

## Kepler - Bug #2266

### SVG - Assigning Icons

11/11/2005 11:49 AM - Matthew Brooke

<b>Status:</b>	Resolved	<b>Start date:</b>	11/11/2005
<b>Priority:</b>	Immediate	<b>Due date:</b>	
<b>Assignee:</b>	Matthew Brooke	<b>% Done:</b>	0%
<b>Category:</b>	interface	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.0.0alpha8	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	2266		
<b>Description</b>			
<p>Icons can be assigned in actor moml, or can be assigned in the actor's java code. Assigning in moml means icons will show up for newly-created workflows, but <b>not</b> for old workflows. [Note that some actors can't be set via moml (eg EML200 datasource), so would need to add some to java code anyway]</p> <p>Adding icon paths directly to the actors' java code (note that existing svg icons are currently defined in java code) means icons show up for all actors, in new <b>and</b> old workflows. Could move actual actor-&gt;svg icon mappings into a separate settings file? Or would this cause too much indirection and add complication?</p> <hr/> <p>conf call of 11/10/05 - decided to put assignments in actor code, with actual mappings in a separate config file to allow for easy changes</p>			
<b>Related issues:</b>			
Blocked by Kepler - Bug #2245: TRACKING: Batik SVG Rendering - remaining tasks		<b>In Progress</b>	<b>11/04/2005</b>

### History

#### #1 - 11/22/2005 04:12 PM - Matthew Brooke

added code to:  
kepler/src/exp/ptolemy/actor/Director.java and  
kepler/src/exp/ptolemy/kernel/ComponentEntity.java  
to look in resource bundle:  
kepler/configs/ptolemy/configs/kepler/ResourceBundle\_svgIconMappings.properties  
for a key matching the the current actor or director classname. Resource bundle  
file contains key/value pairs of the form:  
"fully.qualified.class.Name=path/to/svg/icon/from/classpath"  
If resource bundle not found, or if it doesn't contain a value matching the key,  
the batik svg icon is not added, so rendering will default back to the original  
svg icon definition (in moml or java code, if defined).

STILL TO DO:

-----

A) The resource bundle file -

kepler/configs/ptolemy/configs/kepler/ResourceBundle\_svgIconMappings.properties

contains only a few mappings, since I'm really not sure which actors are  
supposed to use which icons, apart from the really obvious ones I've already  
done. Maybe Laura can help with this? Or maybe someone else who is familiar with  
most of the actors? Maybe this could be the same person who addresses bug [#2286](#)  
at the same time?

#### #2 - 11/23/2005 06:30 AM - Matthew Brooke

per my IRC conversation with matt, mappings should use LSIDs as keys instead of  
(or perhaps in addition to?) classnames. Explanation follows:

One executable class very well might support multiple actors. For example, the R  
actor can be used to implement a linear regression, and it should therefore be  
iconized as a Statistics actor. However, R can also be used to implement, say, a

population dynamic simulation - in which case it might need a modeling icon. The actor classname would be the same in both cases (RExpression), and currently they would have the same icon (a big R). The icon currently represents more how the actor is implemented than what it does, whereas the user cares about what it does more.

The LSID can be obtained from the actor code, since it is a property (so can use a `getAttribute()` call):

There is an LSID for the actor itself, which is the one we should use:

```
<property name="entityId" class="org.kepler.moml.NamedObjId"
value="urn:lsid:localhost:actor:7:1">
```

(There is also one for the class implementing the actor, which is **not** the one to use, since this has a 1:1 mapping to the class name: `<property name="classId" class="org.kepler.moml.NamedObjId" value="urn:lsid:localhost:class:883:1">`)

We might want to consider assigning "default" icons based on classname, and then let the lsid-based icon assignments override these. if they exist.

Note that the previous comments still apply (I will need help with actual mappings)

### #3 - 12/08/2005 12:19 PM - Matthew Brooke

implemented assignment by LSID. if none found, assigns by classname. Finally, if none found there, uses default blank teal icon. Issues that still need to be resolved:

1) The new icons don't show up for any actors that already have an icon defined in the MoML using this old-style type of assignment:

```
<property name="SequenceToArrayIcon"
class="ptolemy.vergil.icon.EditorIcon">
...etc
</property>
```

example - search for "sequence" in actor lib, and see "Sequence To Array" and "Array To Sequence" actors

2) Any actors that are Attributes instead of AtomicActors (eg all the actors that appear in the tree under Components -> Variables -> Parameters) now default to the Director icon for some reason

3) Composite actors (eg search for "Sinewave" or "RBoxPlot") get a plain teal icon, but with a blue-violet border (probably drawn elsewhere by layering some other icon on there). Composites can't have icons assigned by classname, since they aren't java classes. Assigning by LSID works for the icon in the tree (thumbnail raster icon), but not when it is dragged to canvas.

### #4 - 12/15/2005 09:51 PM - Matthew Brooke

With reference to item #s in previous comment:

1) recent changes to object manager/cache have circumvented this behavior - now works as expected

2) Components -> Variables -> Parameters are showing up as Directors because the ActorMetadata class is incorrectly creating them as Directors; this is therefore the correct behavior as far as the svg icon code is concerned. When ActorMetadata fixed, these actors should pick up the correct icons accordingly

3) Composite actors: blue-violet border fixed by chad's changes. Assigning by LSID not working when dragged to canvas: fixed by changes to XMLIcon.java.

Marking as fixed

### #5 - 03/27/2013 02:19 PM - Redmine Admin

Original Bugzilla ID was 2266