



# class GTK::V3::Gtk::GtkWindow

## Table of Contents

- 0.1 [GtkWindow — Toplevel which can contain other widgets](#)
- 1 [Enums](#)
  - 1.1 [GtkWindowPosition](#)
  - 1.2 [GtkWindowType](#)
- 2 [Methods](#)
  - 2.1 [gtk\\_window\\_new](#)
  - 2.2 [\[gtk\\_window\\_\] set\\_title](#)
  - 2.3 [\[gtk\\_window\\_\] set\\_default\\_size](#)
  - 2.4 [\[gtk\\_window\\_\] set\\_modal](#)
  - 2.5 [\[gtk\\_window\\_\] set\\_position](#)
  - 2.6 [\[gtk\\_window\\_\] set\\_transient\\_for](#)
  - 2.7 [new](#)

```
unit class GTK::V3::Gtk::GtkWindow;  
also is GTK::V3::Gtk::GtkBin;
```

## GtkWindow — Toplevel which can contain other widgets

See readme for an example.

## Enums

### GtkWindowPosition

Window placement can be influenced using this enumeration. Note that using `GTK_WIN_POS_CENTER_ALWAYS` is almost always a bad idea. It won't necessarily work well with all window managers or on all windowing systems.

- `GTK_WIN_POS_NONE`. No influence is made on placement.
- `GTK_WIN_POS_CENTER`. Windows should be placed in the center of the screen.
- `GTK_WIN_POS_MOUSE`. Windows should be placed at the current mouse position.
- `GTK_WIN_POS_CENTER_ALWAYS`. Keep window centered as it changes

size, etc.

- `GTK_WIN_POS_CENTER_PARENT`. Center the window on its transient parent.

## GtkWindowType

A `GtkWindow` can be one of these types. Most things you'd consider a "window" should have type `GTK_WINDOW_TOPLEVEL`; windows with this type are managed by the window manager and have a frame by default (call `gtk_window_set_decorated()` to toggle the frame). Windows with type `GTK_WINDOW_POPUP` are ignored by the window manager; window manager keybindings won't work on them, the window manager won't decorate the window with a frame, many GTK+ features that rely on the window manager will not work (e.g. resize grips and maximization/minimization).

`GTK_WINDOW_POPUP` is used to implement widgets such as `GtkMenu` or tooltips that you normally don't think of as windows per se. Nearly all windows should be `GTK_WINDOW_TOPLEVEL`. In particular, do not use `GTK_WINDOW_POPUP` just to turn off the window borders; use `gtk_window_set_decorated()` for that.

- `GTK_WINDOW_TOPLEVEL`. A regular window, such as a dialog.
- `GTK_WINDOW_POPUP`. A special window such as a tooltip.

## Methods

### `gtk_window_new`

```
method gtk_window_new ( GtkWindowType $type )
```

Creates a new `GtkWindow`, which is a toplevel window that can contain other widgets. Nearly always, the type of the window should be `GTK_WINDOW_TOPLEVEL`. If you're implementing something like a popup menu from scratch (which is a bad idea, just use `GtkMenu`), you might use `GTK_WINDOW_POPUP`. `GTK_WINDOW_POPUP` is not for dialogs, though in some other toolkits dialogs are called "popups". In GTK+, `GTK_WINDOW_POPUP` means a pop-up menu or pop-up tooltip. On X11, popup windows are not controlled by the window manager.

If you simply want an undecorated window (no window borders), use `gtk_window_set_decorated()`, don't use `GTK_WINDOW_POPUP`.

All top-level windows created by `gtk_window_new()` are stored in an internal top-level window list. This list can be obtained from `gtk_window_list_toplevels()`. Due to Gtk+ keeping a reference to the window internally, `gtk_window_new()` does not return a reference to the caller.

To delete a GtkWindow, call `gtk_widget_destroy()`.

## **[gtk\_window\_] set\_title**

```
method gtk_window_set_title ( Str $title )
```

Sets the title of the GtkWindow. The title of a window will be displayed in its title bar; on the X Window System, the title bar is rendered by the window manager, so exactly how the title appears to users may vary according to a user's exact configuration. The title should help a user distinguish this window from other windows they may have open. A good title might include the application name and current document filename, for example.

## **[gtk\_window\_] set\_default\_size**

```
method gtk_window_set_default_size ( Int $width, Int $height )
```

Sets the default size of a window. See also [the developer docs](#).

## **[gtk\_window\_] set\_modal**

```
method gtk_window_set_modal ( Int $modal )
```

Sets a window modal or non-modal. Modal windows prevent interaction with other windows in the same application. To keep modal dialogs on top of main application windows, use `gtk_window_set_transient_for()` to make the dialog transient for the parent; most window managers will then disallow lowering the dialog below the parent.

## **[gtk\_window\_] set\_position**

```
method gtk_window_set_position ( Int $position )
```

Sets a position constraint for this window. If the old or new constraint is `GTK_WIN_POS_CENTER_ALWAYS`, this will also cause the window to be repositioned to satisfy the new constraint.

## **[gtk\_window\_] set\_transient\_for**

```
method gtk_window_set_transient_for ( GTK::V3::Glib::GObject $main-window )
```

Dialog windows should be set transient for the main application window they were spawned from. This allows window managers to e.g. keep the dialog on top of the main window, or center the dialog over the main window.

## new

```
multi submethod BUILD (
  Bool :$empty!, GtkWindowType :$window-type = GTK_WINDOW_TOPLEVEL
)
```

Create an empty top level window or popup.

```
multi submethod BUILD ( :$widget! )
```

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

```
multi submethod BUILD ( Str :$build-id! )
```

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

Generated using `Pod::Render`, `Pod::To::HTML`, `Camelia™` (butterfly) is © 2009 by Larry Wall