

## Kepler - Bug #3985

### Types resolved to unacceptable types

04/14/2009 05:46 PM - Oliver Soong

<b>Status:</b>	Resolved	<b>Start date:</b>	04/14/2009
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	ben leinfelder	<b>% Done:</b>	0%
<b>Category:</b>	general	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Unspecified	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	3985		
<b>Description</b>			
URL provides a sample workflow that has been demonstrating the error on my system. Because this problem has been confusingly inconsistent in the past, I'm not 100% convinced it will produce an error for others, but it seems to be causing problems on three builds on my system, two builds in a VM, and two systems in the vislab here, so hopefully I won't be alone this time. It's affecting both 1.0.0 and 1.x dev.			
<b>Related issues:</b>			
Blocked by Kepler - Bug #3986: RExpression confounding R working directory an...		<b>Resolved</b>	<b>04/14/2009</b>

### History

#### #1 - 04/14/2009 05:56 PM - Oliver Soong

This bug is slightly more fun. Change the SDF Director to allow disconnected graphs, delete the relation, then run it. It should work. Now rehook the actors and now it should run as expected. Save the file, close it, and reopen. The bug is back (or never went away, or the cached data that seems to allow things to work has been cleared, or something similar).

#### #2 - 05/04/2009 04:39 PM - Christopher Brooks

I downloaded this from <http://www.nceas.ucsb.edu/~soong/kepler/Unacceptable%20Types%20Problem.xml> and am uploading it so that we have a copy with the bug (urls are notorious for changing).

It looks like an R problem. When I run this, I get:

```
ptolemy.actor.TypeConflictException: Types resolved to unacceptable types in .Unacceptable Types Problem due to the following inequalities:
(ptolemy.actor.TypedIOPort {Unacceptable Types Problem.RExpression.irisdf}, unknown) <= (ptolemy.actor.TypedIOPort {Unacceptable Types Problem.RExpression2.irisdf}, unknown)
```

```
at ptolemy.actor.TypedCompositeActor.resolveTypes(TypedCompositeActor.java:324)
  at ptolemy.actor.Manager.resolveTypes(Manager.java:1058)
  at ptolemy.actor.Manager.preinitializeAndResolveTypes(Manager.java:940)
  at ptolemy.actor.Manager.initialize(Manager.java:601)
  at ptolemy.actor.Manager.execute(Manager.java:338)
  at ptolemy.actor.Manager.run(Manager.java:1109)
  at ptolemy.actor.Manager$3.run(Manager.java:1150)
```

If I explicitly set the type of the irisdf port to integer, then the model runs.

I'm not sure if this is a bug, does R actually propagate types?

#### #3 - 05/05/2009 09:50 AM - Oliver Soong

It seems like the error is thrown before R even runs, in which case it probably wouldn't be a problem with R itself. R does have its own types, which the R actor converts to Ptolemy types. Since the Expression actor also works with undeclared types, there is something in particular about the R actor that the type resolution code doesn't seem to like. Although explicitly specifying the type will work, it's clearly not necessary from a user's perspective (why do Expressions work but RExpressions not) and it imposes constraints on the code users are able to run. This problem doesn't happen every time two R actors are connected, either. I might be able to help more if I knew more about unknown <= unknown, since presumably type = type is allowable.

In playing around a little more, if I set the two port types to int, run the workflow, then set the port types to be blank (unknown?) and change the code to be irisdf <- "1", the token will be converted to a number on output. Setting the output port explicitly to a string will cause another type error that claims the input port is an int type. Ke/Pt is throwing an exception based on outdated information on undeclared types, which is also what the workaround in comment 1 relies upon.

#### #4 - 08/12/2009 11:27 AM - ben leinfelder

from the RExpression actor we won't "know" the type of the port until the R script executes and we're emitting/reading to or from the port.  
Christopher -  
Why does the type lattice not allow two ports that are resolved to "unknown" to be connected and the model to be run? Can we change that?

**#5 - 08/12/2009 12:39 PM - ben leinfelder**

Added a check during preinitialize() that will find unknown-to-unknown connections and set the RExpression unknown-typed output port to "general" so that the type check passes.

**#6 - 03/27/2013 02:25 PM - Redmine Admin**

Original Bugzilla ID was 3985

**Files**

---

UnacceptableTypesProblem.xml	26.4 KB	05/04/2009	Christopher Brooks
------------------------------	---------	------------	--------------------