# class GTK::V3::Glib::GSignal

## Table of Contents



```
unit class GTK::V3::Glib::GSignal;
```

## GSignal — A means for customization of object behaviour and a general purpose notification mechanism

# Synopsis

```
# define method
method mouse-event ( :widget($w), :event($e)) { ... }

# get the window object
my GTK::V3::Gtk::GtkWindow $w .= new( ... );

# define proper handler. you must study the GTK develper guides. you will
# then notice that C<connect-object> is a bit different than the real mcCoy.
my Callable $handler;
$handler = -> N-GObject $ignore-w, GdkEvent $e, OpaquePointer $ignore-d {
  self.mouse-event( :widget($w), :event($e) );
}

# connect signal to the handler
$w.connect-object( 'button-press-event', $handler);
```

It will be easier to use the register-signal method

```
# define method
method mouse-event ( :widget($w), :event($e)) { ... }

# get the window object
my GTK::V3::Gtk::GtkWindow $w .= new( ... );

# then register
$w.register-signal( self, 'mouse-event', 'button-press-event', :time(now));
```

# Enumerations

## GConnectFlags

- G_CONNECT_AFTER; whether the handler should be called before or after the default handler of the signal.
- G_CONNECT_SWAPPED; whether the instance and data should be swapped when calling the handler; see g_signal_connect_swapped() for an example.

# Methods

## g_signal_connect_object

```
method g_signal_connect_object(
  Str $signal, Callable $handler, int32 $connect_flags = 0
  --> uint64
)
```

This is similar to g_signal_connect_data(), but uses a closure which ensures that the gobject stays alive during the call to c_handler by temporarily adding a reference count to gobject .

When the gobject is destroyed the signal handler will be automatically disconnected. Note that this is not currently threadsafe (ie: emitting a signal while gobject is being destroyed in another thread is not safe).

- $signal; a string of the form signal-name::detail.
- $handler; the callback to connect.
- $connect_flags; a combination of GConnectFlags.

## g_signal_connect

```
sub g_signal_connect ( Str $signal, Callable $handler --> uint64 )
```

Connects a callback function to a signal for a particular object.

- $signal; a string of the form "signal-name::detail".
- $handler; callback function to connect.