# Kepler - Bug #4478

## Configuring ports

10/16/2009 03:36 PM - shirley shirley

| | | | | |
|---|---|---|---|---|
| **Status:** | New | | **Start date:** | 10/16/2009 |
| **Priority:** | Normal | | **Due date:** | |
| **Assignee:** | Chad Berkley | | **% Done:** | 0% |
| **Category:** | actors | | **Estimated time:** | 0.00 hour |
| **Target version:** | 2.X.Y | | **Spent time:** | 0.00 hour |
| **Bugzilla-Id:** | 4478 | | | |

### Description

When the Kepler dev has been running for a long time (a few hours), there appears to be a problem with the configuring ports on all of the actors. When you add a port and hit commit, it doesn't do anything and the window w/ of the configuring ports just stays open. when you exit out if you decide to save ports, nothing happens and the window remains open. If you decide to not save port modifications, then the window closes but nothing's saved. This has happened a few times on mine but is solved when you close and restart Kepler.

### History

**#1 - 10/16/2009 03:53 PM - Christopher Brooks**

You are probably out of memory, next time use the OS to check how much
memory is being used.  Also, look under Tools -> Check System Settings
for how much memory is being used.

**#2 - 10/20/2009 10:32 AM - Oliver Soong**

I think I'm able to replicate this.  FYI, I still have 12506K memory free.  I'm not sure it's a memory issue, and I'm not sure it's related to running Kepler for a long time.  I think it has to do with misconfiguring ports.  I suspect some error handling is causing some sort of inconsistent state.  I'll try and find out a few more specifics on this, as it's going to bite me a bunch of times in the near future.  I get this in the console:

Exception in thread "AWT-EventQueue-0" ptolemy.kernel.util.InternalErrorException: Port stored in _ports not found in actual target.
at ptolemy.actor.gui.PortConfigurerDialog._populateActualPorts(PortConfigurerDialog.java:1955)
at ptolemy.actor.gui.PortConfigurerDialog._apply(PortConfigurerDialog.java:729)
at ptolemy.actor.gui.PortConfigurerDialog._processButtonPress(PortConfigurerDialog.java:783)
at ptolemy.actor.gui.PtolemyDialog.actionPerformed(PtolemyDialog.java:145)
at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:1995)
at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2318)
at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:387)
at javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:242)
at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:236)
at java.awt.Component.processMouseEvent(Component.java:6263)
at javax.swing.JComponent.processMouseEvent(JComponent.java:3267)
at java.awt.Component.processEvent(Component.java:6028)
at java.awt.Container.processEvent(Container.java:2041)
at java.awt.Component.dispatchEventImpl(Component.java:4630)
at java.awt.Container.dispatchEventImpl(Container.java:2099)
at java.awt.Component.dispatchEvent(Component.java:4460)
at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4574)
at java.awt.LightweightDispatcher.processMouseEvent(Container.java:4238)
at java.awt.LightweightDispatcher.dispatchEvent(Container.java:4168)
at java.awt.Container.dispatchEventImpl(Container.java:2085)
at java.awt.Window.dispatchEventImpl(Window.java:2475)
at java.awt.Component.dispatchEvent(Component.java:4460)
at java.awt.EventQueue.dispatchEvent(EventQueue.java:599)
at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:269)
at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:184)
at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:174)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:169)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:161)
at java.awt.EventDispatchThread.run(EventDispatchThread.java:122)

**#3 - 10/20/2009 05:38 PM - Christopher Brooks**

Yep, it looks like a bug.  Let me know if you can replicate it.  I just tried it in Ptolemy and could not get a stack trace.

**#4 - 10/21/2009 09:41 AM - Oliver Soong**

I can't seem to figure out exactly what replicates it. The closest I can come is getting ports to swap relations.

I don't know if they're actually related. Take two ports (e.g., input named in and output named out). Hook them up (so the pre-actor outputs to "in" and the post-actor receives from "out". Open the ports window and swap the names, then click apply. You get an error. You can't apply again, but you can commit. As soon as you do that, the relations swap sides.

**#5 - 10/21/2009 10:27 AM - shirley shirley**

I can't seem to go through the actions that replicate it but it happened again yesterday for me and I wasn't doing anything different.

**#6 - 10/21/2009 11:54 AM - Christopher Brooks**

I was able to reproduce this once, but now I can't.

The problem here is that swapping the name of two ports (or any two objects that
extend NamedObj) require renaming one to a temporary name first.
So, to swap A and B.
1. Rename A to C
2. Click Apply
3. Rename B to A
4. Click Apply
5. Rename C to B
6. Click Commit.

When I use the port configuration dialog to just swap two names:
1. Rename A to B
2. Rename B to A
3. Click Apply

I get a name duplication exception, which is as expected. What happens
here is that the table is processed by row and any change requests are
added to StringBuffer of change requests. When all the rows are done,
all the change requests are executed. So, when swapping two names,
the first change request tries to change the name of A to B, but there
is already a B, so we get a name duplication exception. Then, the
second change request occurs, which is to swap B to A, but A still
exists, so we get a second name duplication exception. This is as expected.

One strange thing is that the error is reported to stderr instead of being
popped up in a dialog. This is because the error occurs in the Swing Event Thread.
I modified the code so that it calls MessageHandler.error() so that the
an error dialog will appear.

PortConfigurerDialog checks to see if there are duplicate names, checking
for swapped names seems rather tricky and error prone. To do this would
require doing a bunch of the name duplication checks that we do anyway.
So, unless we get a more compelling and reproducible test case that brings
up the stack trace, I'm tempted to leave this one as is with the error
message coming up in a dialog and the error message giving a hint about
how to swap names.

**#7 - 10/21/2009 12:04 PM - Oliver Soong**

It's frustrating that we can all see the original bug, but can't reliably replicate it.

As for the port swapping thing, one way to get rid of the port swapping problem is to process all the port changes, then error check, rather than error check at each step. I don't know how hard that is to do, but as long as there's an "original copy" to go back to in case the final fails validation, internal intermediate clashes won't matter.

**#8 - 10/21/2009 02:00 PM - Christopher Brooks**

I modified PortConfigurerDialog._apply() so that if the change request fails,
then the port configurer dialog will close. The reason is that if the
change request fails, then who knows what the problems are with the ports.
It is easier to start over. A better solution would be to have better undo/redo
in the dialog.

**#9 - 03/27/2013 02:26 PM - Redmine Admin**

Original Bugzilla ID was 4478