

# Bug ID S-28: Graph X Axis Problem

## Documentation

### Table of contents

1 Problems.....	2
2 Solutions.....	2
3 Modified file list.....	2
4 Code explanation.....	2
4.1 org.simBio.sim.analyzer.graph.Axis.java.....	2
4.2 org.simBio.sim.analyzer.graph.Graph.java.....	3
5 JUnit test.....	3

## 1. Problems

1. When an XML model graph runs, the curve of the graph slowly slips to the left until it moves off the graph. The left part of the curve then appears on the right hand side of the graph. Because of this, the shape of the curve may not be displayed properly, and so cannot be used in comparisons.
2. When the Stop button is pressed, the maximum value of the X axis is set to the final elapsed time. This causes the slippage of the curve to the right side, and causes the same problem as the above (1).
3. When calculating a model with an adaptable time step, the final elapsed time sometimes exceeds the expected elapsed time due to calculation errors. Then, the max and min value of axis X are incremented to the new values, and only the final plot appears on the graph.

## 2. Solutions

1. Increment at even intervals of the X axis, when the data (the value being plotted on the graph) is greater than the max value or smaller than the min value.
2. If the Stop button is pressed before the simulation is completed, don't change the max and min value of the X axis.
3. When the simulation ends without interruption, compare the expected final time with the elapsed time, and if they do not match, adjust the max value of X axis to the final elapsed time.

## 3. Modified file list

- org.simBio.sim.analyzer.graph.Axis.java
- org.simBio.sim.analyzer.graph.Graph.java

## 4. Code explanation

### 4.1. org.simBio.sim.analyzer.graph.Axis.java

In the p method of Axis, the wrong variable had been used to increment the X axis, causing the graph to slip instead of incrementing at even intervals. When the data (the value being plotted on the graph) variable is greater than the max (the maximum value of the X axis) variable, the X axis is incremented by the value of the delta variable. Using data instead of max to calculate the value of delta caused the X axis to gradually slip, because the data value was slightly larger than the max value. Changing this code:

```
double delta = data + max * (extendRate - 1) - min * extendRate;
```

to this:

```
double delta = (max - min) * extendRate;
```

fixed the slippage problem.

The code for incrementing the X axis when the data was less than the min value had the same kind of problem:

```
double delta = data - max * extendRate + min * (extendRate - 1);
```

and so it was changed in the same way.

```
double delta = (min - max) * extendRate;
```

## 4.2. org.simBio.sim.analyzer.graph.Graph.java

The end method of the Graph class is called then the simulation ends whether the Stop button is pressed or not. Then, the new code first checks the following two conditions. If both of these conditions are met, then the X axis is adjusted to match the elapsed time, and the graph is redrawn.

1. Whether the elapsed time is greater than the expected final time  

```
if ((expectedFinalTime < elapsedTime)
```
2. Whether the elapsed time is greater than the X axis maximum. If the elapsed time is less than the X axis maximum, this indicates that the Stop button was pressed before the simulation was completed, and that the end method should not adjust the X axis (problem (2)). If the elapsed time is greater than the X axis maximum, this may cause the problem (3).  

```
&& (elapsedTime > maxNode.getValue())) {
```

The expected final time is calculated in the prepare method, based on the duration node, which is the time that the simulation is expected to run for, and the elapsedTime, which at the start of the simulation is equal to the starting time .

```
expectedFinalTime = getRoot().getNode("duration").getValue() +
    getRoot().getNode("elapsedTime").getValue();
```

## 5. JUnit test

### The test classes

- org.simBio.sim.analyzer.graph.AxisXTest
- org.simBio.sim.analyzer.graph.GraphTest

### The test methods

#### AxisXTest

Method	Test
testPDataBetweenMinAndMax	The input data value is between min and max. After the p method is called, the min and max values of the X axis should remain unchanged.
testPDataLessThanMin	The input data value is less than min. After the p method is called, the min and max values should be changed by delta: $(min - max) * scaleFactor$ .
testPDataGreaterThanMax	The input data value is greater than max. After the p method is called, the min and max values should be changed by delta: $(max - min) * scaleFactor$ .

The class org.simBio.sim.analyzer.graph.AxisXTest.java runs tests on the p method of Axis.java to check three conditions:

1. When the input data value is between the minimum and maximum values of the X axis, the p method should not change the X axis.
2. When the input data value is less than the minimum value of the X axis, the p method should reduce the X axis min and max.
3. When the input data value is greater than the maximum value of the X axis, the p method should increase the X axis min and max.

The min and max values of the X axis after the p method is called are compared to their values before the p method is called, to check whether they have changed by the expected amount. This test uses the stub class GraphStub in place of Graph, in order to test the behaviour of the p method more independently from the rest of the program. The GraphStub.java file is used by the model file axisXTest.xml, which includes only the essential nodes for test purposes.

Extend mode test:

The above tests use the autoscroll mode: when the data is outside the limits of the min and max nodes, the min and max values are adjusted and the graph scrolls sideways automatically. The following tests are the same as the above tests except that they use two different modes: fixed and autoscale.

Method	Test
testPDataBetweenMinAndMaxModeFixed	After the p method is called, the min and max values should stay as they are, because the axis extend mode is fixed.
testPDataLessThanMinModeFixed	After the p method is called, the min and max values should stay as they are, because the axis extend mode is fixed.
testPDataGreaterThanOrEqualToMaxModeFixed	After the p method is called, the min and max values should stay as they are, because the axis extend mode is fixed.
testPDataBetweenMinAndMaxModeAutoscale	After the p method is called, the min and max values should stay as they are, because the data is between min and max.
testPDataLessThanMinModeAutoscale	After the p method is called, the max value should stay as it is and the min value should change. The delta value should be twice as large as it was before the p method was called.
testPDataGreaterThanOrEqualToMaxModeAutoscale	After the p method is called, the max value should change and the min value should stay as it is. The delta value should be twice as large as it was before the p method was called.

In the first 3 methods above, the extendMode value is MODE\_FIXED, and so the min and max values of the X axis stay as they are, regardless of the data value. In the next 3 methods above, the extendMode value is MODE\_AUTOSCALE. If the data is outside the bounds of min and max, either min or max

changes but not both. Therefore the delta value - the interval between min and max - doubles in size.

Extend rate test:

The tests above all use axisXTest.xml, which has a scale factor of 1.0, to create the model instance. The following tests are like those above, but with a scale factor of 0.9 set using the method changeScaleFactorValue.

Method	Test
testPDataLessThanMinLowerExtendRate	After the p method is called, the max and min values should change based on the delta value.
testPDataGreaterThanMaxLowerExtendRate	After the p method is called, the max and min values should change based on the delta value.
testPDataLessThanMinModeAutoscaleLowerExtende	The same as testPDataLessThanMinModeAutoscale, but with a different scalefactor value
testPDataGreaterThanMaxModeAutoscaleLowerExtende	The same as testPDataGreaterThanMaxModeAutoscale, but with a different scalefactor value

In testPDataLessThanMinLowerExtendRate and testPDataGreaterThanMaxLowerExtendRate, there is the same kind of change to the min and max values as there was in the testPDataLessThanMin and testPDataLessThanMax methods, but the size of the change is smaller, because the scale factor is smaller. This is also true of testPDataLessThanMinModeAutoscaleLowerExtendRate and testPDataGreaterThanMaxModeAutoscaleLowerExtendRate when compared with testPDataLessThanMinModeAutoscale and testPDataGreaterThanMaxModeAutoscale.

### GraphTest

Method	Test
testElapsedGreaterThanExpected	If the elapsed time is greater than the expected time, then the elapsed time should be used to adjust the min and max nodes.
testElapsedNotGreaterThanExpected	If the elapsed time is less than or equal to the expected time, then the min and max nodes should not be adjusted in the end() method.

A model instance is created using the file axisXTest.xml, containing a Graph object. In the test methods, the min and max values for the nodes of the X axis after the end() method is called are compared with their expected values.

In the testElapsedNotGreaterThanExpected method, the elapsed time is less than the expected time, so the end() method should not adjust min and max. Therefore, the values of min and max after the end() method is called are expected to be the same as their values before the end method is called.

In the `testElapsedGreaterThanExpected` method, the elapsed time is greater than the expected time, so the max value should be equal to the elapsed time after the end method is called, and the min value should be equal to the elapsed time minus the interval between the min and max values.

### **How to run the tests**

The code has been tested using JUnit 3. To run the tests, right-click on `AxisXTest.java` or `GraphTest.java` in the Eclipse Package Explorer and select [Run As]->[JUnit Test] from the popup menu.