**Graph Element**

**Test Cases and Procedures**

**Authors:**

Sam Green

Nick Hudson

Stanton Sievers

Jarrod Stormo

# Test Cases

**Cget**

**Test Case 1**

**Test Case ID –** RBC.graph.element.cget.1

**Test Item –** The *element cget* function of the *graph* BLT component.

**Input Specification –** The name of a element configure option.

**Output Specification –** The value of the given element configure option name.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Activate**

**Test Case 1**

**Test Case ID –** RBC.graph.element.activate.1

**Test Item –** The *element activate* function of the *graph* BLT component.

**Input Specification –** An element name.

**Output Specification –** The element name is on the activate list.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Bind**

**Test Case 1**

**Test Case ID –** RBC.graph.element.bind.1

**Test Item –** The *element bind* function of the *graph* BLT component.

**Input Specification –** A tag name and a command.

**Output Specification –** The bindtags contain the name of the tag.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Closest**

**Test Case 1**

**Test Case ID –** RBC.graph.element.closest.1

**Test Item –** The *element closest* function of the *graph* BLT component.

**Input Specification –** An x and y screen coordinate and a variable name in which the results will be stored.

**Output Specification –** The variable contains the name of the closest element, the index of the closest data points, and the graph coordinates of the point. The function returns 0 if no data point within the threshold is found, otherwise it returns 1.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 2**

**Test Case ID –** RBC.graph.element.closest.2

**Test Item –** The *element closest* function of the *graph* BLT component with the *-along* flag.

**Input Specification –** An x and y screen coordinate, a variable name in which the results will be stored, and a direction to the *-along* flag (x, y, both).

**Output Specification –** The variable contains the name of the closest element, the index of the closest data points, and the graph coordinates of the point. The function returns 0 if no data point within the threshold is found, otherwise it returns 1.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 3**

**Test Case ID –** RBC.graph.element.closest.3

**Test Item –** The *element closest* function of the *graph* BLT component with the *-halo* flag.

**Input Specification –** An x and y screen coordinate, a variable name in which the results will be stored, and a pixel value for the *-halo* flag.

**Output Specification –** The variable contains the name of the closest element, the index of the closest data points, and the graph coordinates of the point. The function returns 0 if no data point within the threshold is found, otherwise it returns 1.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 4**

**Test Case ID –** RBC.graph.element.closest.4

**Test Item –** The *element closest* function of the *graph* BLT component with the *-interpolate* flag.

**Input Specification –** An x and y screen coordinate, a variable name in which the results will be stored, and any of the following to the *-along* flag: 1, 0, true, false, yes, no

**Output Specification –** The variable contains the name of the closest element, the index of the closest data points, and the graph coordinates of the point. The function returns 0 if no data point within the threshold is found, otherwise it returns 1.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Configure**

**Test Case 1**

**Test Case ID –** RBC.graph.element.configure.1

**Test Item –** The *element configure -activepen* function of the *graph* BLT component.

**Input Specification –** The name of a pen or “” to not draw the element when it is active.

**Output Specification –** The activepen property of the element component is set to the input value, which will be used to draw the element when it is active.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1,RBC.graph.element.create.1, RBC.graph.pen.1

**Test Case 2**

**Test Case ID –** RBC.graph.element.configure.2

**Test Item –** The *element configure -bindtags* function of the *graph* BLT component.

**Input Specification –** A list of binding tag names.

**Output Specification –** The bindtag list has the given list of tag names.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1, RBC.graph.element.create.1, RBC.graph.element.bind.1

**Test Case 3**

**Test Case ID –** RBC.graph.element.configure.3

**Test Item –** The *element configure -color* function of the *graph* BLT component.

**Input Specification –** A string representation of a color name.

**Output Specification –** The color of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1, RBC.graph.element.create.1

**Test Case 4**

**Test Case ID –** RBC.graph.element.configure.4

**Test Item –** The *element configure -dashes* function of the *graph* BLT component.

**Input Specification –** A list of up to 11 numbers that alternately represent the lengths of the dashes and gaps between the elements.

**Output Specification –** The dash style of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1, RBC.graph.element.create.1

**Test Case 5**

**Test Case ID –** RBC.graph.element.configure.5

**Test Item –** The *element configure -data* function of the *graph* BLT component.

**Input Specification –** A list of numeric expressions representing the X-Y coordinate pairs of each data point.

**Output Specification –** The data property of the element is set to the list of input.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1,RBC.graph.element.create.1

**Test Case 6**

**Test Case ID –** RBC.graph.element.configure.6

**Test Item –** The *element configure -fill* function of the *graph* BLT component.

**Input Specification –** A string representation of a color name, defcolor, or “”.

**Output Specification –** The fill of the element component is set to the input color name, the fill color will use the -color option if given defcolor, or the fill color will be transparent if given “”.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1, RBC.graph.element.create.1

**Test Case 7**

**Test Case ID –** RBC.graph.element.configure.7

**Test Item –** The *element configure -hide* function of the *graph* BLT component.

**Input Specification –** Any of the following: 1, 0, true, false, yes, no

**Output Specification –** The hide property of the element component is set according to the input value (either 1 or 0).

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 8**

**Test Case ID –** RBC.graph.element.configure.8

**Test Item –** The *element configure -label* function of the *graph* BLT component.

**Input Specification –** Text label for the element or “” for no legend entry.

**Output Specification –** The label of the element is set to the input value and is displayed in the legend or no entry is displayed in the legend and the value of the label is empty.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1, RBC.graph.element.create.1

**Test Case 9**

**Test Case ID –** RBC.graph.element.configure.9

**Test Item –** The *element configure -linewidth* function of the *graph* BLT component.

**Input Specification –** A positive numerical value.

**Output Specification –** The linewidth of the element component is set according to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 10**

**Test Case ID –** RBC.graph.element.configure.10

**Test Item –** The *element configure -mapx* function of the *graph* BLT component.

**Input Specification –** The name of a graph axis instance.

**Output Specification –** The element’s x-coordinates are mapped onto the given axis.

**Special Procedural Requirements –** A graph widget can be created, an axis component can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1, RBC.graph.axis.[create]

**Test Case 11**

**Test Case ID –** RBC.graph.element.configure.11

**Test Item –** The *element configure -mapy* function of the *graph* BLT component.

**Input Specification –** The name of a graph axis instance.

**Output Specification –** The element’s y-coordinates are mapped onto the given axis.

**Special Procedural Requirements –** A graph widget can be created, an axis component can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1, RBC.graph.axis.[create]

**Test Case 12**

**Test Case ID –** RBC.graph.element.configure.12

**Test Item –** The *element configure -offdash* function of the *graph* BLT component.

**Input Specification –** A string representation of a color name, defcolor, or “”.

**Output Specification –** The offdash of the element component is set to the input color name, the offdash color will use the -color option if given defcolor, or the offdash pixels will represent gaps instead of stripes if given “”.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1, RBC.graph.element.create.1

**Test Case 13**

**Test Case ID –** RBC.graph.element.configure.13

**Test Item –** The *element configure -outline* function of the *graph* BLT component.

**Input Specification –** A string representation of a color name, defcolor, or “”.

**Output Specification –** The outline of the element component is set to the input color name, the outline color will use the -color option if given defcolor, or the outline will not be drawn if given “”.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1, RBC.graph.element.create.1

**Test Case 14**

**Test Case ID –** RBC.graph.element.configure.14

**Test Item –** The *element configure -pen* function of the *graph* BLT component.

**Input Specification –** The name of a pen.

**Output Specification –** The pen property of the element component is set to the input value, which will be used to draw the element when it is active.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.cget.1,RBC.graph.element.create.1, RBC.graph.pen.1

**Test Case 15**

**Test Case ID –** RBC.graph.element.configure.15

**Test Item –** The *element configure -outlinewidth* function of the *graph* BLT component.

**Input Specification –** A positive numerical value.

**Output Specification –** The outlinewidth of the element component is set according to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 16**

**Test Case ID –** RBC.graph.element.configure.16

**Test Item –** The *element configure -scalesymbols* function of the *graph* BLT component.

**Input Specification –** Any of the following: 1, 0, true, false, yes, no

**Output Specification –** The scalessymbols property of the element component is set according to the input value (either 1 or 0).

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 17**

**Test Case ID –** RBC.graph.element.configure.17

**Test Item –** The *element configure -smooth* function of the *graph* BLT component.

**Input Specification –** Any of the following: linear, step, natural, quadratic

**Output Specification –** The smooth property of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 18**

**Test Case ID –** RBC.graph.element.configure.18

**Test Item –** The *element configure -styles* function of the *graph* BLT component.

**Input Specification –** A list containing a pen name and optionally a minimum and maximum element weight range.

**Output Specification –** The styles property of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 19**

**Test Case ID –** RBC.graph.element.configure.19

**Test Item –** The *element configure -symbol* function of the *graph* BLT component.

**Input Specification –** Any of the following: square, circle, diamond, plus, cross, splus, scross, triangle, “”, bitmap.

**Output Specification –** The symbol property of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 20**

**Test Case ID –** RBC.graph.element.configure.20

**Test Item –** The *element configure -trace* function of the *graph* BLT component.

**Input Specification –** Any of the following: increasing, decreasing, both

**Output Specification –** The trace property of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 21**

**Test Case ID –** RBC.graph.element.configure.21

**Test Item –** The *element configure -weights* function of the *graph* BLT component.

**Input Specification –** A vector name or a list of numeric expressions representing the weights for each data point.

**Output Specification –** The weights property of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 22**

**Test Case ID –** RBC.graph.element.configure.22

**Test Item –** The *element configure -xdata* function of the *graph* BLT component.

**Input Specification –** A vector name or a list of numeric expressions.

**Output Specification –** The xdata property of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 23**

**Test Case ID –** RBC.graph.element.configure.23

**Test Item –** The *element configure -ydata* function of the *graph* BLT component.

**Input Specification –** A vector name or a list of numeric expressions.

**Output Specification –** The ydata property of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Test Case 24**

**Test Case ID –** RBC.graph.element.configure.24

**Test Item –** The *element configure -pixels* function of the *graph* BLT component.

**Input Specification –** A positive number.

**Output Specification –** The pixels property of the element component is set to the input value.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Create**

**Test Case 1**

**Test Case ID –** RBC.graph.element.create.1

**Test Item –** The *element create* function of the *graph* BLT component.

**Input Specification –** A unique name for the element.

**Output Specification –** A new element that can be referred to by the input name.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.names.1

**Deactivate**

**Test Case 1**

**Test Case ID –** RBC.graph.element.deactivate.1

**Test Item –** The *element deactivate* function of the *graph* BLT component.

**Input Specification –** An element name.

**Output Specification –** The element name is not on the activate list.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1, RBC.graph.element.activate.1

**Delete**

**Test Case 1**

**Test Case ID –** RBC.graph.element.delete.1

**Test Item –** The *element delete* function of the *graph* BLT component.

**Input Specification –** One or more element name.

**Output Specification –** The element name is deleted.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1

**Exists**

**Test Case 1**

**Test Case ID –** RBC.graph.element.exists.1

**Test Item –** The *element exists* function of the *graph* BLT component.

**Input Specification –** An element name.

**Output Specification –** 1 if the element is found, 0 otherwise.

**Special Procedural Requirements –** A graph widget can be created.

**Inter-case Dependencies –** RBC.graph.element.create.1, RBC.graph.element.activate.1

**Names**

**Test Case 1**

**Test Case ID –** RBC.graph.element.names.1

**Test Item –** The *element names* function of the *graph* BLT component.

**Input Specification –** A pattern.

**Output Specification –** The names of element objects that match the pattern.

**Special Procedural Requirements –** A graph widget can be created

**Inter-case dependencies –** RBC.graph.element.create.1

**Show**

**Test Case 1**

**Test Case ID –** RBC.graph.element.show.1

**Test Item –** The *element show* function of the *graph* BLT component.

**Input Specification –** One or more element names.

**Output Specification –** The given element names are part of the show list.

**Special Procedural Requirements –** A graph widget can be created

**Inter-case dependencies –** RBC.graph.element.create.1

**Type**

**Test Case 1**

**Test Case ID –** RBC.graph.element.type.1

**Test Item –** The *element type* function of the *graph* BLT component.

**Input Specification –** An element name.

**Output Specification –** One of the following: BarElement, LineElement.

**Special Procedural Requirements –** A graph widget can be created

**Inter-case dependencies –** RBC.graph.element.create.1

# Automated Tests

**Activate**

**Test Procedure – Graph Element Activate: Valid Element Name**

**Test Case 1**

**Purpose –** Ensure the *element activate* command works correctly when given a valid element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.activate.1.1

**Test Procedure – Graph Element Activate: Invalid Element Name**

**Test Case 1**

**Purpose –** Ensure the *element activate* command works correctly when given an invalid element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.activate.1.2

**Bind**

**Test Procedure – Graph Element Bind: Tag, Sequence, Command**

**Test Case 1**

**Purpose –** Ensure the *element bind* command works correctly when given a tag name, an event sequence, and a command.

**Special Requirements –** None

**TclTest –** RBC.graph.element.bind.1.1

**Cget**

**Test Procedure – Graph Element Cget: Valid Option Name**

**Test Case 1**

**Purpose –** Ensure the *element cget* command works correctly when given a valid element configuration option name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.cget.1.1

**Test Procedure – Graph Element Cget: Invalid Option Name**

**Test Case 1**

**Purpose –** Ensure the *crosshairs cget* command works correctly when given an invalid element configuration option name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.cget.1.2

**Closest**

**Test Procedure – Graph Element Closest: Existing Closest Element**

**Test Case 1**

**Purpose –** Ensure the *element closest* command works correctly when a closest element exists.

**Special Requirements –** None

**TclTest –** RBC.graph.element.closest.1.1

**Test Procedure – Graph Element Closest: Existing Closest Element**

**Test Case 1**

**Purpose –** Ensure the *element closest* command works correctly when a closest element does not exist.

**Special Requirements –** None

**TclTest –** RBC.graph.element.closest.1.2

**Test Procedure – Graph Element Closest: Along – X**

**Test Case 2**

**Purpose –** Ensure the *element closest -along* command works correctly when given x as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.closest.2.1

**Test Procedure – Graph Element Closest: Along – Y**

**Test Case 2**

**Purpose –** Ensure the *element closest -along* command works correctly when given y as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.closest.2.2

**Test Procedure – Graph Element Closest: Along – Both**

**Test Case 2**

**Purpose –** Ensure the *element closest -along* command works correctly when given both as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.closest.2.3

**Test Procedure – Graph Element Closest: Along – Invalid Input**

**Test Case 2**

**Purpose –** Ensure the *element closest -along* command works correctly when given invalid input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.closest.2.4

**Test Procedure – Graph Element Closest: Along – Positive Integer Value**

**Test Case 3**

**Purpose –** Ensure the *element closest -halo* command works correctly when given a positive integer value as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.closest.3.1

**Test Procedure – Graph Element Closest: Along – 1**

**Test Case 4**

**Purpose –** Ensure the *element closest -along* command works correctly when given 1 as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.11.1

**Configure**

**Test Procedure – Graph Element Configure: Activepen – Valid Pen Name**

**Test Case 1**

**Purpose –** Ensure the *element configure -activepen* command works correctly when given a valid pen name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.1.1

**Test Procedure – Graph Element Configure: Activepen – Invalid Pen Name**

**Test Case 1**

**Purpose –** Ensure the *element configure -activepen* command works correctly when given an invalid pen name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.1.2

**Test Procedure – Graph Element Configure: Bindtags – Single Tag**

**Test Case 2**

**Purpose –** Ensure the *element configure -bindtags* command works correctly when given a single tag name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.2.1

**Test Procedure – Graph Element Configure: Bindtags – Tag List**

**Test Case 2**

**Purpose –** Ensure the *element configure -bindtags* command works correctly when given a list of tag names.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.2.2

**Test Procedure – Graph Element Configure: Color – Valid Color Name**

**Test Case 3**

**Purpose –** Ensure the *element configure -color* command works correctly when given a valid color name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.3.1

**Test Procedure – Graph Element Configure: Color – Invalid Color Name**

**Test Case 3**

**Purpose –** Ensure the *element configure -color* command works correctly when given an invalid color name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.3.2

**Test Procedure – Graph Element Configure: Dashes – Valid Dash List**

**Test Case 4**

**Purpose –** Ensure the *element configure -dashes* command works correctly when given a valid dash list.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.4.1

**Test Procedure – Graph Element Configure: Dashes – Empty Dash List**

**Test Case 4**

**Purpose –** Ensure the *element configure -dashes* command works correctly when given an empty dash list.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.4.2

**Test Procedure – Graph Element Configure: Dashes – Long Dash List**

**Test Case 4**

**Purpose –** Ensure the *element configure -dashes* command works correctly when given a dash list that is too long.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.4.3

**Test Procedure – Graph Element Configure: Dashes – Invalid Numerical Dash List**

**Test Case 4**

**Purpose –** Ensure the *element configure -dashes* command works correctly when given an invalid numerical dash list.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.4.4

**Test Procedure – Graph Element Configure: Dashes – Dash List with Characters**

**Test Case 4**

**Purpose –** Ensure the *element configure -dashes* command works correctly when given a dash list with characters.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.4.5

**Test Procedure – Graph Element Configure: Data – Valid Coordinate List**

**Test Case 5**

**Purpose –** Ensure the *element configure -data* command works correctly when given a valid coordinate list as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.5.1

**Test Procedure – Graph Element Configure: Data – Invalid Coordinate List**

**Test Case 5**

**Purpose –** Ensure the *element configure -data* command works correctly when given an invalid coordinate list as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.5.2

**Test Procedure – Graph Element Configure: Data – Not Proper Form**

**Test Case 5**

**Purpose –** Ensure the *element configure -data* command works correctly when given input that is not in the proper form (x1 y1 x2 y1).

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.5.3

**Test Procedure – Graph Element Configure: Data – Odd List**

**Test Case 5**

**Purpose –** Ensure the *element configure -data* command works correctly when given input that does not have an even number of values.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.5.4

**Test Procedure – Graph Element Configure: Fill – Valid Color Name**

**Test Case 6**

**Purpose –** Ensure the *element configure -fill* command works correctly when given a valid color name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.6.1

**Test Procedure – Graph Element Configure: Fill – Invalid Color Name**

**Test Case 6**

**Purpose –** Ensure the *element configure -fill* command works correctly when given an invalid color name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.6.2

**Test Procedure – Graph Element Configure: Fill – Defcolor**

**Test Case 6**

**Purpose –** Ensure the *element configure -fill* command works correctly when given ‘defcolor’ as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.6.3

**Test Procedure – Graph Element Configure: Fill – Empty String**

**Test Case 6**

**Purpose –** Ensure the *element configure -fill* command works correctly when given “” as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.6.4

**Test Procedure – Graph Element Configure: Hide – 1**

**Test Case 7**

**Purpose –** Ensure the *element configure -hide* command works correctly when given 1.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.7.1

**Test Procedure – Graph Element Configure: Hide – 0**

**Test Case 7**

**Purpose –** Ensure the *element configure -hide* command works correctly when given 0.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.7.2

**Test Procedure – Graph Element Configure: Hide – True**

**Test Case 7**

**Purpose –** Ensure the *element configure -hide* command works correctly when given true.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.7.3

**Test Procedure – Graph Element Configure: Hide – False**

**Test Case 7**

**Purpose –** Ensure the *element configure -hide* command works correctly when given false.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.7.4

**Test Procedure – Graph Element Configure: Hide – Yes**

**Test Case 7**

**Purpose –** Ensure the *element configure -hide* command works correctly when given yes.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.7.5

**Test Procedure – Graph Element Configure: Hide – No**

**Test Case 7**

**Purpose –** Ensure the *element configure -hide* command works correctly when given no.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.7.6

**Test Procedure – Graph Element Configure: Hide – Invalid Input**

**Test Case 7**

**Purpose –** Ensure the *element configure -hide* command works correctly when given an invalid input value.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.7.7

**Test Procedure – Graph Element Configure: Label – Valid Text**

**Test Case 8**

**Purpose –** Ensure the *element configure -label* command works correctly when given valid text.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.8.1

**Test Procedure – Graph Element Configure: Label – Empty String**

**Test Case 8**

**Purpose –** Ensure the *element configure -label* command works correctly when given the empty string as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.8.2

**Test Procedure – Graph Element Configure: Label – Default Value**

**Test Case 8**

**Purpose –** Ensure the *element configure -label* command does not change the default value when not given any input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.8.3

**Test Procedure – Graph Element Configure: Linewidth – Valid Integer Input**

**Test Case 9**

**Purpose –** Ensure the *element configure -linewidth* command works correctly when given an integer pixel value.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.9.1

**Test Procedure – Graph Element Configure: Linewidth – Valid Decimal Input**

**Test Case 9**

**Purpose –** Ensure the *element configure -linewidth* command works correctly when given a decimal pixel value.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.9.2

**Test Procedure – Graph Element Configure: Linewidth – Invalid Numerical Input**

**Test Case 9**

**Purpose –** Ensure the *element configure -linewidth* command works correctly when given an invalid numerical pixel value.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.9.3

**Test Procedure – Graph Element Configure: Linewidth – Character Input**

**Test Case 9**

**Purpose –** Ensure the *element configure -linewidth* command works correctly when given a character as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.9.4

**Test Procedure – Graph Element Configure: Mapx – Valid Axis Name**

**Test Case 10**

**Purpose –** Ensure the *element configure -mapx* command works correctly when given valid axis name as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.10.1

**Test Procedure – Graph Element Configure: Mapx – Non-Existent Axis Name**

**Test Case 10**

**Purpose –** Ensure the *element configure -mapx* command works correctly when given a non-existent axis name as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.10.2

**Test Procedure – Graph Element Configure: Mapx – No Input**

**Test Case 10**

**Purpose –** Ensure the *element configure -mapx* command works correctly when not input is given.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.10.3

**Test Procedure – Graph Element Configure: Mapy – Valid Axis Name**

**Test Case 11**

**Purpose –** Ensure the *element configure -mapy* command works correctly when given valid axis name as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.11.1

**Test Procedure – Graph Element Configure: Mapy – Non-Existent Axis Name**

**Test Case 11**

**Purpose –** Ensure the *element configure -mapy* command works correctly when given a non-existent axis name as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.11.2

**Test Procedure – Graph Element Configure: Mapy – No Input**

**Test Case 11**

**Purpose –** Ensure the *element configure -mapy* command works correctly when not input is given.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.11.3

**Test Procedure – Graph Element Configure: Offdash – Valid Color Name**

**Test Case 12**

**Purpose –** Ensure the *element configure -offdash* command works correctly when given a valid color name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.12.1

**Test Procedure – Graph Element Configure: Offdash – Invalid Color Name**

**Test Case 12**

**Purpose –** Ensure the *element configure -offdash* command works correctly when given an invalid color name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.12.2

**Test Procedure – Graph Element Configure: Offdash – Defcolor**

**Test Case 12**

**Purpose –** Ensure the *element configure -offdash* command works correctly when given ‘defcolor’ as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.12.3

**Test Procedure – Graph Element Configure: Offdash – Empty String**

**Test Case 12**

**Purpose –** Ensure the *element configure -offdash* command works correctly when given “” as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.12.4

**Test Procedure – Graph Element Configure: Outline – Valid Color Name**

**Test Case 13**

**Purpose –** Ensure the *element configure -outline* command works correctly when given a valid color name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.13.1

**Test Procedure – Graph Element Configure: Outline – Invalid Color Name**

**Test Case 13**

**Purpose –** Ensure the *element configure -outline* command works correctly when given an invalid color name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.13.2

**Test Procedure – Graph Element Configure: Outline – Defcolor**

**Test Case 13**

**Purpose –** Ensure the *element configure -outline* command works correctly when given ‘defcolor’ as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.13.3

**Test Procedure – Graph Element Configure: Pen – Valid Pen Name**

**Test Case 14**

**Purpose –** Ensure the *element configure -pen* command works correctly when given a valid pen name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.14.1

**Test Procedure – Graph Element Configure: Pen – Invalid Pen Name**

**Test Case 14**

**Purpose –** Ensure the *element configure -pen* command works correctly when given an invalid pen name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.14.2

**Test Procedure – Graph Element Configure: Outlinewidth – Valid Integer Input**

**Test Case 15**

**Purpose –** Ensure the *element configure -outlinewidth* command works correctly when given an integer pixel value.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.15.1

**Test Procedure – Graph Element Configure: Outlinewidth – Valid Decimal Input**

**Test Case 15**

**Purpose –** Ensure the *element configure -outlinewidth* command works correctly when given a decimal pixel value.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.15.2

**Test Procedure – Graph Element Configure: Outlinewidth – Invalid Numerical Input**

**Test Case 15**

**Purpose –** Ensure the *element configure -outlinewidth* command works correctly when given an invalid numerical pixel value.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.15.3

**Test Procedure – Graph Element Configure: Outlinewidth – Character Input**

**Test Case 15**

**Purpose –** Ensure the *element configure -outlinewidth* command works correctly when given a character as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.15.4

**Test Procedure – Graph Element Configure: Scalesymbols – 1**

**Test Case 16**

**Purpose –** Ensure the *element configure -scalesymbols* command works correctly when given 1.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.16.1

**Test Procedure – Graph Element Configure: Scalesymbols – 0**

**Test Case 16**

**Purpose –** Ensure the *element configure -scalesymbols* command works correctly when given 0.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.16.2

**Test Procedure – Graph Element Configure: Scalesymbols – True**

**Test Case 16**

**Purpose –** Ensure the *element configure -scalesymbols* command works correctly when given true.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.16.3

**Test Procedure – Graph Element Configure: Scalesymbols – False**

**Test Case 16**

**Purpose –** Ensure the *element configure -scalesymbols* command works correctly when given false.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.16.4

**Test Procedure – Graph Element Configure: Scalesymbols – Yes**

**Test Case 16**

**Purpose –** Ensure the *element configure -scalesymbols* command works correctly when given yes.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.16.5

**Test Procedure – Graph Element Configure: Scalesymbols – No**

**Test Case 16**

**Purpose –** Ensure the *element configure -scalesymbols* command works correctly when given no.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.16.6

**Test Procedure – Graph Element Configure: Scalesymbols – Invalid Input**

**Test Case 16**

**Purpose –** Ensure the *element configure -scalesymbols* command works correctly when given an invalid input value.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.16.7

**Test Procedure – Graph Element Configure: Smooth – Valid Input**

**Test Case 17**

**Purpose –** Ensure the *element configure -smooth* command works correctly when given valid input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.17.1

**Test Procedure – Graph Element: Smooth – Invalid Input**

**Test Case 17**

**Purpose –** Ensure the *element configure -smooth* command works correctly when given invalid input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.17.2

**Test Procedure – Graph Element: Stylelist – Valid Pen Name**

**Test Case 18**

**Purpose –** Ensure the *element configure -styles* command works correctly when given a valid pen name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.18.1

**Test Procedure – Graph Element: Stylelist – Non-Existent Pen**

**Test Case 18**

**Purpose –** Ensure the *element configure -styles* command works correctly when given an invalid pen name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.18.2

**Test Procedure – Graph Element Configure: Stylelist – Pen and Weights**

**Test Case 18**

**Purpose –** Ensure the *element configure -styles* command works correctly when given a valid pen name and a minimum and maximum element weight range.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.18.3

**Test Procedure – Graph Element Configure: Symbol – Valid Symbol Name**

**Test Case 19**

**Purpose –** Ensure the *element configure -symbol* command works correctly when given a valid symbol name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.19.1

**Test Procedure – Graph Element Configure: Symbol – Invalid Symbol Name**

**Test Case 19**

**Purpose –** Ensure the *element configure -symbol* command works correctly when given an invalid symbol name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.19.2

**Test Procedure – Graph Element Configure: Trace – Valid Direction**

**Test Case 20  
Purpose –** Ensure the *element configure -trace* command works correctly when given a valid direction.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.20.1

**Test Procedure – Graph Element Configure: Trace – Invalid Direction**

**Test Case 20**

**Purpose –** Ensure the *element configure -trace* command works correctly when given an invalid direction.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.20.2

**Test Procedure – Graph Element Configure: Weights – Valid Vector Name**

**Test Case 21**

**Purpose –** Ensure the *element configure -weights* command works correctly when given a valid vector name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.21.1

**Test Procedure – Graph Element Configure: Weights – Invalid Vector Name**

**Test Case 21**

**Purpose –** Ensure the *element configure -weights* command works correctly when given an invalid vector name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.21.2

**Test Procedure – Graph Element Configure: Weights – Valid List**

**Test Case 21**

**Purpose –** Ensure the *element configure -weights* command works correctly when given a valid list of numerical values.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.21.3

**Test Procedure – Graph Element Configure: Weights – Invalid Input**

**Test Case 21**

**Purpose –** Ensure the *element configure -weights* command works correctly when given an invalid list.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.21.4

**Test Procedure – Graph Element Configure: Xdata – Valid Vector Name**

**Test Case 22**

**Purpose –** Ensure the *element configure -xdata* command works correctly when given a valid vector name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.22.1

**Test Procedure – Graph Element Configure: Xdata – Invalid Vector Name**

**Test Case 22**

**Purpose –** Ensure the *element configure -xdata* command works correctly when given an invalid vector name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.22.2

**Test Procedure – Graph Element Configure: Xdata – Valid List**

**Test Case 22**

**Purpose –** Ensure the *element configure -xdata* command works correctly when given a valid list of numerical values.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.22.3

**Test Procedure – Graph Element Configure: Xdata – Invalid Input**

**Test Case 22**

**Purpose –** Ensure the *element configure -xdata* command works correctly when given an invalid list.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.22.4

**Test Procedure – Graph Element Configure: Ydata – Valid Vector Name**

**Test Case 23**

**Purpose –** Ensure the *element configure -ydata* command works correctly when given a valid vector name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.23.1

**Test Procedure – Graph Element Configure: Ydata – Invalid Vector Name**

**Test Case 23**

**Purpose –** Ensure the *element configure -ydata* command works correctly when given an invalid vector name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.23.2

**Test Procedure – Graph Element Configure: Ydata – Valid List**

**Test Case 23**

**Purpose –** Ensure the *element configure -ydata* command works correctly when given a valid list of numerical values.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.23.3

**Test Procedure – Graph Element Configure: Ydata – Invalid Input**

**Test Case 23**

**Purpose –** Ensure the *element configure -ydata* command works correctly when given an invalid list.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.23.4

**Test Procedure – Graph Element Configure: Pixels – Positive Integer**

**Test Case 24**

**Purpose –** Ensure the *element configure -pixels* command works correctly when given a positive integer.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.24.1

**Test Procedure – Graph Element Configure: Pixels – Positive Decimal**

**Test Case 24**

**Purpose –** Ensure the *element configure -pixels* command works correctly when given a positive decimal.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.24.2

**Test Procedure – Graph Element Configure: Pixels – Invalid Numerical Input**

**Test Case 24**

**Purpose –** Ensure the *element configure -pixels* command works correctly when given invalid numerical input (e.g. a negative value).

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.24.3

**Test Procedure – Graph Element Configure: Pixels – Character Input**

**Test Case 24**

**Purpose –** Ensure the *element configure -pixels* command works correctly when given characters as input.

**Special Requirements –** None

**TclTest –** RBC.graph.element.configure.24.4

**Create**

**Test Procedure – Graph Element Create: Unique Name**

**Test Case 1**

**Purpose –** Ensure the *element create* command works correctly when given a unique element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.create.1.1

**Test Procedure – Graph Element Create: Existing Name**

**Test Case 1**

**Purpose –** Ensure the *element create* command works correctly when given an existing element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.create.1.2

**Deactivate**

**Test Procedure – Graph Element Deactivate: Valid Element Name**

**Test Case 1**

**Purpose –** Ensure the *element deactivate* command works correctly when given a valid element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.deactivate.1.1

**Test Procedure – Graph Element Deactivate: Invalid Element Name**

**Test Case 1**

**Purpose –** Ensure the *element deactivate* command works correctly when given an invalid element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.deactivate.1.2

**Delete**

**Test Procedure – Graph Element Delete: Single Element Name**

**Test Case 1**

**Purpose –** Ensure the *element delete* command works correctly when given a single element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.delete.1.1

**Test Procedure – Graph Element Delete: Multiple Element Names**

**Test Case 1**

**Purpose –** Ensure the *element delete* command works correctly when given multiple element names.

**Special Requirements –** None

**TclTest –** RBC.graph.element.delete.1.2

**Test Procedure – Graph Element Delete: Invalid Element Name**

**Test Case 1**

**Purpose –** Ensure the *element delete* command works correctly when given an invalid element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.delete.1.3

**Exists**

**Test Procedure – Graph Element Exists: Existing Element Name**

**Test Case 1**

**Purpose –** Ensure the *element delete* command works correctly when given an existing element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.exists.1.1

**Test Procedure – Graph Element Exists: Non-Existent Element Name**

**Test Case 1**

**Purpose –** Ensure the *element delete* command works correctly when given a non-existent element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.exists.1.2

**Names**

**Test Procedure – Graph Element Names: No Pattern**

**Test Case 1**

**Purpose –** Ensure the *element names* command works correctly when no pattern is given.

**Special Requirements –** None

**TclTest –** RBC.graph.element.names.1.1

**Test Procedure – Graph Element Names: Exact Pattern**

**Test Case 1**

**Purpose –** Ensure the *element names* command works correctly when an exact pattern is given.

**Special Requirements –** None

**TclTest –** RBC.graph.element.names.1.2

**Test Procedure – Graph Element Names: Wildcard Pattern**

**Test Case 1**

**Purpose –** Ensure the *element names* command works correctly when a wildcard pattern is given.

**Special Requirements –** None

**TclTest –** RBC.graph.element.names.1.3

**Test Procedure – Graph Element Names: Incorrect Pattern**

**Test Case 1**

**Purpose –** Ensure the *element names* command works correctly when an incorrect pattern is given.

**Special Requirements –** None

**TclTest –** RBC.graph.element.names.1.4

**Show**

**Test Procedure – Graph Element Show: Single Element Name**

**Test Case 1**

**Purpose –** Ensure the *element show* command works correctly when given a single element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.show.1.1

**Test Procedure – Graph Element Show: Multiple Element Names**

**Test Case 1**

**Purpose –** Ensure the *element show* command works correctly when given multiple element names.

**Special Requirements –** None

**TclTest –** RBC.graph.element.show.1.2

**Test Procedure – Graph Element Show: Non-Existent Element Name**

**Test Case 1**

**Purpose –** Ensure the *element show* command works correctly when given a non-existent element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.show.1.3

**Type**

**Test Procedure – Graph Element Type: Valid Line Element Name**

**Test Case 1**

**Purpose –** Ensure the *element type* command works correctly when given the name of a valid line element.

**Special Requirements –** None

**TclTest –** RBC.graph.element.type.1.1

**Test Procedure – Graph Element Type: Valid Bar Element Name**

**Test Case 1**

**Purpose –** Ensure the *element type* command works correctly when given the name of a valid bar element.

**Special Requirements –** None

**TclTest –** RBC.graph.element.type.1.2

**Test Procedure – Graph Element Type: Invalid Element Name**

**Test Case 1**

**Purpose –** Ensure the *element type* command works correctly when given an invalid element name.

**Special Requirements –** None

**TclTest –** RBC.graph.element.type.1.3

# Manual Tests

None written