**Graph Component**

**Manual Test Procedures**

**Authors:**

Sam Green

Nick Hudson

Stanton Sievers

Jarrod Stormo

Contents

[Axis 4](#_Toc218666576)

[Configure 4](#_Toc218666577)

[Bar 4](#_Toc218666578)

[Configure 5](#_Toc218666579)

[Crosshairs 17](#_Toc218666580)

[Cget 17](#_Toc218666581)

[Configure 17](#_Toc218666582)

[Off 17](#_Toc218666583)

[On 17](#_Toc218666584)

[Toggle 17](#_Toc218666585)

[Element 17](#_Toc218666586)

[Cget 17](#_Toc218666587)

[Configure 17](#_Toc218666588)

[Activate 17](#_Toc218666589)

[Bind 17](#_Toc218666590)

[Closest 17](#_Toc218666591)

[Deactivate 17](#_Toc218666592)

[Delete 17](#_Toc218666593)

[Exists 17](#_Toc218666594)

[Names 17](#_Toc218666595)

[Show 17](#_Toc218666596)

[Type 17](#_Toc218666597)

[Extents 17](#_Toc218666598)

[Grid 17](#_Toc218666599)

[Cget 17](#_Toc218666600)

[Configure Color 17](#_Toc218666601)

[Off 17](#_Toc218666602)

[On 17](#_Toc218666603)

[Toggle 18](#_Toc218666604)

[Inside 18](#_Toc218666605)

[Invtransform 18](#_Toc218666606)

[Legend 18](#_Toc218666607)

[Marker 18](#_Toc218666608)

[Pen 18](#_Toc218666609)

[Postscript 18](#_Toc218666610)

[Snap 18](#_Toc218666611)

[Transform 18](#_Toc218666612)

[Test Case 1 18](#_Toc218666613)

# Axis

## Configure

Test Case 1

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

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

**Input Specification –** A name value pair for the configuration options

**Output Specification –** The configuration options being changed on the graph

**Special Procedural Requirements –** None

**Inter-case Dependencies –** None

### Test Procedure – Configure Aspect

**Purpose –** Ensure configuring aspect changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.1.Setup” Tcl command
* Pre-Condition – There is a blank graph displayed on the screen
* Body
* Call the “graph.configure::RBC.graph.configure.1.1.Body” Tcl command
* Post-Condition – The graph shrinks in vertically in half to a different aspect ratio
* Cleanup – Call the “graph.configure::RBC.graph.configure.1.1.Cleanup” command

### Test Procedure – Configure Background

**Purpose –** Ensure configuring background changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

Setup – Call the “graph.configure::RBC.graph.configure.1.2.Setup” Tcl command

Pre-Condition – There is a blank graph displayed on the screen

Body

Call the “graph.configure::RBC.graph.configure.1.2.Body” Tcl command

Post-Condition – The graph axis background changes to a red color

Cleanup – Call the “graph.configure::RBC.graph.configure.1.2.Cleanup” command

# Bar

### Test Procedure – Add Bar Element

**Test Case 1**

**Purpose –** Ensure bar components may be added to a line graph.

**Special Requirements –** None

**Script –** RBC.graph.bar.1.tcl

**Procedural Steps**

* Setup – Call the “graph.bar::RBC.graph.bar.1.1.Setup” Tcl command
* Pre-Condition – There is a graph with line data displayed on the graph
* Body

1. Call the “graph.bar::RBC.graph.bar.1.1.Body” Tcl command

* Post-Condition – The graph now includes bar data displaying.
* Cleanup – Call the “graph.bar:::RBC.graph.bar.1.1.Cleanup” command

# Configure

### Test Procedure – Configure Aspect

**Test Case 1**

**Purpose –** Ensure configuring aspect changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.1.Setup” Tcl command
* Pre-Condition – There is a blank graph displayed on the screen
* Body

1. Call the “graph.configure::RBC.graph.configure.1.1.Body” Tcl command

* Post-Condition – The graph shrinks in vertically in half to a different aspect ratio
* Cleanup – Call the “graph.configure::RBC.graph.configure.1.1.Cleanup” command

### Test Procedure – Configure Background

**Test Case 1**

**Purpose –** Ensure configuring background changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.2.Setup” Tcl command
* Pre-Condition – There is a blank graph displayed on the screen
* Body

1. Call the “graph.configure::RBC.graph.configure.1.2.Body” Tcl command

* Post-Condition – The graph axis background changes to a red color

Cleanup – Call the “graph.configure::RBC.graph.configure.1.2.Cleanup” command

### Test Procedure – Configure Barmode

**Test Case 1**

**Purpose –** Ensure configuring barmode changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.3.Setup” Tcl command
* Pre-Condition – The graph has bar data with two stacks on the third data point
* Body

1. Call the “graph.configure::RBC.graph.configure.1.3.Body” Tcl command

* Post-Condition – The two stacks are placed side by side rather than on top of each other

Cleanup – Call the “graph.configure::RBC.graph.configure.1.3.Cleanup” command

### Test Procedure – Configure Barwidth

**Test Case 1**

**Purpose –** Ensure configuring barwidth changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.4.Setup” Tcl command
* Pre-Condition – There is a graph display a bar chart
* Body

1. Call the “graph.configure::RBC.graph.configure.1.4.Body” Tcl command

* Post-Condition – The bar are stretched across the screen to be extremely wide

Cleanup – Call the “graph.configure::RBC.graph.configure.1.4.Cleanup” command

### Test Procedure – Configure Baseline (Doesn’t Work)

**Test Case 1**

**Purpose –** Ensure configuring baseline changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.5.Setup” Tcl command
* Pre-Condition – The graph has bar data with two stacks on the third data point
* Body

1. Call the “graph.configure::RBC.graph.configure.1.5.Body” Tcl command

* Post-Condition – The two stacks are placed side by side rather than on top of each other

Cleanup – Call the “graph.configure::RBC.graph.configure.1.5.Cleanup” command

### Test Procedure – Configure Borderwidth (Doesn’t Change output when value changes)

**Test Case 1**

**Purpose –** Ensure configuring barmode changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.6.Setup” Tcl command
* Pre-Condition – There is a blank graph displayed
* Body

1. Call the “graph.configure::RBC.graph.configure.1.6.Body” Tcl command

* Post-Condition – The axis are moved closer inward towards the graph

Cleanup – Call the “graph.configure::RBC.graph.configure.1.6.Cleanup” command

### Test Procedure – Configure Bottommargin

**Test Case 1**

**Purpose –** Ensure configuring bottommargin changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.7.Setup” Tcl command
* Pre-Condition – There is a graph display of a bar chart
* Body

1. Call the “graph.configure::RBC.graph.configure.1.7.Body” Tcl command

* Post-Condition – The bottom axis is shifted down vertically

Cleanup – Call the “graph.configure::RBC.graph.configure.1.7.Cleanup” command

### Test Procedure – Configure Bottomvariable (No idea how this works)

**Test Case 1**

**Purpose –** Ensure configuring bottomvariable changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.8.Setup” Tcl command
* Pre-Condition – There is a graph displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.8.Body” Tcl command

* Post-Condition – ???

Cleanup – Call the “graph.configure::RBC.graph.configure.1.8.Cleanup” command

### Test Procedure – Configure Cursors

**Purpose –** Ensure configuring cursors changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.9.Setup” Tcl command
* Pre-Condition – There is a graph display a bar chart using default cursors (crosshair)
* Body

1. Call the “graph.configure::RBC.graph.configure.1.9.Body” Tcl command

* Post-Condition – The cursor changes into a circle when moving across graph

Cleanup – Call the “graph.configure::RBC.graph.configure.1.9.Cleanup” command

### Test Procedure – Configure Data (No idea how this works)

**Test Case 1**

**Purpose –** Ensure configuring data changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.10.Setup” Tcl command
* Pre-Condition – There is a graph displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.10.Body” Tcl command

* Post-Condition – ???

Cleanup – Call the “graph.configure::RBC.graph.configure.10.Cleanup” command

### Test Procedure – Configure Datacommand (No idea how this works)

**Test Case 1**

**Purpose –** Ensure configuring datacommand changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.11.Setup” Tcl command
* Pre-Condition – There is a graph displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.11.Body” Tcl command

* Post-Condition – ???

Cleanup – Call the “graph.configure::RBC.graph.configure.1.11.Cleanup” command

### Test Procedure – Configure Font

**Test Case 1**

**Purpose –** Ensure configuring font changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.12.Setup” Tcl command
* Pre-Condition – There is a graph with a title displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.12.Body” Tcl command

* Post-Condition – The title font changes into a smaller variant

Cleanup – Call the “graph.configure::RBC.graph.configure.1.12.Cleanup” command

### Test Procedure – Configure Foreground

**Test Case 1**

**Purpose –** Ensure configuring foreground changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.13.Setup” Tcl command
* Pre-Condition – There is a graph with a title displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.13.Body” Tcl command

* Post-Condition – The title font change to be the color yellow

Cleanup – Call the “graph.configure::RBC.graph.configure.1.13.Cleanup” command

### Test Procedure – Configure Height

**Test Case 1**

**Purpose –** Ensure configuring height changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.14.Setup” Tcl command
* Pre-Condition – There is a graph is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.14.Body” Tcl command

* Post-Condition – The height shrinks to a slim vertical distance

Cleanup – Call the “graph.configure::RBC.graph.configure.1.14.Cleanup” command

### Test Procedure – Configure HighlightBackground (No effect)

**Test Case 1**

**Purpose –** Ensure configuring highlightbackground changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.15.Setup” Tcl command
* Pre-Condition – There is a graph is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.15.Body” Tcl command

* Post-Condition – ???

Cleanup – Call the “graph.configure::RBC.graph.configure.1.15.Cleanup” command

### Test Procedure – Configure Highlightcolor (No effect)

**Test Case 1**

**Purpose –** Ensure configuring highlightcolor changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.16.Setup” Tcl command
* Pre-Condition – There is a graph is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.16.Body” Tcl command

* Post-Condition – ???

Cleanup – Call the “graph.configure::RBC.graph.configure.1.16.Cleanup” command

### Test Procedure – Configure Highlightthickness (No effect)

**Test Case 1**

**Purpose –** Ensure configuring highlightthickness changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.17.Setup” Tcl command
* Pre-Condition – There is a graph is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.157Body” Tcl command

* Post-Condition – ???

Cleanup – Call the “graph.configure::RBC.graph.configure.1.17.Cleanup” command

### Test Procedure – Configure Invertxy

**Test Case 1**

**Purpose –** Ensure configuring invertxy changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.18.Setup” Tcl command
* Pre-Condition – There is a graph with axis labels on (X on bottom Y on left)
* Body

1. Call the “graph.configure::RBC.graph.configure.1.18.Body” Tcl command

* Post-Condition – The axis labels flip axis (X on left, Y on bottom)

Cleanup – Call the “graph.configure::RBC.graph.configure.1.18.Cleanup” command

### Test Procedure – Configure Justify

**Test Case 1**

**Purpose –** Ensure configuring justify changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.19.Setup” Tcl command
* Pre-Condition – There is a graph with a title with the second line centered below the first
* Body

1. Call the “graph.configure::RBC.graph.configure.1.19.Body” Tcl command

* Post-Condition – The title’s second line of text is now aligned on the left side

Cleanup – Call the “graph.configure::RBC.graph.configure.1.19.Cleanup” command

### Test Procedure – Configure Leftmargin

**Test Case 1**

**Purpose –** Ensure configuring leftmargin changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.20.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying on screen
* Body

1. Call the “graph.configure::RBC.graph.configure.1.20.Body” Tcl command

* Post-Condition – The left side of the graph is shifted to the left

Cleanup – Call the “graph.configure::RBC.graph.configure.1.20.Cleanup” command

### Test Procedure – Configure Leftvariable (No idea how this works)

**Test Case 1**

**Purpose –** Ensure configuring leftvariable changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.21.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying on screen
* Body

1. Call the “graph.configure::RBC.graph.configure.1.21.Body” Tcl command

* Post-Condition – ???

Cleanup – Call the “graph.configure::RBC.graph.configure.1.21.Cleanup” command

### Test Procedure – Configure Plotbackground

**Test Case 1**

**Purpose –** Ensure configuring plotbackground changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.22.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying on screen with a white plot area
* Body

1. Call the “graph.configure::RBC.graph.configure.1.22.Body” Tcl command

* Post-Condition – The plot area changes to red

Cleanup – Call the “graph.configure::RBC.graph.configure.1.22.Cleanup” command

### Test Procedure – Configure Plotborderwidth

**Test Case 1**

**Purpose –** Ensure configuring plotborderwidth changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.23.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying on screen
* Body

1. Call the “graph.configure::RBC.graph.configure.1.23.Body” Tcl command

* Post-Condition – The border around the white plot area increases to make it look sunken

Cleanup – Call the “graph.configure::RBC.graph.configure.1.23.Cleanup” command

### Test Procedure – Configure Plotpadx

**Test Case 1**

**Purpose –** Ensure configuring plotpadx changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.24.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.24.Body” Tcl command

* Post-Condition – The plot area horizontally expands making the bottom axis look smaller and compressed

Cleanup – Call the “graph.configure::RBC.graph.configure.1.24.Cleanup” command

### Test Procedure – Configure Plotpady

**Test Case 1**

**Purpose –** Ensure configuring plotpadx changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.25.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.25.Body” Tcl command

* Post-Condition – The plot area vertically expands making the axis numbers look smaller and compressed

Cleanup – Call the “graph.configure::RBC.graph.configure.1.25.Cleanup” command

### Test Procedure – Configure Plotrelief

**Test Case 1**

**Purpose –** Ensure configuring plotrelief changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.26.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.26.Body” Tcl command

* Post-Condition – The plot area becomes raised instead of sunken

Cleanup – Call the “graph.configure::RBC.graph.configure.1.26.Cleanup” command

### Test Procedure – Configure Relief

**Test Case 1**

**Purpose –** Ensure configuring relief changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.27.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.27.Body” Tcl command

* Post-Condition – The whole graph becomes raised instead of sunken

Cleanup – Call the “graph.configure::RBC.graph.configure.1.27.Cleanup” command

### Test Procedure – Configure Rightmargin

**Test Case 1**

**Purpose –** Ensure configuring rightmargin changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.28.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.28.Body” Tcl command

* Post-Condition – The right side of the graph shrinks inwards

Cleanup – Call the “graph.configure::RBC.graph.configure.1.28.Cleanup” command

### Test Procedure – Configure Rightvariable (No idea how this works)

**Test Case 1**

**Purpose –** Ensure configuring rightvariable changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.29.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying on screen
* Body

1. Call the “graph.configure::RBC.graph.configure.1.29.Body” Tcl command

* Post-Condition – ???

Cleanup – Call the “graph.configure::RBC.graph.configure.1.29.Cleanup” command

### Test Procedure – Configure Shadow (No effect)

**Test Case 1**

**Purpose –** Ensure configuring shadow changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.30.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying
* Body

1. Call the “graph.configure::RBC.graph.configure.1.30.Body” Tcl command

* Post-Condition – The graph has a red shadow behind it

Cleanup – Call the “graph.configure::RBC.graph.configure.1.30.Cleanup” command

### Test Procedure – Configure Takefocus

**Test Case 1**

**Purpose –** Ensure configuring takefocus changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.31.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying with a button. The graph cannot be switched to using tab.
* Body

1. Call the “graph.configure::RBC.graph.configure.1.31.Body” Tcl command
2. Press tab several times to switch focus between graph and button

* Post-Condition – The graph can now take the focus when pressing tab

Cleanup – Call the “graph.configure::RBC.graph.configure.1.31.Cleanup” command

### Test Procedure – Configure Tile

**Test Case 1**

**Purpose –** Ensure configuring tile changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.32.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying.
* Body

1. Call the “graph.configure::RBC.graph.configure.1.32.Body” Tcl command

* Post-Condition – The plot area changes background to a buckskin image

Cleanup – Call the “graph.configure::RBC.graph.configure.1.32.Cleanup” command

### Test Procedure – Configure Title

**Test Case 1**

**Purpose –** Ensure configuring title changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.33.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying.
* Body

1. Call the “graph.configure::RBC.graph.configure.1.33.Body” Tcl command

* Post-Condition – The graph now displays a title (“Test”) above the plot area

Cleanup – Call the “graph.configure::RBC.graph.configure.1.33.Cleanup” command

### Test Procedure – Configure Topmargin

**Test Case 1**

**Purpose –** Ensure configuring topmargin changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.34.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying.
* Body

1. Call the “graph.configure::RBC.graph.configure.1.34.Body” Tcl command

* Post-Condition – The top of the graph moves down due to bigger top margin

Cleanup – Call the “graph.configure::RBC.graph.configure.1.34.Cleanup” command

### Test Procedure – Configure Topmargin

**Test Case 1**

**Purpose –** Ensure configuring topmargin changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.34.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying.
* Body

1. Call the “graph.configure::RBC.graph.configure.1.34.Body” Tcl command

* Post-Condition – The top of the graph moves down due to bigger top margin

Cleanup – Call the “graph.configure::RBC.graph.configure.1.34.Cleanup” command

### Test Procedure – Configure Topvariable (No idea how this works)

**Test Case 1**

**Purpose –** Ensure configuring topvariable changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.35.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying.
* Body

1. Call the “graph.configure::RBC.graph.configure.1.35.Body” Tcl command

* Post-Condition – The top of the graph moves down due to bigger top margin

Cleanup – Call the “graph.configure::RBC.graph.configure.1.35.Cleanup” command

### Test Procedure – Configure Width

**Test Case 1**

**Purpose –** Ensure configuring width changes on screen.

**Special Requirements –** None

**Script –** RBC.graph.configure.1.tcl

**Procedural Steps**

* Setup – Call the “graph.configure::RBC.graph.configure.1.36.Setup” Tcl command
* Pre-Condition – A blank graph window is displaying.
* Body

1. Call the “graph.configure::RBC.graph.configure.1.36.Body” Tcl command

* Post-Condition – The width of the graph shrinks to a slim horizontal distance

Cleanup – Call the “graph.configure::RBC.graph.configure.1.36.Cleanup” command

# Crosshairs

## Cget

## Configure

## Off

### Test Procedure – Graph Crosshairs Off

**Test Case 1**

**Purpose –** Ensure the crosshairs can be hidden on a graph.

**Special Requirements –** None

**Script –** RBC.graph.crosshairs.off.1.tcl

**Procedural Steps**

* Setup – Call the “graph.Crosshairs:RBC.graph.crosshairs.off.1.1.Setup” Tcl command
* Pre-Condition – There is a graph with the grid currently displaying
* Body

1. Call the “graph.Crosshairs:RBC.graph.crosshairs.off.1.1.Body” Tcl command

* Post-Condition – The crosshairs on the graph are hidden
* Cleanup – Call the “graph.Crosshairs:RBC.graph.crosshairs.off.1.1.Cleanup” command

## On

### Test Procedure – Graph Crosshairs On

**Test Case 1**

**Purpose –** Ensure the crosshairs can be displayed on a graph.

**Special Requirements –** None

**Script –** RBC.graph.crosshairs.on.1.tcl

**Procedural Steps**

* Setup – Call the “graph.Crosshairs:RBC.graph.crosshairs.on.1.1.Setup” Tcl command
* Pre-Condition – There is a graph with the crosshairs currently hidden
* Body

1. Call the “graph.Crosshairs:RBC.graph.grid.configure.7.1.Body” Tcl command

* Post-Condition – The crosshairs on the graph are visible
* Cleanup – Call the “graph.Crosshairs:RBC.graph.crosshairs.on.1.1.Cleanup” command

## Toggle

### Test Procedure – Graph Crosshairs Toggle 1

**Test Case 1**

**Purpose –** Ensure the crosshairs can be toggled on a graph.

**Special Requirements –** None

**Script –** RBC.graph.crosshairs.toggle.1.tcl

**Procedural Steps**

* Setup – Call the “graph.Crosshairs:RBC.graph.crosshairs.toggle.1.1.Setup” Tcl command
* Pre-Condition – There is a graph with the crosshairs currently displaying
* Body

1. Call the “graph.Crosshairs:RBC.graph.crosshairs.toggle.1.1.Body” Tcl command

* Post-Condition – The crosshairs on the graph are hidden
* Cleanup – Call the “graph.Crosshairs:RBC.graph.crosshairs.toggle.1.1.Cleanup” command

### Test Procedure – Graph Crosshairs Toggle 2

**Test Case 1**

**Purpose –** Ensure the crosshairs can be toggled on a graph.

**Special Requirements –** None

**Script –** RBC.graph.crosshairs.toggle.1.tcl

**Procedural Steps**

* Setup – Call the “graph.grid::RBC.graph.crosshairs.toggle.1.2.Setup” Tcl command
* Pre-Condition – There is a graph with the crosshairs currently hidden
* Body

1. Call the “graph.Crosshairs:RBC.graph.crosshairs.toggle.1.2.Body” Tcl command

* Post-Condition – The crosshairs on the graph are visible
* Cleanup – Call the “graph.Crosshairs:RBC.graph.crosshairs.toggle.1.2.Cleanup” command

# Element

## Cget

## Configure

## Activate

## Bind

## Closest

## Deactivate

## Delete

## Exists

## Names

## Show

## Type

# Extents

### Test Procedure – Leftmargin

**Test Case 1**

**Purpose –** Ensure the graph extents are valid for leftmargin.

**Special Requirements –** None

**Script –** RBC.graph.extents.1.tcl

**Procedural Steps**

* Setup – Call the “graph.extents::RBC.graph.extents.1.1.Setup” Tcl command
* Pre-Condition – None
* Body

1. Call the “graph.extents::RBC.graph.extents.1.1.Body” Tcl command

* Post-Condition – Command returns 20
* Cleanup – Call the “graph.extents::RBC.graph.extents.1.1.Cleanup” command

### Test Procedure – Rightmargin

**Test Case 1**

**Purpose –** Ensure the graph extents are valid for rightmargin.

**Special Requirements –** None

**Script –** RBC.graph.extents.1.tcl

**Procedural Steps**

* Setup – Call the “graph.extents::RBC.graph.extents.1.2.Setup” Tcl command
* Pre-Condition – None
* Body

1. Call the “graph.extents::RBC.graph.extents.1.2.Body” Tcl command

* Post-Condition – Command returns 20
* Cleanup – Call the “graph.extents::RBC.graph.extents.1.2.Cleanup” command

### Test Procedure – Topmargin

**Test Case 1**

**Purpose –** Ensure the graph extents are valid for topmargin.

**Special Requirements –** None

**Script –** RBC.graph.extents.1.tcl

**Procedural Steps**

* Setup – Call the “graph.extents::RBC.graph.extents.1.3.Setup” Tcl command
* Pre-Condition – None
* Body

1. Call the “graph.extents::RBC.graph.extents.1.3.Body” Tcl command

* Post-Condition – Command returns 20
* Cleanup – Call the “graph.extents::RBC.graph.extents.1.3.Cleanup” command

### Test Procedure – Bottommargin

**Test Case 1**

**Purpose –** Ensure the graph extents are valid for bottommargin.

**Special Requirements –** None

**Script –** RBC.graph.extents.1.tcl

**Procedural Steps**

* Setup – Call the “graph.extents::RBC.graph.extents.1.4.Setup” Tcl command
* Pre-Condition – None
* Body

1. Call the “graph.extents::RBC.graph.extents.1.4.Body” Tcl command

* Post-Condition – Command returns 20
* Cleanup – Call the “graph.extents::RBC.graph.extents.1.4.Cleanup” command

### Test Procedure – Plotwidth

**Test Case 1**

**Purpose –** Ensure the graph extents are valid for plotwidth.

**Special Requirements –** None

**Script –** RBC.graph.extents.1.tcl

**Procedural Steps**

* Setup – Call the “graph.extents::RBC.graph.extents.1.5.Setup” Tcl command
* Pre-Condition – None
* Body

1. Call the “graph.extents::RBC.graph.extents.1.5.Body” Tcl command

* Post-Condition – Command returns 2
* Cleanup – Call the “graph.extents::RBC.graph.extents.1.5.Cleanup” command

### Test Procedure – Plotheight

**Test Case 1**

**Purpose –** Ensure the graph extents are valid for plotheight.

**Special Requirements –** None

**Script –** RBC.graph.extents.1.tcl

**Procedural Steps**

* Setup – Call the “graph.extents::RBC.graph.extents.1.6.Setup” Tcl command
* Pre-Condition – None
* Body

1. Call the “graph.extents::RBC.graph.extents.1.6.Body” Tcl command

* Post-Condition – Command returns 2
* Cleanup – Call the “graph.extents::RBC.graph.extents.1.6.Cleanup” command

# Grid

## Cget

## Configure Color

## Off

### Test Procedure – Graph Grid Off

**Purpose –** Ensure the grid can be hidden on a graph.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.grid.off.1.tcl” file and then call the “graph.grid::RBC.graph.grid.off.1.1.Setup” Tcl command
* Pre-Condition – There is a graph with the grid currently displaying
* Body

1. Call the “graph.grid::RBC.graph.grid.off.1.1.Body” Tcl command

* Post-Condition – The grid on the graph is hidden
* Cleanup – Call the “graph.grid::RBC.graph.grid.off.1.1.Cleanup” command

## On

### Test Procedure – Graph Grid On

**Purpose –** Ensure the grid can be displayed on a graph.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.grid.on.1.tcl” file and then call the “graph.grid::RBC.graph.grid.on.1.1.Setup” Tcl command
* Pre-Condition – There is a graph with the grid currently hidden
* Body

1. Call the “graph.grid::RBC.graph.grid.on.1.1.Body” Tcl command

* Post-Condition – The grid on the graph is visible
* Cleanup – Call the “graph.grid::RBC.graph.grid.on.1.1.Cleanup” command

## Toggle

### Test Procedure – Graph Grid Toggle: Off

**Purpose –** Ensure the grid can be toggled on a graph.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.grid.toggle.1.tcl” file and then call the “graph.grid::RBC.graph.grid.toggle.1.1.Setup” Tcl command
* Pre-Condition – There is a graph with the grid currently displaying
* Body

1. Call the “graph.grid::RBC.graph.grid.toggle.1.1.Body” Tcl command

* Post-Condition – The grid on the graph is hidden
* Cleanup – Call the “graph.grid::RBC.graph.grid.toggle.1.1.Cleanup” command

### Test Procedure – Graph Grid Toggle: On

**Purpose –** Ensure the grid can be toggled on a graph.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.grid.toggle.1.tcl” file and then call the “graph.grid::RBC.graph.grid.toggle.1.2.Setup” Tcl command
* Pre-Condition – There is a graph with the grid currently hidden
* Body

1. Call the “graph.grid::RBC.graph.grid.toggle.1.2.Body” Tcl command

* Post-Condition – The grid on the graph is visible
* Cleanup – Call the “graph.grid::RBC.graph.grid.toggle.1.2.Cleanup” command

# Inside

### Test Procedure – Inside: Not Inside

**Purpose –** Ensure that inside returns 0 for when it is outside

**Special Requirements –** None

**Script –** RBC.graph.inside.1.tcl

**Procedural Steps**

* Setup – Call the “graph.inside::RBC.graph.inside.1.1.Setup” Tcl command
* Pre-Condition – A graph is displayed
* Body

1. Call the “graph.inside::RBC.graph.inside.1.1.Body” Tcl command

* Post-Condition – Command returns 0
* Cleanup – Call the “graph.inside::RBC.graph.inside.1.1.Cleanup” command

### Test Procedure – Inside: Inside

**Purpose –** Ensure that inside returns 1 for when it is inside the plot.

**Special Requirements –** None

**Script –** RBC.graph.inside.1.tcl

**Procedural Steps**

* Setup – Call the “graph.inside::RBC.graph.inside.1.2.Setup” Tcl command
* Pre-Condition – A graph is displayed
* Body

1. Call the “graph.inside::RBC.graph.inside.1.2.Body” Tcl command

* Post-Condition – Command returns 1
* Cleanup – Call the “graph.inside::inside.graph.inside.1.2.Cleanup” command

# Invtransform

### Test Procedure – Invtransform

**Purpose –** Ensure the invtransform command works correctly.

**Special Requirements –** None

**Script –** RBC.graph.invtransform.1.tcl

**Procedural Steps**

* Setup – Call the “graph.invtransform::RBC.graph.invtransform.1.1.Setup” Tcl command
* Pre-Condition – A graph is displayed
* Body

1. Call the “graph.invtransform::RBC.graph.invtransform.1.1.Body” Tcl command

* Post-Condition – Command returns -0.0765550239234 0.972392638037
* Cleanup – Call the “graph.invtransform::RBC.graph.invtransform.1.1.Cleanup” command

# Legend

## Test Case 6

### Test Procedure – Activate All

**Purpose –** Ensure that activating all legend elements works properly.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.6.tcl” file and then call the “graph.legend::RBC.graph.legend.6.1.Setup” Tcl command
* Pre-Condition – A graph with two elements, Line1 and Line2, is showing. The two elements appear deactivated in the legend.
* Body

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

* Post-Condition – In the legend, Line1 and Line2 appear activated (i.e. their color has changed to dark gray).
* Cleanup – Call the “graph.legend::RBC.graph.legend.6.1.Cleanup” command

### Test Procedure – Activate a Subset

**Purpose –** Ensure that activating a subset of legend elements works properly.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.6.tcl” file and then call the “graph.legend::RBC.graph.legend.6.2.Setup” Tcl command
* Pre-Condition – A graph with two elements, Line1 and Line2, is showing. The two elements appear deactivated in the legend.
* Body

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

* Post-Condition – In the legend, Line1 and only Line1 appears activated (i.e. its color has changed to dark gray).
* Cleanup – Call the “graph.legend::RBC.graph.legend.6.2.Cleanup” command

## Test Case 7

### Test Procedure – Deactivate All

**Purpose –** Ensure that deactivating all legend elements works properly.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.7.tcl” file and then call the “graph.legend::RBC.graph.legend.7.1.Setup” Tcl command
* Pre-Condition – A graph with two elements, Line1 and Line2, is showing. The two elements appear activated in the legend.
* Body

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

* Post-Condition – In the legend, Line1 and Line2 appear deactivated (i.e. their color has changed to light gray).
* Cleanup – Call the “graph.legend::RBC.graph.legend.7.1.Cleanup” command

### Test Procedure – Deactivate a Subset

**Purpose –** Ensure that deactivating a subset of legend elements works properly.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.7.tcl” file and then call the “graph.legend::RBC.graph.legend.7.2.Setup” Tcl command
* Pre-Condition – A graph with two elements, Line1 and Line2, is showing. The two elements appear activated in the legend.
* Body

1. Call the “graph.legend::RBC.graph.legend.7.2.Body” Tcl command

* Post-Condition – In the legend, Line1 and only Line1 appears deactivated (i.e. its color has changed to light gray).
* Cleanup – Call the “graph.legend::RBC.graph.legend.7.2.Cleanup” command

## Test Case 8

### Test Procedure – Configure Active Background

**Purpose –** Ensure that deactivating a subset of legend elements works properly.

**Special Requirements –** Legend elements must be able to be activated.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.1.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.1.Body” Tcl command

* Post-Condition – The legend element’s background is now a salmon color.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.1.Cleanup” command

### Test Procedure – Configure Active Border Width

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

**Special Requirements –** Legend elements must be able to be activated.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.2.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.2.Body” Tcl command

* Post-Condition – The legend element’s border is at a width of 20 pixels.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.2.Cleanup” command

### Test Procedure – Configure Active Foreground

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

**Special Requirements –** Legend elements must be able to be activated.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.3.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.3.Body” Tcl command

* Post-Condition – The legend element’s text is now white.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.3.Cleanup” command

### Test Procedure – Configure Active Relief Raised

**Purpose –** Ensure that the activerelief configuration works for raised reliefs.

**Special Requirements –** Legend elements must be able to be activated. Activeborderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.4.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated. The legend element should also have an active border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.4.Body” Tcl command

* Post-Condition – The legend element’s relief is now raised.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.4.Cleanup” command

### Test Procedure – Configure Active Relief Flat

**Purpose –** Ensure that the activerelief configuration works for flat reliefs.

**Special Requirements –** Legend elements must be able to be activated. Activeborderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.5.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated. The legend element should also have active border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.5.Body” Tcl command

* Post-Condition – The legend element’s relief is now flat.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.5.Cleanup” command

### Test Procedure – Configure Active Relief Grooved

**Purpose –** Ensure that the activerelief configuration works for grooved reliefs.

**Special Requirements –** Legend elements must be able to be activated. Activeborderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.6.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated. The legend element should also have an active border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.6.Body” Tcl command

* Post-Condition – The legend element’s relief is now grooved.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.6.Cleanup” command

### Test Procedure – Configure Active Relief Ridged

**Purpose –** Ensure that the activerelief configuration works for ridged reliefs.

**Special Requirements –** Legend elements must be able to be activated. Activeborderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.7.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated. The legend element should also have an active border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.7.Body” Tcl command

* Post-Condition – The legend element’s relief is now ridged.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.7.Cleanup” command

### Test Procedure – Configure Active Relief Solid

**Purpose –** Ensure that the activerelief configuration works for solid reliefs.

**Special Requirements –** Legend elements must be able to be activated. Activeborderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.8.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated. The legend element should also have an active border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.8.Body” Tcl command

* Post-Condition – The legend element’s relief is now solid.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.8.Cleanup” command

### Test Procedure – Configure Active Relief Sunken

**Purpose –** Ensure that the activerelief configuration works for sunken reliefs.

**Special Requirements –** Legend elements must be able to be activated. Activeborderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.9.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and the element in the legend is activated. The legend element should also have an active border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.9.Body” Tcl command

* Post-Condition – The legend element’s relief is now sunken.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.9.Cleanup” command

### Test Procedure – Configure All Anchors

**Purpose –** Ensure that the anchor configuration works for all anchors.

**Special Requirements –** The legend must be able to be positioned in the plot area. As the bodies are executed, take note of the position of the legend.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.10.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing. The legend is in the top center of the plot area.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.10.Body1” Tcl command
2. Call the “graph.legend::RBC.graph.legend.8.10.Body2” Tcl command
3. Call the “graph.legend::RBC.graph.legend.8.10.Body3” Tcl command
4. Call the “graph.legend::RBC.graph.legend.8.10.Body4” Tcl command
5. Call the “graph.legend::RBC.graph.legend.8.10.Body5” Tcl command
6. Call the “graph.legend::RBC.graph.legend.8.10.Body6” Tcl command
7. Call the “graph.legend::RBC.graph.legend.8.10.Body7” Tcl command
8. Call the “graph.legend::RBC.graph.legend.8.10.Body8” Tcl command
9. Call the “graph.legend::RBC.graph.legend.8.10.Body9” Tcl command

* Post-Condition – The legend moved around the plot area in the following order:
  + Center
  + North Center
  + Northeast
  + East Center
  + Southeast
  + South Center
  + Southwest
  + West Center
  + Northwest
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.10.Cleanup” command

### Test Procedure – Configure No Background

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

**Special Requirements –** Legend background must be able to be set for valid colors

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.11.Setup” Tcl command
* Pre-Condition – A graph with a single element is displaying. The graph’s background color is salmon and the legend’s background is gray.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.11.Body” Tcl command

* Post-Condition – The legend’s background is now salmon.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.11.Cleanup” command

### Test Procedure – Configure Valid Background Color

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.12.Setup” Tcl command
* Pre-Condition – A graph with a single element is displaying. The legend’s background is gray.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.12.Body” Tcl command

* Post-Condition – The legend’s background is now black.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.12.Cleanup” command

### Test Procedure – Configure Border Width

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.13.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.13.Body” Tcl command

* Post-Condition – The legend element’s border is at a width of 20 pixels.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.13.Cleanup” command

### Test Procedure – Configure Font

**Purpose –** Ensure Ensure that the font configuration works for valid font strings.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.14.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.14.Body” Tcl command

* Post-Condition – The legend element’s border is now Arial, bold, 14 point.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.14.Cleanup” command

### Test Procedure – Configure Foreground

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.15.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.15.Body” Tcl command

* Post-Condition – The legend element’s text is now white.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.15.Cleanup” command

### Test Procedure – Configure Hide

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.16.Setup” Tcl command
* Pre-Condition – A graph with a single element and the legend are showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.16.Body” Tcl command

* Post-Condition – The legend is now hidden.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.16.Cleanup” command

### Test Procedure – Configure Ipadx

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.17.Setup” Tcl command
* Pre-Condition – A graph with a single element and the legend are showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.17.Body” Tcl command

* Post-Condition – The legend’s internal padding on the left and right sides is now 5 pixels wide.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.17.Cleanup” command

### Test Procedure – Configure Ipady

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.18.Setup” Tcl command
* Pre-Condition – A graph with a single element and the legend are showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.18.Body” Tcl command

* Post-Condition – The legend’s internal padding on the top and bottom is now 5 pixels wide.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.18.Cleanup” command

### Test Procedure – Configure Padx

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.19.Setup” Tcl command
* Pre-Condition – A graph with a single element and the legend are showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.19.Body” Tcl command

* Post-Condition – The legend’s external padding on the left and right sides is now 5 pixels wide.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.19.Cleanup” command

### Test Procedure – Configure Pady

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.20.Setup” Tcl command
* Pre-Condition – A graph with a single element and the legend are showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.20.Body” Tcl command

* Post-Condition – The legend’s internal padding on the top and bottom is now 5 pixels wide.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.20.Cleanup” command

### Test Procedure – Configure Relief Raised

**Purpose –** Ensure that the relief configuration works for raised reliefs.

**Special Requirements –** Borderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.21.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing. The legend element should also have a border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.21.Body” Tcl command

* Post-Condition – The legend element’s relief is now raised.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.21.Cleanup” command

### Test Procedure – Configure Relief Flat

**Purpose –** Ensure that the relief configuration works for flat reliefs.

**Special Requirements –** Borderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.22.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing. The legend element should also have border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.22.Body” Tcl command

* Post-Condition – The legend element’s relief is now flat.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.22.Cleanup” command

### Test Procedure – Configure Relief Grooved

**Purpose –** Ensure that the relief configuration works for grooved reliefs.

**Special Requirements –** Borderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.23.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing. The legend element should also have an border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.23.Body” Tcl command

* Post-Condition – The legend element’s relief is now grooved.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.23.Cleanup” command

### Test Procedure – Configure Relief Ridged

**Purpose –** Ensure that the relief configuration works for ridged reliefs.

**Special Requirements –** Borderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.24.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing. The legend element should also have an border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.24.Body” Tcl command

* Post-Condition – The legend element’s relief is now ridged.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.24.Cleanup” command

### Test Procedure – Configure Relief Solid

**Purpose –** Ensure that the relief configuration works for solid reliefs.

**Special Requirements –** Borderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.25.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing. The legend element should also have an border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.25.Body” Tcl command

* Post-Condition – The legend element’s relief is now solid.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.25.Cleanup” command

### Test Procedure – Configure Relief Sunken

**Purpose –** Ensure that the relief configuration works for sunken reliefs.

**Special Requirements –** Borderwidth must be able to be set.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.26.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing. The legend element should also have an border width of 10.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.26.Body” Tcl command

* Post-Condition – The legend element’s relief is now sunken.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.26.Cleanup” command

### Test Procedure – Configure Shadow

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

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.27.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.27.Body” Tcl command

* Post-Condition – The legend element’s text now has a red shadow.
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.27.Cleanup” command

### Test Procedure – Configure Shadow and Depth

**Purpose –** Ensure that the shadow configuration works for a shadow and a depth.

**Special Requirements –** None

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.8.tcl” file and then call the “graph.legend::RBC.graph.legend.8.28.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing.
* Body

1. Call the “graph.legend::RBC.graph.legend.8.28.Body” Tcl command

* Post-Condition – The legend element’s text now has a red shadow that is 3 pixels behind the text (i.e. it is offset to the bottom right of the text 3 pixels).
* Cleanup – Call the “graph.legend::RBC.graph.legend.8.28.Cleanup” command

## Test Case 9

### Test Procedure – Binding Creation

**Purpose –** Ensure that bindings can be created for a legend element.

**Special Requirements –** Legend elements may be activated and deactivated.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.9.tcl” file and then call the “graph.legend::RBC.graph.legend.9.1.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and it is deactivated.
* Body

1. Call the “graph.legend::RBC.graph.legend.9.1.Body” Tcl command
2. Double-left-click the legend element.

* Post-Condition – The legend element is now activated.
* Cleanup – Call the “graph.legend::RBC.graph.legend.9.1.Cleanup” command

### Test Procedure – Bind Append

**Purpose –** Ensure that bindings can be appended for a legend element.

**Special Requirements –** Legend elements may be activated and deactivated.

**Procedural Steps**

* Setup – Run the “RBC.graph.legend.9.tcl” file and then call the “graph.legend::RBC.graph.legend.9.2.Setup” Tcl command
* Pre-Condition – A graph with a single element is showing and it is deactivated.
* Body

1. Call the “graph.legend::RBC.graph.legend.9.2.Body” Tcl command
2. Double-left-click the legend element.
3. Double-right-click the legend element.

* Post-Condition – The legend element is activated when double-left-clicked and deactivated when double-right-click.
* Cleanup – Call the “graph.legend::RBC.graph.legend.9.2.Cleanup” command

# Marker

## Test Case 1

### Test Procedure – Graph Text Marker Appears

**Purpose –** Ensure creating a text marker actually creates the marker on the graph.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.1.2.Setup” Tcl command
* Pre-Condition – There is a blank graph on the Tk window
* Body
  1. Run “graph.marker::RBC.graph.marker.1.2.Body” Tcl command
* Post-Condition - A text marker appears on the screen
* Cleanup - Run “graph.marker::RBC.graph.marker.1.2.Cleanup” command

### Test Procedure – Graph Line Marker Appears

**Purpose –** Ensure creating a line marker actually creates the marker on the graph.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.1.3.Setup” Tcl command
* Pre-Condition – There is a blank graph on the Tk window
* Body

1. Run “graph.marker::RBC.graph.marker.1.3.Body” Tcl command

* Post-Condition - A line marker appears on the screen from the middle to upper right corner
* Cleanup - Run “graph.marker::RBC.graph.marker.1.3.Cleanup” command

### Test Procedure – Graph Bitmap Marker Appears

**Purpose –** Ensure creating a bitmap marker actually creates the marker on the graph.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.1.4.Setup” Tcl command
* Pre-Condition – There is a blank graph on the Tk window
* Body

1. Run “graph.marker::RBC.graph.marker.1.4.Body” Tcl command

* Post-Condition - A bitmap image appears on the screen
* Cleanup - Run “graph.marker::RBC.graph.marker.1.4.Cleanup” command

### Test Procedure – Graph Image Marker Appears

**Purpose –** Ensure creating a image marker actually creates the marker on the graph.

**Special Requirements –** The greenback.xbm file is in the interpreter directory

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.1.5.Setup” Tcl command
* Pre-Condition – There is a blank graph on the Tk window
* Body

1. Run “graph.marker::RBC.graph.marker.1.5.Body” Tcl command

* Post-Condition – An image appears on the screen
* Cleanup - Run “graph.marker::RBC.graph.marker.1.5.Cleanup” command

### Test Procedure – Graph Polygon Marker Appears

**Purpose –** Ensure creating a polygon marker actually creates the marker on the graph.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.1.6.Setup” Tcl command
* Pre-Condition – There is a blank graph on the Tk window
* Body

1. Run “graph.marker::RBC.graph.marker.1.6.Body” Tcl command

* Post-Condition - A triangle in the upper right appears on the screen
* Cleanup - Run “graph.marker::RBC.graph.marker.1.6.Cleanup” command

### Test Procedure – Graph Window Marker Appears

**Purpose –** Ensure creating a window marker actually creates the marker on the graph.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.1.7.Setup” Tcl command
* Pre-Condition – There is a blank graph on the Tk window
* Body

1. Run “graph.marker::RBC.graph.marker.1.7.Body” Tcl command

* Post-Condition – A button appears on the bottom of the graph
* Cleanup - Run “graph.marker::RBC.graph.marker.1.7.Cleanup” command

## Test Case 2 (Destroy)

### Test Procedure – Delete Single Marker Removes from Graph

**Purpose –** Ensure deleting a text marker actually removes the marker on the graph.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.2.3.Setup” Tcl command
* Pre-Condition – There is a text marker on the graph
* Body

1. Run “graph.marker::RBC.graph.marker.2.3.Body” Tcl command

* Post-Condition – The text marker is gone and a blank graph is showing
* Cleanup - Run “graph.marker::RBC.graph.marker.2.3.Cleanup” command

### Test Procedure – Delete Multiple Marker Removes from Graph

**Purpose –** Ensure deleting multiple markers actually remove them from the graph.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.2.4.Setup” Tcl command
* Pre-Condition – There is a text marker on the graph
* Body

1. Run “graph.marker::RBC.graph.marker.2.4.Body” Tcl command

* Post-Condition – The text marker is gone and a blank graph is showing
* Cleanup - Run “graph.marker::RBC.graph.marker.2.4.Cleanup” command

## Test Case 6 (Configure)

### Test Procedure – Displaying Configure Coords

**Purpose –** Ensure coords changes position on screen.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.6.9.Setup” Tcl command
* Pre-Condition – There is a text marker in the center of the screen
* Body

1. Run “graph.marker::RBC.graph.marker.6.9.Body” Tcl command

* Post-Condition – The text marker is lower than the it was previously

Cleanup - Run “graph.marker::RBC.graph.marker.6.9.Cleanup” command

### Test Procedure – Displaying Configure Element

**Purpose –** Ensure element links to the marker.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.6.10.Setup” Tcl command
* Pre-Condition – There is a text marker on the graph
* Body

1. Run “graph.marker::RBC.graph.marker.6.10.Body” Tcl command

* Post-Condition – The text marker is gone and a blank graph is showing

Cleanup - Run “graph.marker::RBC.graph.marker.6.10.Cleanup” command

### Test Procedure – Displaying Configure Hide

**Purpose –** Ensure hide removes the marker from view.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.6.11.Setup” Tcl command
* Pre-Condition – There is a text marker on the graph
* Body

1. Run “graph.marker::RBC.graph.marker.6.11.Body” Tcl command

* Post-Condition – The text marker is gone and a blank graph is showing

Cleanup - Run “graph.marker::RBC.graph.marker.6.11.Cleanup” command

### Test Procedure – Displaying Configure Under

**Purpose –** Ensure under draws the marker below data elements.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.6.12.Setup” Tcl command
* Pre-Condition – The text marker is drawn above data elements.
* Body

1. Run “graph.marker::RBC.graph.marker.6.12.Body” Tcl command

* Post-Condition – The data elements are now drawn above the marker

Cleanup - Run “graph.marker::RBC.graph.marker.6.12.Cleanup” command

### Test Procedure – Displaying Configure XOffset

**Purpose –** Ensure xoffset draws the marker at a horizontal offset

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.6.13.Setup” Tcl command
* Pre-Condition – There is a text marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.6.13.Body” Tcl command

* Post-Condition – The text marker is shifted horizontally to the right

Cleanup - Run “graph.marker::RBC.graph.marker.6.13.Cleanup” command

### Test Procedure – Displaying Configure YOffset

**Purpose –** Ensure yoffset draws the marker at a vertical offset.

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.6.14.Setup” Tcl command
* Pre-Condition – There is a text marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.6.14.Body” Tcl command

* Post-Condition – The text marker is shifted down vertically

Cleanup - Run “graph.marker::RBC.graph.marker.6.14.Cleanup” command

#### Test Case 7

### Test Procedure – Displaying Configure Anchor

**Purpose –** Ensure anchor changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.12.Setup” Tcl command
* Pre-Condition – There is a text marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.7.12.Body” Tcl command

* Post-Condition – The text marker is shifted down vertically to its anchor

Cleanup - Run “graph.marker::RBC.graph.marker.7.12.Cleanup” command

### Test Procedure – Displaying Configure Background

**Purpose –** Ensure background changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.13.Setup” Tcl command
* Pre-Condition – There is a text marker with a blank background
* Body

1. Run “graph.marker::RBC.graph.marker.7.13.Body” Tcl command

* Post-Condition – The text marker has a red background

Cleanup - Run “graph.marker::RBC.graph.marker.7.13.Cleanup” command

### Test Procedure – Displaying Configure Font

**Purpose –** Ensure font changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.14.Setup” Tcl command
* Pre-Condition – There is a text marker has the default font
* Body

1. Run “graph.marker::RBC.graph.marker.7.14.Body” Tcl command

* Post-Condition – The text marker’s font has changed to be huge

Cleanup - Run “graph.marker::RBC.graph.marker.7.14.Cleanup” command

### Test Procedure – Displaying Configure Fill

**Purpose –** Ensure fill changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.15.Setup” Tcl command
* Pre-Condition – There is a text marker with a blank background
* Body

1. Run “graph.marker::RBC.graph.marker.7.15.Body” Tcl command

* Post-Condition – The text marker has a red background

Cleanup - Run “graph.marker::RBC.graph.marker.7.15.Cleanup” command

### Test Procedure – Displaying Configure Foreground

**Purpose –** Ensure foreground changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.16.Setup” Tcl command
* Pre-Condition – There is a text marker with a blank foreground
* Body

1. Run “graph.marker::RBC.graph.marker.7.16.Body” Tcl command

* Post-Condition – The text marker has a red foreground

Cleanup - Run “graph.marker::RBC.graph.marker.7.16.Cleanup” command

### Test Procedure – Displaying Configure Justify

**Purpose –** Ensure justify changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.17.Setup” Tcl command
* Pre-Condition – There is a text marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.7.17.Body” Tcl command

* Post-Condition – The text marker’s text now is left justified

Cleanup - Run “graph.marker::RBC.graph.marker.7.17.Cleanup” command

### Test Procedure – Displaying Configure Outline

**Purpose –** Ensure outline changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.18.Setup” Tcl command
* Pre-Condition – There is a text marker with a blank foreground
* Body

1. Run “graph.marker::RBC.graph.marker.7.18.Body” Tcl command

* Post-Condition – The text marker has a red foreground

Cleanup - Run “graph.marker::RBC.graph.marker.7.18.Cleanup” command

### Test Procedure – Displaying Configure Padx

**Purpose –** Ensure padx changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.19.Setup” Tcl command
* Pre-Condition – There is a text marker with an even horizontal padding
* Body

1. Run “graph.marker::RBC.graph.marker.7.19.Body” Tcl command

* Post-Condition – The text marker’s horizontal padding is increased and uneven

Cleanup - Run “graph.marker::RBC.graph.marker.7.19.Cleanup” command

### Test Procedure – Displaying Configure Pady

**Purpose –** Ensure pady changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.20.Setup” Tcl command
* Pre-Condition – There is a text marker with an even vertical padding
* Body

1. Run “graph.marker::RBC.graph.marker.7.20.Body” Tcl command

* Post-Condition – The text marker’s vertical padding is increased and uneven

Cleanup - Run “graph.marker::RBC.graph.marker.7.20.Cleanup” command

### Test Procedure – Displaying Configure Rotate

**Purpose –** Ensure rotate changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.21.Setup” Tcl command
* Pre-Condition – There is a text marker says ‘Marker 1’
* Body

1. Run “graph.marker::RBC.graph.marker.7.21.Body” Tcl command

* Post-Condition – The text marker says ‘Text’

Cleanup - Run “graph.marker::RBC.graph.marker.7.21.Cleanup” command

### Test Procedure – Displaying Configure Text

**Purpose –** Ensure text changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.22.Setup” Tcl command
* Pre-Condition – There is a text marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.7.22.Body” Tcl command

* Post-Condition – The text marker is shifted down vertically

Cleanup - Run “graph.marker::RBC.graph.marker.7.22.Cleanup” command

### Test Procedure – Displaying Configure Shadow

**Purpose –** Ensure Shadow changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.7.23.Setup” Tcl command
* Pre-Condition – There is a text marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.7.23.Body” Tcl command

* Post-Condition – The text marker has a red shadow behind it

Cleanup - Run “graph.marker::RBC.graph.marker.7.23.Cleanup” command

#### Test Case 8

### Test Procedure – Displaying Configure Dashes

**Purpose –** Ensure dashes changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.8.6.Setup” Tcl command
* Pre-Condition – There is a line marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.8.6.Body” Tcl command

* Post-Condition – The line marker changes to a dashed pattern

Cleanup - Run “graph.marker::RBC.graph.marker.8.6.Cleanup” command

### Test Procedure – Displaying Configure Fill (No effect)

**Purpose –** Ensure fill changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.8.7.Setup” Tcl command
* Pre-Condition – There is a line marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.8.7.Body” Tcl command

* Post-Condition – No effect

Cleanup - Run “graph.marker::RBC.graph.marker.8.7.Cleanup” command

### Test Procedure – Displaying Configure Linewidth

**Purpose –** Ensure linewidth changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.8.8.Setup” Tcl command
* Pre-Condition – There is a line marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.8.8.Body” Tcl command

* Post-Condition – The line marker changes to be thicker

Cleanup - Run “graph.marker::RBC.graph.marker.8.8.Cleanup” command

### Test Procedure – Displaying Configure Outline

**Purpose –** Ensure outline changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.8.9.Setup” Tcl command
* Pre-Condition – There is a line marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.8.9.Body” Tcl command

* Post-Condition – The line changes color to red

Cleanup - Run “graph.marker::RBC.graph.marker.8.9.Cleanup” command

### Test Procedure – Displaying Configure Outline

**Purpose –** Ensure outline changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.8.9.Setup” Tcl command
* Pre-Condition – There is a line marker in the center of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.8.9.Body” Tcl command

* Post-Condition – The line changes color to red

Cleanup - Run “graph.marker::RBC.graph.marker.8.9.Cleanup” command

#### Test Case 9

### Test Procedure – Displaying Configure Background

**Purpose –** Ensure background changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.9.7.Setup” Tcl command
* Pre-Condition – A bitmap of a dollar bill is in the lower left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.9.7.Body” Tcl command

* Post-Condition – The back of the bitmap changes color to red

Cleanup - Run “graph.marker::RBC.graph.marker.9.7.Cleanup” command

### Test Procedure – Displaying Configure Bitmap

**Purpose –** Ensure bitmap changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.9.8.Setup” Tcl command
* Pre-Condition – A blank graph window appears
* Body

1. Run “graph.marker::RBC.graph.marker.9.8.Body” Tcl command

* Post-Condition – A bitmap of a dollar bill is in the lower left of a graph

Cleanup - Run “graph.marker::RBC.graph.marker.9.8.Cleanup” command

### Test Procedure – Displaying Configure Fill

**Purpose –** Ensure fill changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.9.9.Setup” Tcl command
* Pre-Condition – A bitmap of a dollar bill is in the lower left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.9.9.Body” Tcl command

* Post-Condition – The back of the bitmap changes color to red

Cleanup - Run “graph.marker::RBC.graph.marker.9.9.Cleanup” command

### Test Procedure – Displaying Configure Foreground

**Purpose –** Ensure foreground changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.9.10.Setup” Tcl command
* Pre-Condition – A bitmap of a dollar bill is in the lower left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.9.10.Body” Tcl command

* Post-Condition – The front writing of the bitmap changes color to red

Cleanup - Run “graph.marker::RBC.graph.marker.9.10.Cleanup” command

### Test Procedure – Displaying Configure Outline

**Purpose –** Ensure outline changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.9.11.Setup” Tcl command
* Pre-Condition – A bitmap of a dollar bill is in the lower left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.9.11.Body” Tcl command

* Post-Condition – The front writing of the bitmap changes color to red

Cleanup - Run “graph.marker::RBC.graph.marker.9.11.Cleanup” command

### Test Procedure – Displaying Configure Rotate

**Purpose –** Ensure rotate changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.9.12.Setup” Tcl command
* Pre-Condition – A bitmap of a dollar bill is in the lower left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.9.12.Body” Tcl command

* Post-Condition – The bitmap rotates so the right side is facing towards the top right

Cleanup - Run “graph.marker::RBC.graph.marker.9.12.Cleanup” command

### Test Procedure – Displaying Configure Anchor

**Purpose –** Ensure anchor changes on screen

**Special Requirements –** none

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.9.13.Setup” Tcl command
* Pre-Condition – A bitmap of a dollar bill is in the lower left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.9.13.Body” Tcl command

* Post-Condition – The bitmap lowers from its previous position

Cleanup - Run “graph.marker::RBC.graph.marker.9.13.Cleanup” command

#### Test Case 10

### Test Procedure – Displaying Configure Anchor

**Purpose –** Ensure anchor changes on screen

**Special Requirements –** buckskin.gif needs to be in executing directory

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.10.3.Setup” Tcl command
* Pre-Condition – An image of a texture is in the lower right of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.10.3.Body” Tcl command

* Post-Condition – The image lowers from its previous position

Cleanup - Run “graph.marker::RBC.graph.marker.10.3.Cleanup” command

### Test Procedure – Displaying Configure Image

**Purpose –** Ensure image changes on screen

**Special Requirements –** buckskin.gif and stopsign.gif need to be in the executing directory

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.10.4.Setup” Tcl command
* Pre-Condition – An image of a texture is in the lower right of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.10.4.Body” Tcl command

* Post-Condition – The image changes to one of a stop sign

Cleanup - Run “graph.marker::RBC.graph.marker.10.4.Cleanup” command

#### Test Case 11

### Test Procedure – Displaying Configure Dashes

**Purpose –** Ensure dashes changes on screen

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.11.6.Setup” Tcl command
* Pre-Condition – A triangle is drawn across the upper left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.11.6.Body” Tcl command

* Post-Condition – The triangle lines have a dashed patterns

Cleanup - Run “graph.marker::RBC.graph.marker.11.6.Cleanup” command

### Test Procedure – Displaying Configure Fill

**Purpose –** Ensure fill changes on screen

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.11.7.Setup” Tcl command
* Pre-Condition – A triangle is drawn across the upper left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.11.7.Body” Tcl command

* Post-Condition – The triangle is filled with blue

Cleanup - Run “graph.marker::RBC.graph.marker.11.7.Cleanup” command

### Test Procedure – Displaying Configure Linewidth

**Purpose –** Ensure linewdith changes on screen

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.11.8.Setup” Tcl command
* Pre-Condition – A triangle is drawn across the upper left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.11.8.Body” Tcl command

* Post-Condition – The triangle lines increase in width

Cleanup - Run “graph.marker::RBC.graph.marker.11.8.Cleanup” command

### Test Procedure – Displaying Configure Outline

**Purpose –** Ensure outline changes on screen

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.11.9.Setup” Tcl command
* Pre-Condition – A triangle is drawn across the upper left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.11.9.Body” Tcl command

* Post-Condition – The triangle lines are changed to a green color

Cleanup - Run “graph.marker::RBC.graph.marker.11.9.Cleanup” command

### Test Procedure – Displaying Configure Stipple

**Purpose –** Ensure stipple changes on screen

**Special Requirements –** greenback.xbm is in the executing directory

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.11.10.Setup” Tcl command
* Pre-Condition – A triangle is drawn across the upper left of a graph
* Body

1. Run “graph.marker::RBC.graph.marker.11.10.Body” Tcl command

* Post-Condition – The triangle has a fill of a dollar bill image.

Cleanup - Run “graph.marker::RBC.graph.marker.11.10.Cleanup” command

#### Test Case 12

### Test Procedure – Displaying Configure Anchor

**Purpose –** Ensure anchor changes on screen

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.12.5.Setup” Tcl command
* Pre-Condition – A button with Button text is in the lower right of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.12.5.Body” Tcl command

* Post-Condition – The button is lowered from its previous position

Cleanup - Run “graph.marker::RBC.graph.marker.12.5.Cleanup” command

### Test Procedure – Displaying Configure Height

**Purpose –** Ensure height changes on screen

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.12.6.Setup” Tcl command
* Pre-Condition – A button with Button text is in the lower right of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.12.6.Body” Tcl command

* Post-Condition – The button shrinks in height

Cleanup - Run “graph.marker::RBC.graph.marker.12.6.Cleanup” command

### Test Procedure – Displaying Configure Width

**Purpose –** Ensure width changes on screen

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.12.7.Setup” Tcl command
* Pre-Condition – A button with Button text is in the lower right of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.12.7.Body” Tcl command

* Post-Condition – The button shrinks in width

Cleanup - Run “graph.marker::RBC.graph.marker.12.7.Cleanup” command

### Test Procedure – Displaying Configure Window

**Purpose –** Ensure window changes on screen

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker::RBC.graph.marker.12.8.Setup” Tcl command
* Pre-Condition – A button with Button text is in the lower right of the graph
* Body

1. Run “graph.marker::RBC.graph.marker.12.8.Body” Tcl command

* Post-Condition – The button changes to a new button with text “test”.

Cleanup - Run “graph.marker::RBC.graph.marker.12.8.Cleanup” command

## Bind

#### Test Case 13

### Test Procedure – Bind actions execute

**Purpose –** Ensure bind actions execute

**Special Requirements –** None

**Procedural Steps**

* Setup –Run the “graph.marker:: RBC.graph.marker.13.6.Setup” Tcl command
* Pre-Condition – A Text marker appears in the middle of a plot
* Body

1. Run “graph.marker::RBC.graph.marker.13.6.Body” Tcl command
2. Double click the text marker

* Post-Condition – The double click should output onto console the word “test”

Cleanup - Run “graph.marker::RBC.graph.marker.13.6.Cleanup” command

# Pen

## 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

# Postscript

## Output

#### Test Case 4

### Test Procedure – Output to Console

**Purpose –** Ensure that correct postscript output is returned.

**Special Requirements –** Current working directory for the wish interpreter must be known.

**Procedural Steps**

* Setup – Run the “RBC.graph.postscript.4.tcl” file and then call the “graph.postscript::RBC.graph.postscript.4.1.Setup” Tcl command
* Pre-Condition – An empty graph exists with only the axes showing. There is a file “RBC.graph.postcript.4.1.ps” in the current working directory with a size of 0Kb.
* Body

1. Call the “graph.postscript::RBC.graph.postscript.4.1.Body” Tcl command

* Post-Condition – The file “RBC.graph.postscript.4.1.ps” has content in it that matches the content in the “postScriptOutputTest.ps” file
* Cleanup – Call the “graph.postscript::RBC.graph.postscript.4.1.Cleanup” command

### Test Procedure – Output to File

**Purpose –** Ensure that correct postscript output is written to a file.

**Special Requirements –** Current working directory for the wish interpreter must be known.

**Procedural Steps**

* Setup – Run the “RBC.graph.postscript.4.tcl” file and then call the “graph.postscript::RBC.graph.postscript.4.2.Setup” Tcl command
* Pre-Condition – An empty graph exists with only the axes showing.
* Body

1. Call the “graph.postscript::RBC.graph.postscript.4.2.Body” Tcl command

* Post-Condition – The file “RBC.graph.postscript.4.2.ps” has content in it that matches the content in the “postScriptOutputTest.ps” file
* Cleanup – Call the “graph.postscript::RBC.graph.postscript.4.2.Cleanup” command

# Snap

#### Test Case 1

### Test Procedure – Full Sized

**Purpose –** Ensure snap takes a picture and stores it as a Tk image.

**Special Requirements –** None

**Script –** RBC.graph.snap.1.tcl

**Procedural Steps**

* Setup – Call the “graph.snap::RBC.graph.snap.1.1.Setup” Tcl command
* Pre-Condition – A graph is displayed with line data on it
* Body

1. Call the “graph.snap::RBC.graph.snap.1.1.Body” Tcl command

* Post-Condition – A button with a red background and image of the pre-condition graph on it.
* Cleanup – Call the “graph.snap::RBC.graph.snap.1.1.Cleanup” command

### Test Procedure – Vertically Smaller (Doesn’t Work)

**Purpose –** Ensure snap takes a set height picture and stores it as a Tk image.

**Special Requirements –** None

**Script –** RBC.graph.snap.1.tcl

**Procedural Steps**

* Setup – Call the “graph.snap::RBC.graph.snap.1.2.Setup” Tcl command
* Pre-Condition – A graph is displayed with line data on it
* Body

1. Call the “graph.snap::RBC.graph.snap.1.2.Body” Tcl command

* Post-Condition – A button with a red background and image of the vertically smaller pre-condition graph on it.
* Cleanup – Call the “graph.snap::RBC.graph.snap.1.2.Cleanup” command

### Test Procedure – Horizontally Smaller (Doesn’t Work)

**Purpose –** Ensure snap takes a set width picture and stores it as a Tk image

**Special Requirements –** None

**Script –** RBC.graph.snap.1.tcl

**Procedural Steps**

* Setup – Call the “graph.snap::RBC.graph.snap.1.3.Setup” Tcl command
* Pre-Condition – A graph is displayed with line data on it
* Body

1. Call the “graph.snap::RBC.graph.snap.1.3.Body” Tcl command

* Post-Condition – A button with a red background and image of the horizontally smaller pre-condition graph on it.
* Cleanup – Call the “graph.snap::RBC.graph.snap.1.3.Cleanup” command

# Transform

#### Test Case 1

### Test Procedure – Transform

**Purpose –** Ensure the transform command works correctly.

**Special Requirements –** None

**Script –** RBC.graph.transform.1.tcl

**Procedural Steps**

* Setup – Call the “graph.transform::RBC.graph.transform.1.1.Setup” Tcl command
* Pre-Condition – A graph is displayed
* Body

1. Call the “graph.transform::RBC.graph.transform.1.1.Body” Tcl command

* Post-Condition – Command returns 5060 -1288
* Cleanup – Call the “graph.transform::RBC.graph.transform.1.1.Cleanup” command