**Graph Pen**

**Test Cases and Procedures**

**Authors:**

Sam Green

Nick Hudson

Stanton Sievers

Jarrod Stormo

# Test Cases

**Create**

**Test Case 1**

**Test Case ID –** RBC.graph.pen.1

**Test Item –** The *pen create* command of the *graph* BLT component.

**Input Specification –** A *name* and any *option*-*value* pairs

**Output Specification –** A pen with *name* and configuration that reflects the *option*-*value* pairs exists

**Special Procedural Requirements –** None

**Inter-case Dependencies –** RBC.graph.pen.4, RBC.graph.pen.5

**Delete**

**Test Case 2**

**Test Case ID –** RBC.graph.pen.2

**Test Item –** The *pen delete* command of the *graph* BLT component.

**Input Specification –** An existing pen

**Output Specification –** The pen should no longer exist

**Special Procedural Requirements –** None

**Inter-case Dependencies –** RBC.graph.pen.1, RBC.graph.pen.4

**Cget**

**Test Case 5**

**Test Case ID –** RBC.graph.pen.5

**Test Item –** The *pen cget* command of the *graph* BLT component.

**Input Specification –** A configuration *option* flag

**Output Specification –** The current value for the *option* flag

**Special Procedural Requirements –** None

**Inter-case Dependencies –** RBC.graph.pen.1, RBC.graph.pen.3

**Configure**

**Test Case 3**

**Test Case ID –** RBC.graph.pen.3

**Test Item –** The *pen configure* command of the *graph* BLT component.

**Input Specification –** A valid configuration *option* flag and *value* pair

**Output Specification –** *pen cget* *-option* should return *value*

**Special Procedural Requirements –** None

**Inter-case Dependencies –** RBC.graph.pen.1, RBC.graph.pen.5

**Test Case 6**

**Test Case ID –** RBC.graph.pen.6

**Test Item –** The *pen configure* command of the *graph* BLT component.

**Input Specification –** A configuration *option* flag

**Output Specification –** Elements on the graph using the pen update to reflect the new configuration values

**Special Procedural Requirements –** None

**Inter-case Dependencies –** RBC.graph.pen.1, RBC.graph.element.2, RBC.graph.element.3

**Delete**

**Test Case 7**

**Test Case ID –** RBC.graph.pen.7

**Test Item –** The *pen delete* command of the *graph* BLT component.

**Input Specification –** A pen

**Output Specification –** The pen is deleted and all elements using the pen still reflect the pen’s configuration (i.e. deleting a pen does not affect any elements using the pen)

**Special Procedural Requirements –** None

**Inter-case Dependencies –** RBC.graph.pen.1, RBC.graph.element.2, RBC.graph.element.3

**Names**

**Test Case 4**

**Test Case ID –** RBC.graph.pen.4

**Test Item –** The *pen names* command of the *graph* BLT component.

**Input Specification –** An optional *pattern*

**Output Specification –** A list of all the names of the current pens that match *pattern*

**Special Procedural Requirements –** None

**Inter-case Dependencies –** RBC.graph.pen.1

# Automated Tests

**Create**

***Test Case 1***

**Test Procedure – Create**

**Purpose –** Ensure that pen creation works when passed only a pen name.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.1.1

**Test Procedure – Create Non-overwriting**

**Purpose –** Ensure that pen creation will not overwrite an existing pen name.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.1.2

**Test Procedure – Default Pens**

**Purpose –** Ensure that two default pens are created automatically when a graph is created

**Special Requirements –** None

**TclTest –** RBC.graph.pen.1.3

**Test Procedure – Creation with Single Option**

**Purpose –** Ensure that pen creation works when passed a single option-value pair.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.1.4

**Test Procedure – Creation with Multiple Options**

**Purpose –** Ensure that pen creation works when passed option-value pairs.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.1.5

**Delete**

***Test Case 2***

**Test Procedure – Deleting Single Pen**

**Purpose –** Ensure that pen deletion works when deleting a single pen.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.2.1

**Test Procedure – Deleting Multiple Pens**

**Purpose –** Ensure that pen deletion works when deleting multiple pens.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.2.2

**Configure**

***Test Case 3***

**Test Procedure – Configure Color**

**Purpose –** Ensure that the color configuration works for valid colors.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.1

**Test Procedure – Configure Dashes**

**Purpose –** Ensure that the dashes configuration works for valid dashes.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.2

**Test Procedure – Configure No Dashes**

**Purpose –** Ensure that the dashes configuration works for no dashes.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.3

**Test Procedure – Configure Fill Color**

**Purpose –** Ensure that the fill configuration works for valid fill colors.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.4

**Test Procedure – Configure No Fill**

**Purpose –** Ensure that the fill configuration works for no fill.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.5

**Test Procedure – Configure Line Width**

**Purpose –** Ensure that the linewidth configuration works for valid linewidths.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.6

**Test Procedure – Configure Off Dash Color**

**Purpose –** Ensure that the offdash configuration works for valid offdash colors.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.7

**Test Procedure – Configure Outline Color**

**Purpose –** Ensure that the outline configuration works for valid outline colors.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.8

**Test Procedure – Configure Outline Default Color**

**Purpose –** Ensure that the outline configuration works for defcolor (the same color as the color configuration option).

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.9

**Test Procedure – Configure Outline Width**

**Purpose –** Ensure that the outlinewidth configuration works for valid widths.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.10

**Test Procedure – Configure No Outline**

**Purpose –** Ensure that the outlinewidth configuration works for no width.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.11

**Test Procedure – Configure Symbol Size**

**Purpose –** Ensure that the pixels configuration works for valid symbol sizes.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.12

**Test Procedure – Configure No Symbol (Size Zero)**

**Purpose –** Ensure that the pixels configuration works for symbole size zero.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.13

**Test Procedure – Configure Square Symbol**

**Purpose –** Ensure that the symbol configuration works for square.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.14

**Test Procedure – Configure Circle Symbol**

**Purpose –** Ensure that the symbol configuration works for circle.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.15

**Test Procedure – Configure Diamond Symbol**

**Purpose –** Ensure that the symbol configuration works for diamond.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.16

**Test Procedure – Configure Plus Symbol**

**Purpose –** Ensure that the symbol configuration works for plus.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.17

**Test Procedure – Configure Cross Symbol**

**Purpose –** Ensure that the symbol configuration works for cross.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.18

**Test Procedure – Configure Splus Symbol**

**Purpose –** Ensure that the symbol configuration works for splus.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.19

**Test Procedure – Configure Scross Symbol**

**Purpose –** Ensure that the symbol configuration works for scross.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.20

**Test Procedure – Configure Triangle Symbol**

**Purpose –** Ensure that the symbol configuration works for triangle.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.21

**Test Procedure – Configure No Symbol**

**Purpose –** Ensure that the symbol configuration works for no symbol.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.22

**Test Procedure – Configure Bitmap Symbol**

**Purpose –** Ensure that the symbol configuration works for bitmap symbols.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.23

**Test Procedure – Configure Default Type**

**Purpose –** Ensure that the default type of a pen is set.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.24

**Test Procedure – Configure Type**

**Purpose –** Ensure that the type configuration works.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.3.25

**Cget**

***Test Case 5***

**Test Procedure – Cget**

**Purpose –** Ensure that cget works with an explicitly set option.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.5.1

**Test Procedure – Cget Default**

**Purpose –** Ensure that cget works with a default value.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.5.2

**Delete**

**Names**

***Test Case 4***

**Test Procedure – Names No Pattern**

**Purpose –** Ensure that names works with no given pattern.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.4.1

**Test Procedure – Names with Pattern**

**Purpose –** Ensure that names works with a pattern parameter.

**Special Requirements –** None

**TclTest –** RBC.graph.pen.4.2

# Manual Tests

**Configure**

***Test Case 6***

**Test Procedure – Configure Color**

**Purpose –** Ensure that the color configuration works for valid colors.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.1.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.1.Body” Tcl command

* Post-Condition – The previously displaying line is now red. This includes the line, the symbols, their outlines and fills, as well as the legend entry.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.1.Cleanup” command

**Test Procedure – Configure Dashes**

**Purpose –** Ensure that the dashes configuration works for valid dashes.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.2.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.2.Body” Tcl command

* Post-Condition – The previously displaying line is now dashed. The dashes are 10 pixels long, while the gaps in between them are only 3 pixels long.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.2.Cleanup” command

**Test Procedure – Configure No Dashes**

**Purpose –** Ensure that the dashes configuration works for no dashes.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.3.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single dashed blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.3.Body” Tcl command

* Post-Condition – The previously displaying dashed line is now solid.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.3.Cleanup” command

**Test Procedure – Configure Fill**

**Purpose –** Ensure that the fill configuration works for valid fill colors.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.4.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.4.Body” Tcl command

* Post-Condition – The previously displaying points are now yellow with a blue outline. The legend also updates to reflect this change.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.4.Cleanup” command

**Test Procedure – Configure No Fill**

**Purpose –** Ensure that the fill configuration works for no fill.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.5.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.5.Body” Tcl command

* Post-Condition – The previously displaying points are no longer filled (i.e. their fill appears to be the background color) with a blue outline. The legend also updates to reflect this change.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.5.Cleanup” command

**Test Procedure – Configure Line Width**

**Purpose –** Ensure that the linewidth configuration works for valid linewidths.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.6.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.6.Body” Tcl command

* Post-Condition – The previously displaying line is now 10 pixels thick, as is the line displaying in the legend.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.6.Cleanup” command

**Test Procedure – Configure Offdash Color**

**Purpose –** Ensure that the offdash configuration works for valid offdash colors.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.7.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single, dashed blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.7.Body” Tcl command

* Post-Condition – The gaps in the line that were previously clear (i.e. the same color as the background) are now red.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.7.Cleanup” command

**Test Procedure – Configure Outline Color**

**Purpose –** Ensure that the outline configuration works for valid outline colors.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.8.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.8.Body” Tcl command

* Post-Condition – The two blue points are now outlined in red.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.8.Cleanup” command

**Test Procedure – Configure Outline Color to Default Color**

**Purpose –** Ensure that the outline configuration works for defcolor (the same color as the *color* configuration option).

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.9.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.9.Body” Tcl command

* Post-Condition – The two blue points, their outlines and fills, as well as the line between the points are now red.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.9.Cleanup” command

**Test Procedure – Configure Outline Width**

**Purpose –** Ensure that the outlinewidth configuration works for valid widths.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.10.Setup” Tcl command
* Pre-Condition – A graph with two blue points outlined in red and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.10.Body” Tcl command

* Post-Condition – The red outlines of the points are now 5 pixels thick.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.10.Cleanup” command

**Test Procedure – Configure Outline Width of Zero**

**Purpose –** Ensure that the outlinewidth configuration works for zero width.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.11.Setup” Tcl command
* Pre-Condition – A graph with two blue points outlined in red and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.11.Body” Tcl command

* Post-Condition – The two blue points no longer have an outline.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.11.Cleanup” command

**Test Procedure – Configure Symbol Size**

**Purpose –** Ensure that the pixels configuration works for valid symbol sizes.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.12.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.12.Body” Tcl command
2. Resize the wish84 slightly by dragging the corner of the window

* Post-Condition – The two blue points are now 30 pixels in diameter.
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.12.Cleanup” command

**Test Procedure – Configure Symbol Size Zero**

**Purpose –** Ensure that the pixels configuration works for symbol size zero.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.13.Setup” Tcl command
* Pre-Condition – A graph with two blue points and a single blue line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.13.Body” Tcl command
2. Resize the wish84 slightly by dragging the corner of the window

* Post-Condition – The two blue points are now 0 pixels in diameter (i.e. not visible)
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.13.Cleanup” command

**Test Procedure – Configure Symbols**

**Purpose –** Ensure that the symbol configuration works for valid symbols.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.6.tcl” file and then call the “graph.pen::RBC.graph.pen.6.14.Setup” Tcl command
* Pre-Condition – A graph with 10 lines, each with 3 points, is now showing. The points should all be circles.
* Body

1. Call the “graph.pen::RBC.graph.pen.6.14.Body” Tcl command

* Post-Condition – The 10 lines, from steepest to shallowest should have the following symbols:
  + An exclamation point
  + Nothing
  + A triangle
  + An x-shaped symbol whose lines are 1 pixel thick
  + A +-shaped symbol whose lines are 1 pixel thick
  + An x-shaped symbol with thick lines
  + A +-shaped symbol with thick lines
  + A diamond
  + A circle
  + A square
* Cleanup – Call the “graph.pen::RBC.graph.pen.6.14.Cleanup” command

**Delete**

***Test Case 7***

**Test Procedure – Pen Deletion**

**Purpose –** Ensure deleting a pen does not affect elements using the pen.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.pen.7.tcl” file and then call the “graph.pen::RBC.graph.pen.7.1.Setup” Tcl command
* Pre-Condition – A graph with two red points and a single red line is showing.
* Body

1. Call the “graph.pen::RBC.graph.pen.7.1.Body” Tcl command

* Post-Condition – A graph with two red points and a single red line is showing.
* Cleanup – Call the “graph.pen::RBC.graph.pen.7.1.Cleanup” command