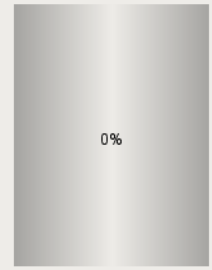
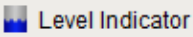
Level Indicator

General



Component Palette Icon:



Description

A component that can be filled up with water. Usually used behind a symbol factor object that has a cutout in it.

Properties

Name	Description	Property Type	Scripting	Category
Anti Alias	Draw with antialias on? Makes text smoother.	boolean	.antiAlias	Appearance
Background Color	The color of the background.	Color	.background	Appearance
Border	The border surrounding this component. NOTE that the border is unaffected by rotation.	Border	.border	Common
Capacity	Total capacity of tank.	double	.capacity	Data
Cursor	The mouse cursor to use when hovering over this component.	int	.cursorCode	Common
Data Quality	The data quality code for any tag bindings on this component.	int	.dataQuality	Data
Filled Color	Set the color of filled portion.	Color	.foreground	Appearance
Font	Font of text on this component.	Font	.font	Appearance
Font Color	The foreground color of the component.	Color	.fontColor	Appearance
Gradient	Draw level as 3D gradient?.	boolean	.gradient	Appearance
Liquid Waves	Draw liquid 'waves'?	boolean	.waves	Appearance
Mouseover Text	The text that is displayed in the tooltip which pops up on mouseover of this component.	String	.toolTipText	Common
Name	The name of this component.	String	.name	Common
Orientation	Determines which way the level "grows" for an increase in value.	int	.orientation	Appearance
Percent Format	Format string used for the percentage.	String	.percentFormat	Appearance
Show Percentage	Show percentage of tank filled?	boolean	.showPercent	Appearance
Show Value	Show numeric value, capacity, and units?	boolean	.showValue	Appearance
Styles	Contains the component's styles.	Dataset	.styles	Appearance
Units	Units of measure for tank contents.	String	.units	Appearance
Value	Numeric value of tank's level.	double	.value	Data
Value Format	Format string used for the value.	String	.valueFormat	Appearance
Visible	If disabled, the component will be hidden.	boolean	.visible	Common
Wave Height	The height of each 'wave'.	int	.waveHeight	Appearance
Wave Length	The length of each 'wave'.	int	.waveLength	Appearance

Scripting

Scripting Functions

This component does not have scripting functions associated with it.

Extension Functions

This component does not have extension functions associated with it.

Event Handlers

mouse

mouseClicked

This event signifies a mouse click on the source component. A mouse click the combination of a mouse press and a mouse release, both of which must have occurred over the source component. Note that this event fires after the pressed and released events have fired.

.source	The component that fired this event
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

mouseEntered

This event fires when the mouse enters the space over the source component.

.source	The component that fired this event
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

mouseExited

This event fires when the mouse leaves the space over the source component.

.source	The component that fired this event
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

mousePressed

This event fires when a mouse button is pressed down on the source component.

.source	The component that fired this event
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

mouseReleased

This event fires when a mouse button is released, if that mouse button's press happened over this component.

.source	The component that fired this event
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

mouseMotion

mouseDragged

Fires when the mouse moves over a component after a button has been pushed.

.source	The component that fired this event
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

mouseMoved

Fires when the mouse moves over a component, but no buttons are pushed.

.source	The component that fired this event
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

propertyChange

propertyChange

Fires whenever a bindable property of the source component changes. This works for standard and custom (dynamic) properties.

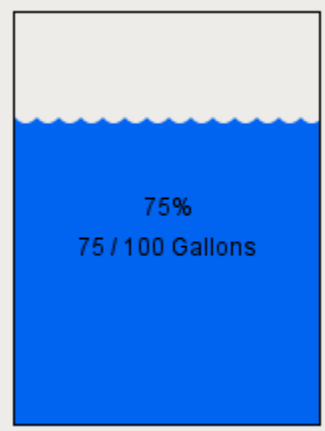
.source	The component that fired this event
.newValue	The new value that this property changed to.
.oldValue	The value that this property was before it changed. Note that not all components include an accurate oldValue in their events.
.propertyName	The name of the property that changed. NOTE: remember to always filter out these events for the property that you are looking for! Components often have many properties that change.

Customizers

- Component Customizers
- Style Customizer

Examples

Level Indicator



Property Name	Value
Border	Line Border
Value	75
Units	Gallons
Show Value	True
Gradient	False
Filled Color	0,100,240