, and older versions would just
ignore actors in the cache that they can't handle. Overall, we need a better strategy for handling .kepler.

#3 - 04/01/2009 11:33 AM - Derik Barseghian

A similar problem occurs when switching between suites.
If in suite A you've developed an actor, and you instantiate and then Export it to kar including the class, and then Import Archive, it will show up in the
library. If you then change-to suite B and launch kepler, the actor will show up in the library but will give an error if you try to use it. Also if you try to
import it, you'll get an error (about being unable to clone). If you run ant clean-cache before launching suite B, you can import the actor.

#4 - 04/02/2009 01:28 PM - David Welker

I have a proposed solution to this problem. I think that the cache should be module aware. If you are running kepler-trunk, a different cache (or part of
the cache) should be used than if you are running kepler-1.0 or daks-base.

When you do a change-to, to kepler-1.0 from kepler-trunk for example, there would be no memory of what was in the cache for kepler-trunk, but then
you would not have any strange incompatibilities. When you changed back to kepler-trunk, your data previously stored in kepler-trunk would be
available again.

Alternatively, I suppose we could develop a cache where there are some basic items that would appear regardless of change-to, but then perhaps
there are specific items that are known to cause problems which behave in the manner described above (i.e. if you do a change-to kepler-1.0, that
cache item is not available until you change-to back to kepler-trunk).

Any thoughts?

#5 - 04/22/2009 03:26 PM - Chad Berkley

Aaron agreed to take this on with help from david and chad. We need to be able to upgrade the .kepler directory between versions and keep version
specific schema, databases, etc in their own folders for backwards compatability.

#6 - 06/01/2009 07:11 PM - Aaron Aaron

Chad, David, and I had a discussion today about how to handle the .kepler directory for the 2.0 release. The conclusion was the following:

- use the authority and namespace to generate a subdirectory in the .kepler directory for each instance of Kepler (refer to as .kepler/instance
directory)
- cache directory, actorLibrary, etc. will go into the .kepler/instance directory
- ModulesTree class will be used to create one directory for each module in the .kepler/instance/modules/ directory (refer to as a module's .kepler
directory)
- A module can then use it's .kepler directory for any disk activity, Also resources from a module can be copied to it's .kepler directory for read/write
capability
- the Kepler cache should only be a mirror of KAR files and resources on disk to prevent any future issues with version management
- Kepler 1.0 caches will be converted to the new system by building KAR files from the old cache and allowing the new cache to be built from those
KAR files on disk
- The current InstanceAuthNamespace will be kept in the .kepler directory
- build function for cleaning cache should be updated to clean only the current .kepler/instance directory

#7 - 06/17/2009 07:10 PM - Aaron Aaron

Progress on this bug. The InstanceAuthNamespace is kept at the project root. A configuration directory is now created in the project root that
contains a directory named according to the authority and namespace of the running kepler. Then there is a modules directory that contains a folder

for each module. This effectively defines an area on disk where a particular installed module can write too without worrying that another installation is also using that space. The .kepler directory is now defined as a place on disk where modules can save and read runtime information that is meant to be shared between different installations of Kepler. Left to do is to detect and export an old cache to KAR files so they can be loaded into the 2.0 installation.

#8 - 08/25/2009 11:27 AM - Christopher Brooks

This bug was discussed in kepler-users:

Ben Leinfelder writes:

> Rick,
> The path is specified in DBConnectionFactory but should probably be moved into the config.xml file so that it is more readily changed.
> I'm fearful that simply changing the HSQL database path will not be a viable option as there are many other cache-specific files in the ~/.kepler directory that may interfere with this mix of old and new databases/cache files.
> Certainly give it a try (strongly recommend backing up your existing .kepler beforehand) to see if you have satisfactory results.
> You might be happier running the different Kepler versions as a different user on the same computer - that will allow you to keep the ~/.kepler directories distinct since they will be under different user directories.
> -ben
>
> On Aug 25, 2009, at 10:33 AM, Rick Moore wrote:
>
> > Thanks for the prompt response Ben.
> >
> > This seems to indicate that the path to the HSQL database is hardcoded somewhere in the Kepler src base. If that is the case, would someone please help me find where that is ? I haven't had any luck with cygwin/grep or Windows file search.
> >
> > I need to be able to run both the prod and dev instances on a daily basis. The prod version to run/maintain existing workflows and the dev version to build/debug new actors. We will eventually go to the newest source base for both, but that may take a couple of months.
> >
> > Rick
> >
> > ben leinfelder wrote:
> >
> > > Rick,
> > > Unfortunately we do not currently have an "upgrade" approach for going from Kepler 1.0 to the development trunk (though it is something we are attempting to support in the future).
> > > To preserve your existing HSQL database (say, if you plan on using Kepler 1.0 again after you successfully run Kepler from the trunk) I'd recommend copying your ~/.kepler directory to a safe location and then re-running Kepler from the trunk. There have been substantial changes to the cache database (the HSQL version was upgraded and various tables have had their schema altered). Your new ~/.kepler (automatically created when you launch Kepler from the trunk) will allow you to use the trunk version. If/when you want to go back to Kepler 1.0, you'll need to swap in your old ~/.kepler and launch 1.0.
> > > Thanks,
> > > -ben
> > >
> > > On Aug 25, 2009, at 9:38 AM, Rick Moore wrote:
> > >
> > > > I just tried to run Kepler 1.0.0 and it failed. This extract from the stack trace suggests that the problem is a corrupt HSQL database.
> > > >
> > > > java.sql.SQLException: Wrong database file version
> > > > at org.hsqldb.jdbc.jdbcUtil.sqlException(Unknown Source)
> > > > at org.hsqldb.jdbc.jdbcConnection.<init>(Unknown Source)
> > > > at org.hsqldb.jdbcDriver.getConnection(Unknown Source)
> > > > at org.hsqldb.jdbcDriver.connect(Unknown Source)
> > > >
> > > > I think this happened because I recently downloaded and built Kepler from the dev tree and it stomped on my production database.
> > > >
> > > > Is there any way to tell one of these instances to look for the HSQL database in a different location than my home directory ?
> > > >
> > > > Thanks,
> > > >
> > > > Rick Moore
> > > > Content Management Specialist
> > > > Information Science
> > > > Cornell University
> > > > email: rem63 at cornell dot edu

#9 - 08/25/2009 11:50 AM - Christopher Brooks

Rick Moore writes:

Turns out that DBConnectionFactory gets it's path from DotKeplerManager. Here are the files/lines I had to change to make this work:

- build-area/src/org/kepler/build/CleanCache.java: 39> File
  keplerCache = new File(userDir, ".kepler");
- core/src/org/ecoinformatics/util/Config.java: 76> private
  static final String settingsdir = ".kepler";
- core/src/org/kepler/util/DotKeplerManager.java   26> +
  File.separator + ".kepler" + File.separator;
- core/src/org/kepler/util/DotKeplerManager.java: 122> + ".kepler"
  + File.separator + "cache"

As it now stands, my production version uses ${user.home}\.kepler and my dev version uses ${user.home}\.kepler-dev

Rick

#10 - 08/26/2009 01:18 PM - Matt Jones

Progress has been made, but there's still work to be done. There are two main directories in which things can be stored in Kepler -- the product 'install' directory (usually in an OS system folder) and the .kepler directory (for user-specific artifacts and files). To close this bug, the following should all be true:

1. The 'install' directory can be removed and replaced by the same version of kepler, or by a new version, and everything still works fine
-- this implies that any files that should not be lost across upgrades must NOT be in the install directory (e.g., InstanceAuthNamespace)

2. User-created KAR files and other artifacts in the .kepler directory should migrate across version upgrades without duplication -- there is no need to have multiple copies of the same KAR file

3. There should never be a need to 'clean' any files from the .kepler directory -- neither during the development cycle nor for a binary product install

#11 - 09/18/2009 11:08 AM - Christopher Brooks

One possible solution would be to have the current devel version of kepler
use .kepler/2.0/ instead of .kepler/

I do like the idea of being able to pass in the .kepler directory as a command
line argument. Java can take properties as a command line argument, so if
the .kepler directory was a property such as kepler.keplerDirectory, then
we could invoke java with something like
-Dkepler.keplerDirectory=/users/me/.kepler-myversion

Chad writes:

> Hi Michal,
>
> This is something we are currently working on for the 2.0 release. There is a bug here:
> http://bugzilla.ecoinformatics.org/show_bug.cgi?id=3898
>
> Unfortunately, the 1.0 directory will not have this functionality in the future.
>
> Aaron Schultz (aschultz@nceas.ucsb.edu) is working on this and I think he could answer any questions you have better than I could.
>
> Thanks,
> chad
>
> Michal Owsiak wrote:
>
>> Hello all,
>>
>> Is it possible to pass home directory as a parameter to Kepler?
>>
>> We are developing application for both: Kepler 1.0 and Kepler 1.x.
>>
>> At the moment we are forced to either remove .kepler home each time we
>> switch distribution or we have to create symbolic links. Not to mention
>> the possibility of running Kepler 1.0 and 1.x simultaneously.
>>
>> If there is some sort of argument/switch/parameter that allows to set
>> different .kepler location it would be great.
>>
>> Thanks in advance for the info
>>
>> Cheers

Michal

#12 - 09/18/2009 11:29 AM - Michal Owsiak

(In reply to comment #11)

This idea (of passing parameter through the Java properties) is exactly what I'd like to have in Kepler.

This way, I would be able to start separate instances of Kepler at the same time. Additionally, regular user (non developer) would not notice a difference if default value of -Dkepler.keplerDirectory would be set to "$HOME/.kepler"

> One possible solution would be to have the current devel version of kepler
> use .kepler/2.0/ instead of .kepler/
>
> I do like the idea of being able to pass in the .kepler directory as a command
> line argument. Java can take properties as a command line argument, so if
> the .kepler directory was a property such as kepler.keplerDirectory, then
> we could invoke java with something like
> -Dkepler.keplerDirectory=/users/me/.kepler-myversion
>
> Chad writes:
>
>> Hi Michal,
>>
>> This is something we are currently working on for the 2.0 release. There is a bug here:
>> http://bugzilla.ecoinformatics.org/show_bug.cgi?id=3898
>>
>> Unfortunately, the 1.0 directory will not have this functionality in the future.
>>
>> Aaron Schultz (aschultz@nceas.ucsb.edu) is working on this and I think he could answer any questions you have better than I could.
>>
>> Thanks,
>> chad
>>
>> Michal Owsiak wrote:
>>
>>> Hello all,
>>>
>>> Is it possible to pass home directory as a parameter to Kepler?
>>>
>>> We are developing application for both: Kepler 1.0 and Kepler 1.x.
>>>
>>> At the moment we are forced to either remove .kepler home each time we
>>> switch distribution or we have to create symbolic links. Not to mention
>>> the possibility of running Kepler 1.0 and 1.x simultaneously.
>>>
>>> If there is some sort of argument/switch/parameter that allows to set
>>> different .kepler location it would be great.
>>>
>>> Thanks in advance for the info
>>>
>>> Cheers
>>>
>>> Michal

#13 - 02/04/2010 12:59 PM - Chad Berkley

Matt identified these three milestones to be able to close this bug. I believe they are all now met. See my comments below each.

> 1. The 'install' directory can be removed and replaced by the same version of
> kepler, or by a new version, and everything still works fine
> -- this implies that any files that should not be lost across upgrades must
> NOT be in the install directory (e.g., InstanceAuthNamespace)

This is now true. All changeable files are now in the ~/KeplerData or the ~/.kepler directories

> 2. User-created KAR files and other artifacts in the .kepler directory should

migrate across version upgrades without duplication -- there is no need to have
multiple copies of the same KAR file

The kar files themselves are actually not stored in the .kepler directory for either 1.0 or 2.0. The cached, serialized version are. I see no way to migrate the 1.0 serialized forms without a major headache. I have changed the DotKeplerManager to create a new cache directory for 2.0 if it detects that 1.0 is already installed. I believe this to be the best workaround for this problem. it allows 2.0 to operate correctly as well as for 1.0 to continue to operate, albeit independently of one another.

   3. There should never be a need to 'clean' any files from the .kepler directory
   -- neither during the development cycle nor for a binary product install

I think we have finally gotten around this with some minor exceptions. You can now switch between suites without cleaning and the 2.0 installer will not require a clean .kepler directory.

I'm closing this bug. If there is a reason for it to still be open, please reopen with comments.

#14 - 03/27/2013 02:24 PM - Redmine Admin

Original Bugzilla ID was 3898