



Gnome::Gtk3::ColorButton

A button to launch a color selection dialog

Table of Contents

1	Description
1.1	Css Nodes
1.2	See Also
2	Synopsis
2.1	Declaration
2.2	Example
3	Methods
3.1	new
3.2	gtk_color_button_new
3.3	[gtk_color_button_] new_with_rgba
3.4	[gtk_color_button_] set_title
3.5	[gtk_color_button_] get_title
4	Properties
4.1	use-alpha
4.2	title
4.3	alpha
4.4	rgba
4.5	show-editor
5	Signals
5.1	Not yet supported signals
5.1.1	Gnome::Gtk3::ColorButton ::color-set:

Description

The `Gnome::Gtk3::ColorButton` is a button which displays the currently selected color and allows to open a color selection dialog to change the color. It is suitable widget for selecting a color in a preference dialog.

Css Nodes

`Gnome::Gtk3::ColorButton` has a single CSS node with name `button`. To differentiate it from a plain `Gnome::Gtk3::Button`, it gets the `.color` style class.

See Also

[Gnome::Gtk3::ColorSelectionDialog](#), [Gnome::Gtk3::FontButton](#)

Synopsis

Declaration

```
unit class Gnome::Gtk3::ColorButton;  
also is Gnome::LIBRARYPARENT;
```

Example

Methods

new

...

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

Create an object using a native object from elsewhere. See also [Gnome::GObject::Object](#).

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

Create an object using a native object from a builder. See also [Gnome::GObject::Object](#).

gtk_color_button_new

Creates a new color button.

This returns a widget in the form of a small button containing a swatch representing the current selected color. When the button is clicked, a color-selection dialog will open, allowing the user to select a color. The swatch will be updated to reflect the new color when the user finishes.

```
method gtk_color_button_new ( --> N-GObject )
```

Returns N-GObject; a new color button

[gtk_color_button_] new_with_rgba

Creates a new color button.

```
method gtk_color_button_new_with_rgba ( N-GObject $rgba --> N-GObject )
```

- N-GObject \$rgba; A `Gnome::Gdk3::RGBA` to set the current color with

Returns N-GObject; a new color button

[gtk_color_button_] set_title

Sets the title for the color selection dialog.

```
method gtk_color_button_set_title ( Str $title)
```

- Str \$title; String containing new window title

[gtk_color_button_] get_title

Gets the title of the color selection dialog.

```
method gtk_color_button_get_title ( --> Str )
```

Returns str; An internal string, do not free the return value

Properties

An example of using a string type property of a `Gnome::Gtk3::Label` object. This is just showing how to set/read a property, not that it is the best way to do it. This is because a) The class initialization often provides some options to set some of the properties and b) the classes provide many methods to modify just those properties.

```
my Gnome::Gtk3::Label $label .= new(:empty);  
my Gnome::GObject::Value $gv .= new(:init(G_TYPE_STRING));  
$label.g-object-get-property( 'label', $gv);  
$gv.g-value-set-string('my text label');
```

use-alpha

If this property is set to 1, the color swatch on the button is rendered against a checkerboard background to show its opacity and the opacity slider is displayed in the color selection dialog.

title

The title of the color selection dialog

alpha

The selected opacity value (0 fully transparent, 65535 fully opaque).

rgba

The RGBA color.

show-editor

Set this property to 1 to skip the palette in the dialog and go directly to the color editor.

This property should be used in cases where the palette in the editor would be redundant, such as when the color button is already part of a palette.

Signals

Register any signal as follows. See also [Gnome::GObject::Object](#).

```
my Bool $is-registered = $my-widget.register-signal (  
  $handler-object, $handler-name, $signal-name,  
  :$user-option1, ..., $user-optionN  
)
```

Not yet supported signals

[Gnome::Gtk3::ColorButton::color-set:](#)

The `::color-set` signal is emitted when the user selects a color. When handling this signal, use `gtk_color_button_get_rgba()` to find out which color was just selected.

Note that this signal is only emitted when the user changes the color. If you need to react to programmatic color changes as well, use the `notify::color` signal.

```
method handler (  
  Gnome::GObject::Object :$widget,  
  :$user-option1, ..., $user-optionN  
) ;
```

- `$widget`; the object which received the signal.