

## Kepler - Bug #4869

### changes made to workflows during dialogs before committing (Edit Parameters/Cancel)

03/04/2010 11:04 AM - Oliver Soong

<b>Status:</b>	In Progress	<b>Start date:</b>	03/04/2010
<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>	2.X.Y	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	4869		
<b>Description</b>			
<p>When I make changes to an actor through the dialog, it seems like the changes are made to the workflow immediately, before the "Commit" button is clicked.</p> <p>For example, create a workflow with an EML 2 Dataset actor. Make sure Kepler has not already authenticated with KNB. Now edit the EML 2 Dataset actor to reference a data package that requires authentication. The KNB login window will immediately pop up, even though no changes have been committed. Dismiss the window in some way. Now click the Help button and close the help window. The dialog will have disappeared. Open the actor dialog again, and the change will be there.</p> <p>I should point out that the "Cancel" button does seem to work, but I suspect (based on debugging comments and comments in code) that this acts by reverting the changes already made.</p> <p>I've seen this for sure on EML 2 Dataset and RExpression, since changes to one trigger visual feedback and I've mucked around in the code of the other. I suspect this is a general problem, though, and it may be a result of the underlying ptolemy design.</p>			
<b>Related issues:</b>			
Is duplicate of Kepler - Bug #5029: Parameter changes can be committed withou...		<b>Resolved</b>	<b>05/25/2010</b>

## History

### #1 - 03/10/2010 09:59 AM - ben leinfelder

It's that attributeChanged() method - I know the EML actor does a lot of things in that method, and I remember it being called more times than you might think for each "change". It definitely gets called before hitting "commit" (otherwise, as you mention, we wouldn't be able to change some of the options in the dialog). Not sure we can get away from that behavior, but it'd be nice if when the dialog window "goes away" the changes are not saved (unless "Commit" was pressed, of course).

### #2 - 03/10/2010 02:41 PM - ben leinfelder

from conf. call:  
-verify that "cancel" does work for all normal cases  
-isolate the sequence that caused the changes to persist (layered UI dialogs)  
-if ptolemy-based, investigate undo method.

### #3 - 03/10/2010 04:16 PM - ben leinfelder

You can get into trouble if you Edit the actor parameters then press other buttons (other than Cancel and Commit):  
-Help...then Cancel/Ok in that screen (Edit dialog disappears leaving your changes committed)  
-Remove..shows a screen...then Ok/Cancel will leave the actor with changes  
-Add...same as above

The "Preferences" button displays a dialog, then returns to the actor preferences edit dialog when you dismiss the Preference dialog. This should be the behavior for the other buttons in this dialog (so that you can still choose to undo/cancel your changes).

### #4 - 03/10/2010 06:16 PM - ben leinfelder

This happens when you just run Vergil:  
-drag a StringConst actor to the stage  
-edit the actor params  
-type some value for it  
-hit "Help"  
-close help window  
oops, no more edit dialog and my change is remembered (the value i typed in is now the value even though I never hit commit).

Ptolemy folks want to chime in on whether this is a bug or a feature? I think the Edit dialog should remain after I close the Help dialog so that I can always choose to cancel my actor parameter changes.

#### #5 - 03/11/2010 10:08 AM - Christopher Brooks

It looks like there is some oddness in Vergil with edit parameters.  
I'll see about taking a look.

#### #6 - 03/11/2010 10:22 AM - ben leinfelder

Christopher,  
The `_handleClosing()` method in `ptolemy.actor.gui.EditParametersDialog` does not restore the values of the actor parameters dialog when certain buttons are pushed. It seems like this was intentional, but maybe it should be revisited.

If you do call `contents.restore()` the changes are not committed when, say, the Help window is presented (and subsequently dismissed).

method below:

```
/** If the contents of this dialog implements the CloseListener
 * interface, then notify it that the window has closed.
 */
protected void _handleClosing() {
    super._handleClosing();

    if (!buttonPressed().equals("Commit") && !buttonPressed().equals("Add")
        && !buttonPressed().equals("Preferences")
        && !buttonPressed().equals("Help")
        && !buttonPressed().equals("Remove")) {
        // Restore original parameter values.
        ((Configurer) contents).restore();
    }
}
```

#### #7 - 03/11/2010 11:39 AM - Christopher Brooks

This happens when you just run Vergil:

- drag a `StringConst` actor to the stage
  - edit the actor params
  - type some value for it
  - hit "Help"
  - close help window
- oops, no more edit dialog and my change is remembered (the value i typed in is now the value even though I never hit commit).

Ptolemy folks want to chime in on whether this is a bug or a feature? I think the Edit dialog should remain after I close the Help dialog so that I can always choose to cancel my actor parameter changes.

I checked Ptolemy II 6.0 and the above behavior occurs.

I checked a Web Start version of Ptolemy II 2.0 and the behavior below occurs:

- Remove..shows a screen...then Ok/Cancel will leave the actor with changes

So, I'm fairly certain this is design feature, not a new bug. The code in `EditParameters._handleClosing()` dates from 2000.

The issue is that Edit Parameters is a modal dialog. If the user moves away from the window by clicking a button that brings up another window, what should happen?

Currently, we commit the change, which seems wrong.

An alternative is to close the window, losing the change and bring up the new dialog. That seems wrong as well, since we are losing data.

A better alternative would be to somehow make the dialog non-modal, bring up the new modal dialog, what for the new modal dialog to close and then make the first dialog be modal again. Good luck getting that one right. I for one, do not want to see a change like go out in Kepler-2.0. The odds of getting it right are vanishingly small.

An even better alternative is to get rid of modal dialogs :-)

#### #8 - 03/11/2010 01:48 PM - Christopher Brooks

Edward and I talked about this and he reminded me that changes are actually committed as you move between fields. If you don't want to commit your changes, then hit cancel. I agree that this is somewhat confusing, but it is how the system was designed. I'd take suggestions about how to make this more clear.

The best solution would be to move away from modal dialogs.

An alternative solution would be some sort of system that would bring up the Edit Parameters dialog after the Help, Add or Remove dialog was closed.

Since Ptolemy has had this misbehavior for about 10 years, and this misbehavior is in Kepler-1.0, I move that this bug be either closed or targeted to 2.X.Y. I'll leave it up to someone else to make that call.

**#9 - 03/19/2010 12:01 PM - Chad Berkley**

Moving this bug to 2.x. It is not realistic to fix this for 2.0.

**#11 - 05/25/2010 11:52 AM - Christopher Brooks**

Sean Riddle posted bug 5029, which is a duplicate of this bug, so I closed 5029  
Below is post:

- 1) Double-click on an actor and change a parameter.
- 2) Click the add button to trigger the "add parameter" dialog
- 3) Click Cancel

At this point, you're dumped back to the canvas, and your alteration to the parameter of the actor has been committed. Seems like the preferred behavior is either dumping the user back to the canvas with all changes rolled back, or (even better) just return the user to the normal parameter change dialog after they're done with the add parameter dialog.

I've managed to produce this on the RC3 installer on Windows XP, and on the Kepler trunk on a Mac.

**#12 - 03/27/2013 02:28 PM - Redmine Admin**

Original Bugzilla ID was 4869