



# GTK::V3::Gtk::GtkScale

**GtkScale — A slider widget for selecting a value from a range**

## Table of Contents

1	<a href="#">Synopsis</a>
2	<a href="#">Methods</a>
2.1	<a href="#">new</a>
2.2	<a href="#">[gtk_scale_] new_with_range</a>
2.3	<a href="#">[gtk_scale_] set_digits</a>
2.4	<a href="#">[gtk_scale_] set_draw_value</a>
2.5	<a href="#">[gtk_scale_] set_has_origin</a>
2.6	<a href="#">[gtk_scale_] set_value_pos</a>
2.7	<a href="#">[gtk_scale_] get_digits</a>
2.8	<a href="#">[gtk_scale_] get_draw_value</a>
2.9	<a href="#">[gtk_scale_] get_has_origin</a>
2.10	<a href="#">[gtk_scale_] get_value_pos</a>
2.11	<a href="#">[gtk_scale_] add_mark</a>
2.12	<a href="#">[gtk_scale_] clear_marks</a>
3	<a href="#">Signals</a>
3.1	<a href="#">Not yet supported signals</a>
3.1.1	<a href="#">format-value</a>

```
unit class GTK::V3::Gtk::GtkScale;  
also is GTK::V3::Gtk::GtkRange;
```

# Synopsis

```
my Gtk::V3::Gtk::GtkScale $scale .= new(:empty);

# Set min and max of scale.
$scale.set-range( -2e0, .2e2);

# Step (keys left/right) and page (mouse scroll on scale).
$scale.set-increments( .2e0, 5e0);

# Value of current position displayed below the scale
$scale.set-value-pos(GTK_POS_BOTTOM);

# Want to have two digits after the comma
$scale.set-digits(2);

# Want to have some scale ticks also below the scale
$scale.add-mark( 0e0, GTK_POS_BOTTOM, 'Zero');
$scale.add-mark( 5e0, GTK_POS_BOTTOM, 'Five');
$scale.add-mark( 10e0, GTK_POS_BOTTOM, 'Ten');
$scale.add-mark( 15e0, GTK_POS_BOTTOM, 'Fifteen');
$scale.add-mark( 20e0, GTK_POS_BOTTOM, 'Twenty');
```

Result will be like [this scale detail](#).

## Methods

### new

```
multi method new ( Int :$orientation!, Num $min!, Num $max!, Num $step! )
```

Creates a new GtkScale providing an orientation and minimum, maximum and step size.

- `$orientation`; the scale's orientation. Value is a GtkOrientation enum from GtkEnums.
- `$min`; minimum value
- `$max`; maximum value
- `$step`; step increment (tick size) used with keyboard shortcuts

```
multi method new ( :$widget! )
```

Create an object using a native object from elsewhere. See also `Gtk::V3::Glib::GObject`.

```
multi method new ( Str :$build-id! )
```

Create an object using a native object from a builder. See also `Gtk::V3::Glib::GObject`.

## **[gtk\_scale\_] new\_with\_range**

Creates a new native scale object

```
method gtk_scale_new_with_range (
  Int $orientation, Num $min, Num $max, Num $step
  --> N-GObject
)
```

Returns a native widget. It is not advised to use it. The `new()/BUILD()` method can handle this better and easier.

- `$orientation`; the scale's orientation. Value is a `GtkOrientation` enum from `GtkEnums`.
- `$adjustment`; a value of type `GtkAdjustment` which sets the range of the scale, or `NULL` to create a new adjustment.

## **[gtk\_scale\_] set\_digits**

Sets the number of decimal places that are displayed in the value. Also causes the value of the adjustment to be rounded to this number of digits, so the retrieved value matches the displayed one, if “draw-value” is 1 when the value changes. If you want to enforce rounding the value when “draw-value” is 0, you can set “round-digits” instead.

Note that rounding to a small number of digits can interfere with the smooth autoscrolling that is built into `GtkScale`. As an alternative, you can use the “format-value” signal to format the displayed value yourself.

```
method gtk_scale_set_digits ( Int $digits )
```

- `$digits`; the number of decimal places to display, e.g. use 1 to display 1.0, 2 to display 1.00, etc

## **[gtk\_scale\_] set\_draw\_value**

Specifies whether the current value is displayed as a string next to the slider.

```
method gtk_scale_set_draw_value ( Int $draw-value )
```

- \$draw-value; set to 1 to draw the value.

## **[gtk\_scale\_] set\_has\_origin**

If “has-origin” is set to 1 (the default), the scale will highlight the part of the trough between the origin (bottom or left side) and the current value.

```
method gtk_scale_set_has_origin ( Int $has-origin )
```

- \$has-origin; 1 if the scale has an origin.

## **[gtk\_scale\_] set\_value\_pos**

Sets the position in which the current value is displayed.

```
method gtk_scale_set_value_pos ( Int $pos )
```

- \$pos; the position in which the current value is displayed. This value is of type GtkPositionType found in [GTK::V3::Gtk::GtkEnums](#).

## **[gtk\_scale\_] get\_digits**

Gets the number of decimal places that are displayed in the value.

```
method gtk_scale_get_digits ( )
```

Returns the number of digits.

## **[gtk\_scale\_] get\_draw\_value**

Returns whether the current value is displayed as a string next to the slider.

```
method gtk_scale_get_draw_value ( --> Int )
```

Returns 1 when the current value is displayed as a string next to the slider.

## **[gtk\_scale\_] get\_has\_origin**

Returns whether the scale has an origin.

```
method gtk_scale_get_has_origin ( --> Int )
```

Returns 1 if the scale has an origin.

## [gtk\_scale\_] get\_value\_pos

Gets the position in which the current value is displayed.

```
method gtk_scale_get_value_pos ( --> Int )
```

Returns the position in which the current value is displayed. The value is an enum type `GtkPositionType` defined in `GTK::V3::Gtk::GtkEnums`.

## [gtk\_scale\_] add\_mark

```
method gtk_scale_add_mark ( Num $value, Int $pos, Str $markup )
```

- `$value`; the value at which the mark is placed, must be between the lower and upper limits of the scales' adjustment
- `$position`; where to draw the mark. For a horizontal scale, `GTK_POS_TOP` and `GTK_POS_LEFT` are drawn above the scale, anything else below. For a vertical scale, `GTK_POS_LEFT` and `GTK_POS_TOP` are drawn to the left of the scale, anything else to the right. This is an enum type of `GtkPositionType` defined in `GTK::V3::Gtk::GtkEnums`.
- `$markup`; Text to be shown at the mark, using Pango markup, or `NULL`.

## [gtk\_scale\_] clear\_marks

Removes any marks that have been added with `gtk_scale_add_mark()`.

```
method gtk_scale_clear_marks ( )
```

## Signals

### Not yet supported signals

#### format-value

Signal which allows you to change how the scale value is displayed. Connect a signal handler which returns an allocated string representing value. That string will then be used to display the scale's value.

If no user-provided handlers are installed, the value will be displayed on its own, rounded according to the value of the “digits” property.