

Kepler - Bug #4826

intermittent exceptions displaying matrix tokens

02/23/2010 10:22 AM - Tom Parris

Status:	Resolved	Start date:	02/23/2010
Priority:	Normal	Due date:	
Assignee:	Christopher Brooks	% Done:	0%
Category:	interface	Estimated time:	0.00 hour
Target version:	2.0.0	Spent time:	0.00 hour
Bugzilla-Id:	4826		

Description

I experience intermittent exceptions displaying matrix tokens in a simple workflow. The problems are more frequent when using the runtime window, but I can generate the exceptions either way.

The offending workflow is attached to this message.

Sample of problem while repeatedly pressing the "run workflow " button (normal presentation window).

```
[run] 16 ms. Memory: 58332K Free: 18864K (32%)
[run] 15 ms. Memory: 58332K Free: 17878K (31%)
[run] 16 ms. Memory: 58332K Free: 17001K (29%)
[run] 16 ms. Memory: 58332K Free: 16048K (28%)
[run] 16 ms. Memory: 58332K Free: 18800K (32%)
[run] 16 ms. Memory: 58332K Free: 17876K (31%)
[run] Exception in thread "AWT-EventQueue-0" java.lang.ArrayIndexOutOfBoundsException: 3 >= 2
[run]   at java.util.Vector.elementAt(Vector.java:427)
[run]   at javax.swing.table.DefaultTableColumnModel.getColumn(DefaultTableColumnModel.java:277)
[run]   at javax.swing.plaf.basic.BasicTableUI.paintCells(BasicTableUI.java:1967)
[run]   at javax.swing.plaf.basic.BasicTableUI.paint(BasicTableUI.java:1767)
[run]   at javax.swing.plaf.ComponentUI.update(ComponentUI.java:143)
[run]   at javax.swing.JComponent.paintComponent(JComponent.java:751)
[run]   at javax.swing.JComponent.paint(JComponent.java:1017)
[run]   at javax.swing.JComponent.paintToOffscreen(JComponent.java:5112)
[run]   at javax.swing.RepaintManager$Painter.paintDoubleBuffered(RepaintManager.java:1475)
[run]   at javax.swing.RepaintManager$Painter.paint(RepaintManager.java:1406)
[run]   at javax.swing.RepaintManager.paint(RepaintManager.java:1220)
[run]   at javax.swing.JComponent._paintImmediately(JComponent.java:5060)
[run]   at javax.swing.JComponent.paintImmediately(JComponent.java:4870)
[run]   at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:803)
[run]   at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:714)
[run]   at javax.swing.RepaintManager.seqPaintDirtyRegions(RepaintManager.java:694)
[run]   at javax.swing.SystemEventQueueUtilities$ComponentWorkRequest.run(SystemEventQueueUtilities.java:125)
[run]   at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:209)
[run]   at java.awt.EventQueue.dispatchEvent(EventQueue.java:597)
[run]   at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:269)
[run]   at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:184)
[run]   at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:174)
[run]   at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:169)
[run]   at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:161)
[run]   at java.awt.EventDispatchThread.run(EventDispatchThread.java:122)
```

Sample of problem while repeatedly pressing the "go" button in the "run time window"

```
[run] 15 ms. Memory: 58332K Free: 15914K (27%)
[run] 16 ms. Memory: 58332K Free: 11679K (20%)
[run] 0 ms. Memory: 58332K Free: 10640K (18%)
[run] Exception in thread "AWT-EventQueue-0" java.lang.ArrayIndexOutOfBoundsException: 2 >= 2
[run]   at java.util.Vector.elementAt(Vector.java:427)
[run]   at javax.swing.table.DefaultTableColumnModel.getColumn(DefaultTableColumnModel.java:277)
[run] 16 ms. Memory: 58332K Free: 13221K (23%)
[run]   at javax.swing.plaf.basic.BasicTableUI.paintGrid(BasicTableUI.java:1928)
[run]   at javax.swing.plaf.basic.BasicTableUI.paint(BasicTableUI.java:1764)
[run]   at javax.swing.plaf.ComponentUI.update(ComponentUI.java:143)
```

```
[run] at javax.swing.JComponent.paintComponent(JComponent.java:751)
[run] at javax.swing.JComponent.paint(JComponent.java:1017)
[run] at javax.swing.JComponent.paintToOffscreen(JComponent.java:5112)
[run] at javax.swing.RepaintManager$Painter.paintDoubleBuffered(RepaintManager.java:1475)
[run] at javax.swing.RepaintManager$Painter.paint(RepaintManager.java:1406)
[run] at javax.swing.RepaintManager.paint(RepaintManager.java:1220)
[run] at javax.swing.JComponent._paintImmediately(JComponent.java:5060)
[run] at javax.swing.JComponent.paintImmediately(JComponent.java:4870)
[run] at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:803)
[run] at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:714)
[run] at javax.swing.RepaintManager.seqPaintDirtyRegions(RepaintManager.java:694)
[run] at javax.swing.SystemEventQueueUtilities$ComponentWorkRequest.run(SystemEventQueueUtilities.java:125)
[run] at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:209)
[run] at java.awt.EventQueue.dispatchEvent(EventQueue.java:597)
[run] at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:269)
[run] at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:184)
[run] at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:174)
[run] at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:169)
[run] at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:161)
[run] at java.awt.EventDispatchThread.run(EventDispatchThread.java:122)
```

History

#1 - 02/23/2010 10:23 AM - Tom Parris

Forgot to specify platform.

#2 - 02/23/2010 10:36 AM - Christopher Brooks

I saw this in Ptolemy, so it is a Ptolemy bug, not a Kepler bug. I'll see if I can fix it before the final version of Kepler2.0 comes out.

#3 - 03/01/2010 02:01 PM - Christopher Brooks

This bug does not seem to affect functionality, so I'm moving it to 2.x.y

#4 - 03/01/2010 09:59 PM - Christopher Brooks

Fixed.

The problem was that actor.gui.MatrixPane has a public variable JTable table. This variable should only be used from within the Swing Event Thread. The fix is to call SwingUtilities.invokeLater().

One additional issue is that if the matrix is very large, then the data values are elided to "...". The reason this occurs is because MatrixPane calls `table.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);` in its constructor.

If this was changed to `AUTO_RESIZE_OFF`, then large matrices get horizontal scrollbars.

I looked into doing this and the problem is that the `_pane` in `MatrixViewer` is only used if the `MatrixViewer` is placed in a run control panel. Otherwise, we use the `tableau`. Getting access to the `MatrixPane` via the `Tableau` looks non-trivial.

I've folded this change into the `ptII rel-8-0-beta-2` branch for inclusion in Kepler-2.0.0 and Ptolemy II 8.0.

#5 - 03/27/2013 02:28 PM - Redmine Admin

Original Bugzilla ID was 4826

Files

GE_Element_test.xml

30.7 KB

02/23/2010

Tom Parris