# Kepler - Bug #4439

## reporting displays output for a different actor

10/06/2009 03:18 PM - Oliver Soong

| | | | |
|---|---|---|---|
| **Status:** | Resolved | **Start date:** | 10/06/2009 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Daniel Crawl | **% Done:** | 0% |
| **Category:** | reporting | **Estimated time:** | 0.00 hour |
| **Target version:** | wrp-modules-2.0.0 | **Spent time:** | 0.00 hour |
| **Bugzilla-Id:** | 4439 | | |

**Description**

I can't upload any useful example, so I'll describe how to generate the problem.

1. In Workflow Editor, add 1 SDF Director (set iterations to 1), 1 RExpression, and 1 Display. Link RExpression.graphicsFileName to the Display.
2. In Report Designer, add RExpression.graphicsFileName to the report.
3. Run and view the report. This should be as expected.
4. In Workflow Editor, delete the RExpression, then run and view the report again. It should be blank, as expected.
5. The sequence of these next steps is rather important. In Workflow Editor, add an RExpression and rename it something, say actor1. I like to change the code to plot(iris[1:2]) to make the plot identifiable.
6. Add another RExpression and rename it something, say actor2. I like to change the code to plot(iris[3:4]).
7. Attach both actor1.graphicsFileName and actor2.graphicsFileName to the Display actor. Run and view the report. This should arguably be blank, yet it is the Sepal plot (actor1).
8. In Workflow Editor, delete actor1. Rerun and view the report. Again, this should be blank, but it is now the Petal plot (actor2).
9. In Workflow Editor, delete actor2. Again, the sequence of these next steps is important. Add 2 RExpression actors (named RExpression and RExpression2). Rename one actor3 and the other actor4. Again, I like to make the code for actor3 plot(iris[1:2]) and for actor4 plot(iris[3:4]).
10. Run and view the report. Again, this should be blank, but it has the Sepal plot (actor3).
11. In Workflow Editor, delete actor3. Rerun and view the report. Now it is blank, as expected.

### History

#### #1 - 10/21/2009 11:48 AM - ben leinfelder

This is looking more and more like an intricacy of Provenance recording. My hunch is that when you first add the RExpression actor to the model it assigns an entityId in provenance. Then, because you rename it before dragging on another RExpression actor it thinks it is the same actor as the first one (and also whenever it is renamed it is still considered "the same" actor in provenance?)

#### #2 - 12/27/2009 10:48 AM - ben leinfelder

More testing makes this look related to provenace:
I create a simple [runnable] workflow so that I can execute it to generate the provenance entries:
1 SDF director
1 RExpression actor
1 Display actor
--run--
there is an ENTITY entry for the RExpression actor
~~rename RExpression actor to "test"~~
~~-drag another RExpression actor to the canvas (still named RExpression like the first renamed one)~~
~~--run-~~
there is only the ENTITY entry for "RExpression" not both "RExpression" and "test"

Perhaps this is how it is intended to work (renaming the actor doesn't actually change the ENTITY table entry)? But it's unclear to me how we can then track the old renamed actor and the newly dropped actor if they don't have independent entries.

Dan, can you advise?

#### #3 - 01/08/2010 03:32 PM - Daniel Crawl

I've updated provenance to handle these types of renames.

It might be nice if reporting displayed an error or warning if the report references entities that do not exist in the workflow. If I delete an actor whose port is in the report, I get an exception:

org.kepler.provenance.QueryException: No tokens found for execution 6
at org.kepler.provenance.sql.SQLQueryV8.getTokensForExecution(SQLQueryV8.java:1304)

at org.kepler.reporting.rio.DynamicReportItem.generateValues(DynamicReportItem.java:197)
at org.kepler.reporting.rio.ReportAssembler.populateReport(ReportAssembler.java:91)
at org.kepler.reporting.rio.ReportAssembler.populateReportWithLatest(ReportAssembler.java:71)
at org.kepler.module.reporting.ReportingListener.writeFiles(ReportingListener.java:138)
at org.kepler.module.reporting.ReportingListener.wrapup(ReportingListener.java:243)

#### #4 - 01/10/2010 10:24 PM - ben leinfelder

i suppose we could be listening for the MOML change request that deletes actors/ports/parameters that are referenced in the reports and making those modifications to the report layout automatically (or issuing a warning that there would now be a broken reference).

#### #5 - 01/11/2010 08:36 AM - Oliver Soong

Why not just check that everything mentioned in the ROML exists just before you create the report (i.e., after execution)? I think of those error messages along the same lines as null pointers.

#### #6 - 01/11/2010 08:39 AM - ben leinfelder

I'm thinking you'd still want the report to render if al but one reference were valid, right? Maybe just a warning that one (or more) references could not be resolved...then carry on with the rendering.

#### #7 - 01/13/2010 03:45 PM - Daniel Crawl

It now prints an error message to stdout and in the report for any unresolved references.

#### #8 - 01/13/2010 03:46 PM - ben leinfelder

oh, neat!

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

Original Bugzilla ID was 4439