**Graph Crosshairs**

**Test Cases and Procedures**

**Authors:**

Sam Green

Nick Hudson

Stanton Sievers

Jarrod Stormo

# Graph Crosshairs Cget

### Test Case 1

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

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

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

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

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

**Inter-case Dependencies –**

# Graph Crosshairs Configure

### Test Case 1

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

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

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

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

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

**Inter-case Dependencies –** RBC.graph.crosshairs.cget.1

### Test Case 2

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

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

**Input Specification –** A list of up to 11 numbers that alternately represent the lengths of the dashes and gaps on the cross hair lines.

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

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

**Inter-case Dependencies –** RBC.graph.crosshairs.cget.1

### Test Case 3

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

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

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

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

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

**Inter-case Dependencies –** RBC.graph.crosshairs.cget.1

### Test Case 4

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

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

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

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

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

**Inter-case Dependencies –** RBC.graph.crosshairs.cget.1

### Test Case 5

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

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

**Input Specification –** Window coordinates in the form “@x,y”.

**Output Specification –** The coordinates of the crosshair intersection is set to the given coordinates.

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

**Inter-case Dependencies –** RBC.graph.crosshairs.cget.1

# Graph Crosshairs Off

### Test Case 1

**Test Case ID –** RBC.graph.crosshairs.off.1

**Test Item –** The *crosshairs off* command of the *graph* BLT component.

**Input Specification –** None

**Output Specification –** The crosshairs on the graph component should be hidden.

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

**Inter-case Dependencies –** None

# Graph Crosshairs On

### Test Case 1

**Test Case ID –** RBC.graph.crosshairs.on.1

**Test Item –** The *crosshairs on* command of the *graph* BLT component.

**Input Specification –** None

**Output Specification –** The crosshairs on the graph component should be visible.

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

**Inter-case Dependencies –** None

# Graph Crosshairs Toggle

### Test Case 1

**Test Case ID –** RBC.graph.crosshairs.toggle.1

**Test Item –** The *crosshairs toggle* command of the *graph* BLT component.

**Input Specification –** None

**Output Specification –** The visibility of the crosshairs on the graph component should be the opposite of what it was to start.

**Special Procedural Requirements –** None

**Inter-case Dependencies –** RBC.graph.crosshairs.off.1, RBC.graph.crosshairs.on.1

# Automated Tests

**Cget**

**Test Procedure – Graph Crosshairs Cget – Valid Option Name**

**Test Case 1**

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

**Special Requirements –** None

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

**Test Procedure – Graph Crosshairs Cget – Invalid Option Name**

**Test Case 1**

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

**Special Requirements –** None

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

**Configure**

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

**Test Case 1**

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

**Special Requirements –** None

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

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

**Test Case 1**

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

**Special Requirements –** None

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

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

**Test Case 2**

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

**Special Requirements –** None

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

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

**Test Case 2**

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

**Special Requirements –** None

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

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

**Test Case 2**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs.configure.2.3

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

**Test Case 2**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs.configure.2.4

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

**Test Case 2**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs.configure.2.5

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

**Test Case 3**

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

**Special Requirements –** None

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

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

**Test Case 3**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs..4.2

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

**Test Case 3**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs.configure.3.3

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

**Test Case 3**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs.configure.3.4

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

**Test Case 3**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs.configure.3.5

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

**Test Case 3**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs.configure.3.6

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

**Test Case 3**

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

**Special Requirements –** None

**TclTest –** RBC.graph.crosshairs.configure.3.7

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

**Test Case 4**

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

**Special Requirements –** None

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

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

**Test Case 4**

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

**Special Requirements –** None

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

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

**Test Case 4**

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

**Special Requirements –** None

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

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

**Test Case 4**

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

**Special Requirements –** None

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

**Test Procedure – Graph Crosshairs Configure: Position – Valid Coordinates**

**Test Case 5**

**Purpose –** Ensure the *crosshairs configure -position* command works correctly when given valid coordinates as input.

**Special Requirements –** None

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

**Test Procedure – Graph Crosshairs Configure: Position – Invalid Coordinates**

**Test Case 5**

**Purpose –** Ensure the *crosshairs configure -position* command works correctly when given invalid coordinates as input.

**Special Requirements –** None

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

**Test Procedure – Graph Crosshairs Configure: Position – Not Coordinate Form**

**Test Case 5**

**Purpose –** Ensure the *crosshairs configure -position* command works correctly when given input that is not in coordinate form (@x,y).

**Special Requirements –** None

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

# Manual Tests

**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