

# The Gtk.HeaderBar

...In combination with using Gtk.Builder and its XML data to build parts of a GUI, while other parts are built conventionally.

...evolving into something completely irrelevant, but more exciting.

Hamilton Python Users Group

Ian Stewart

2021-05-10

# Gtk.Header ~ Gtk Window

conventional\_window.py ✕

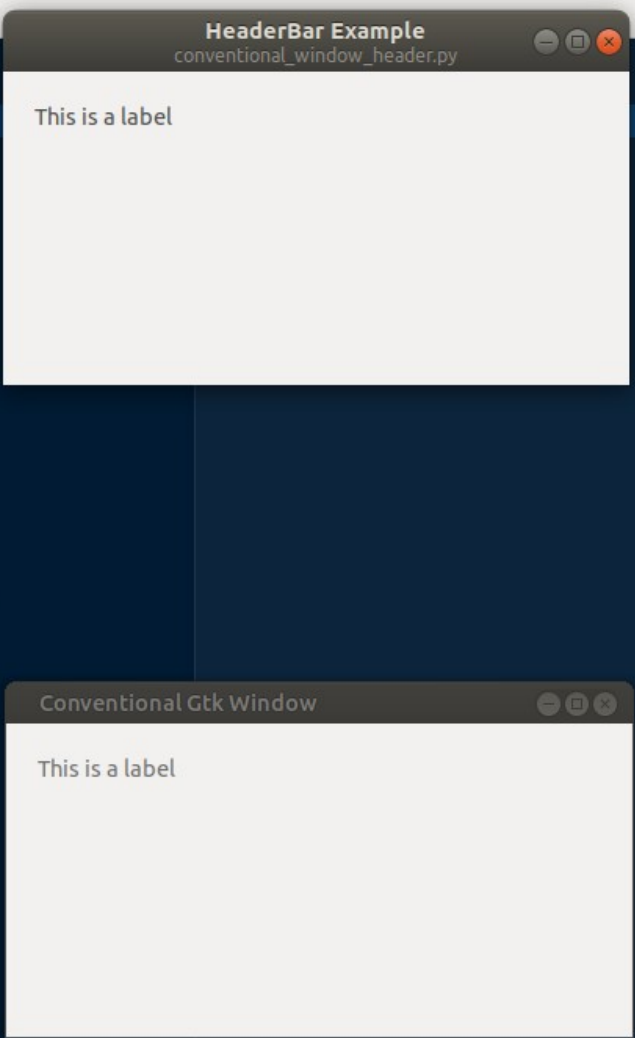
```
1#!/usr/bin/env python3
2#!
3# Conventional Gtk.Window
4import gi
5gi.require_version("Gtk", "3.0")
6from gi.repository import Gtk
7
8
9class ExampleWindow(Gtk.Window):
10    def __init__(self):
11        Gtk.Window.__init__(self, title="Conventional Gtk Window")
12        self.set_default_size(400, 200)
13
14        # Create widgets and add to the Gtk.Window
15        self.grid = Gtk.Grid()
16        self.grid.set_border_width(20)
17        self.label = Gtk.Label(label="This is a label")
18        self.grid.attach(self.label, 0,0,1,1)
19        self.add(self.grid)
20
21win = ExampleWindow()
22win.connect("destroy", Gtk.main_quit)
23win.show_all()
24Gtk.main()
```

Conventional Gtk Window

This is a label

# Gtk.Header ~ Gtk Window

```
conventional_window_header.py ✕
1 #!/usr/bin/env python3
2 #!
3 # Conventional Gtk.Window with header bar
4 import gi
5 gi.require_version("Gtk", "3.0")
6 from gi.repository import Gtk
7
8
9 class ExampleWindow(Gtk.Window):
10     def __init__(self):
11         Gtk.Window.__init__(self, title="Conventional Gtk Window")
12         self.set_default_size(400, 200)
13
14         # Add a header to replace the title bar
15         header_bar = Gtk.HeaderBar()
16         header_bar.set_show_close_button(True)
17         header_bar.props.title = "HeaderBar Example"
18         header_bar.props.subtitle = "conventional_window_header.py"
19         self.set_titlebar(header_bar)
20
21         # Create widgets and add to the Gtk.Window
22         self.grid = Gtk.Grid()
23         self.grid.set_border_width(20)
24         self.label = Gtk.Label(label="This is a label")
25         self.grid.attach(self.label, 0,0,1,1)
26         self.add(self.grid)
27
28 win = ExampleWindow()
29 win.connect("destroy", Gtk.main_quit)
30 win.show_all()
31 Gtk.main()
```



The image shows a code editor window titled 'conventional\_window\_header.py' with Python code for a Gtk application. The code defines a class 'ExampleWindow' that inherits from 'Gtk.Window'. It sets the title to 'Conventional Gtk Window' and the default size to 400x200. It then creates a 'Gtk.HeaderBar' with the title 'HeaderBar Example' and subtitle 'conventional\_window\_header.py', and sets it as the window's titlebar. The window contains a single label 'This is a label'.

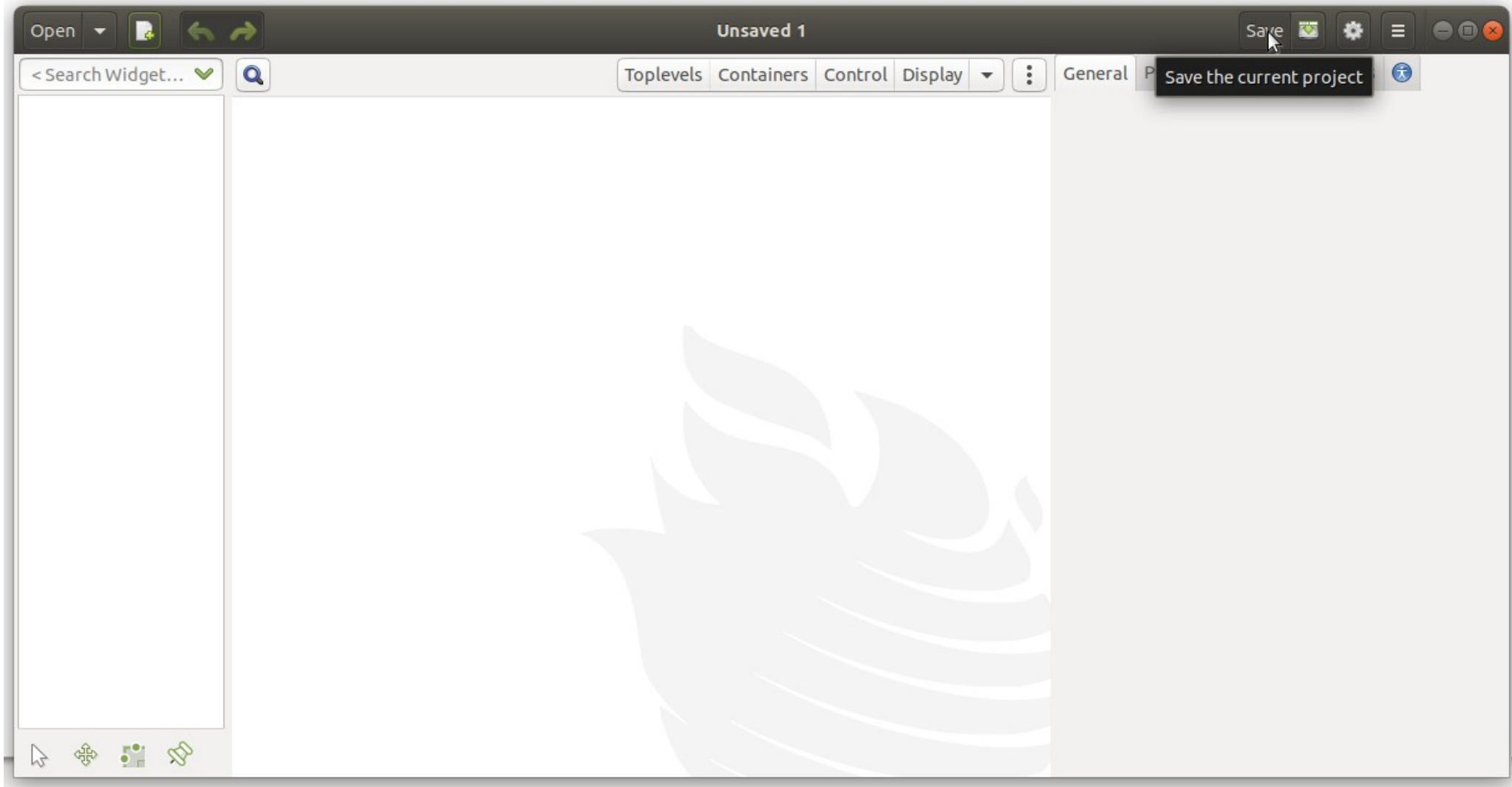
# Gtk.Header

- Install Glade. As of 2021-05-10 at V3.38.1
- Download from <https://gitlab.gnome.org/GNOME/glade>
- Hard to start...
- `$ export LD_LIBRARY_PATH=~/.local/lib/x86_64-linux-gnu/ glade`
- `$ GDK_BACKEND=x11 ./glade`
- `(glade:3683): Gdk-WARNING **: 07:49:13.985: ../../../../gdk/x11/gdkwindow-x11.c:5633 drawable is not a native X11 window`
- `GladeUI-Message: 07:49:14.348: 1 missing displayable value for GtkCellRendererAccel::accel-mode`
- `GladeUI-Message: 07:49:14.391: Glade needs artwork; a default icon will be used for the following classes:`
  - `GtkFileChooserNative` needs an icon named 'widget-gtk-filechoosernative'
  - `GThemedIcon` needs an icon named 'widget-gtk-themedicon'
  - `GladeInstantiableGtkBin` needs an icon named 'widget-gtk-bin'
  - `GFileIcon` needs an icon named 'widget-gtk-fileicon'
  - `GtkApplicationWindow` needs an icon named 'widget-gtk-applicationwindow'

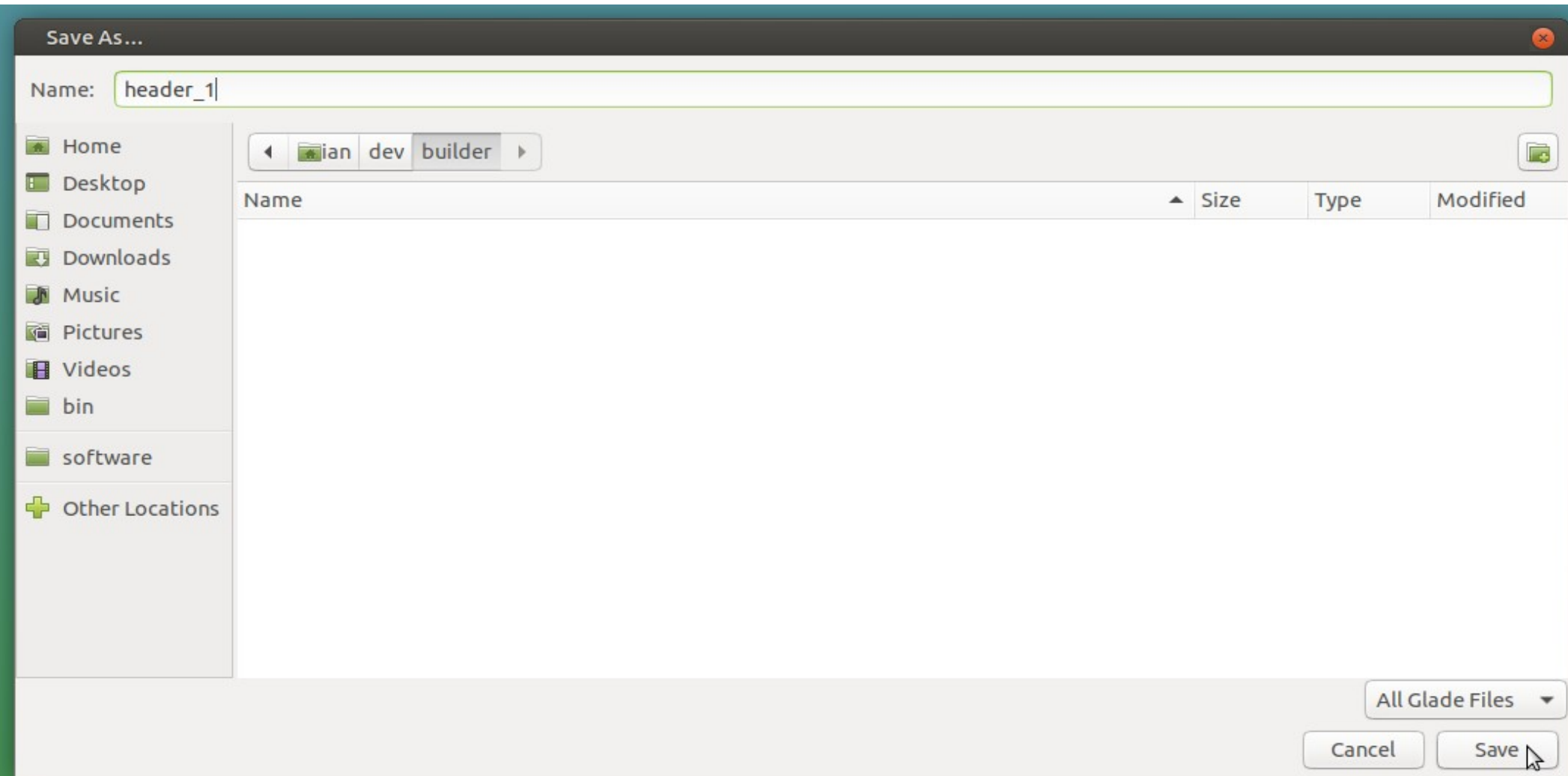
# Gtk.Header ~ Create a new project



# Gtk.Header ~ Create a new project



# Gtk.Header ~ Create a new project

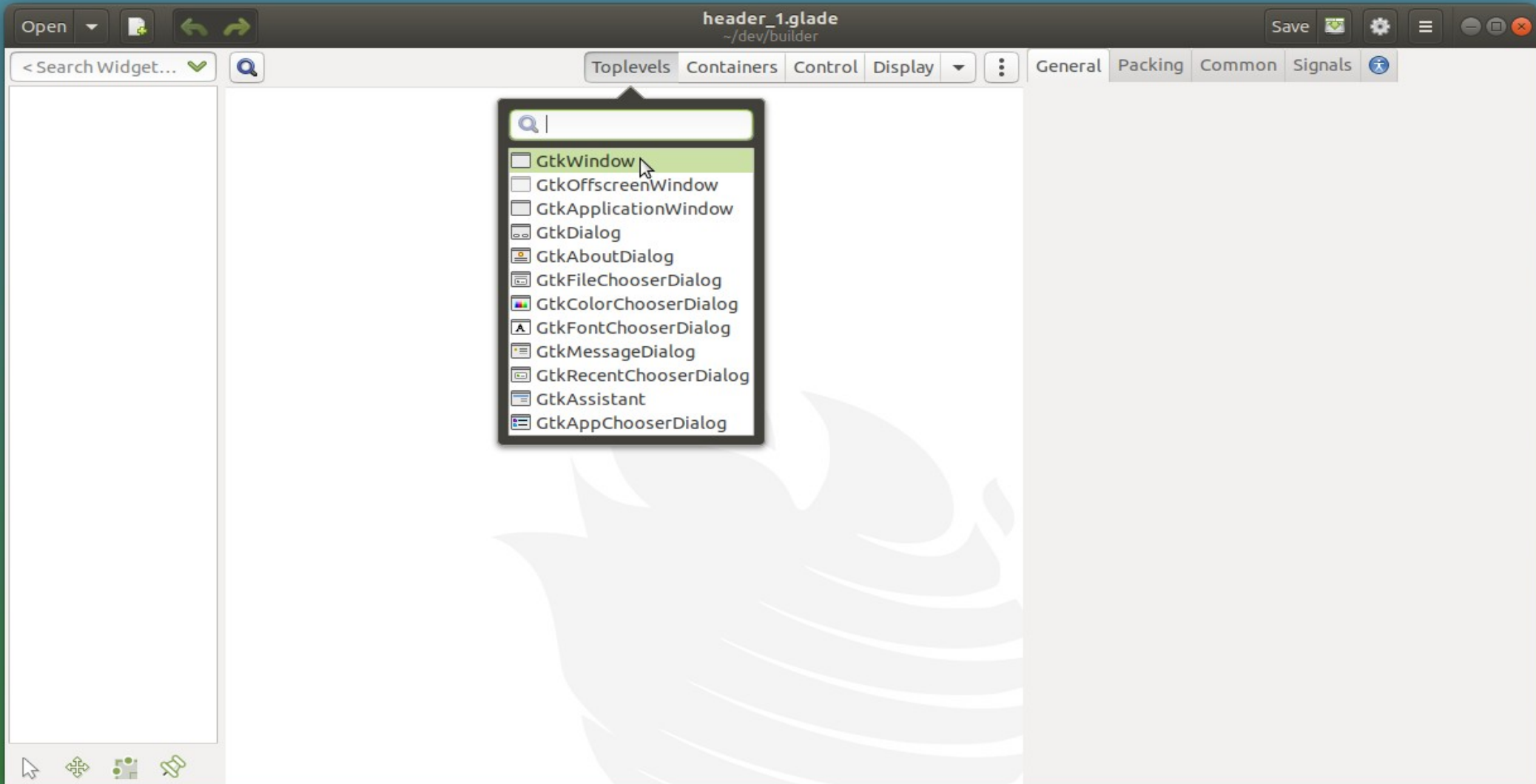


# Gtk.Header ~ Contents of UI XML file: header\_1.glade

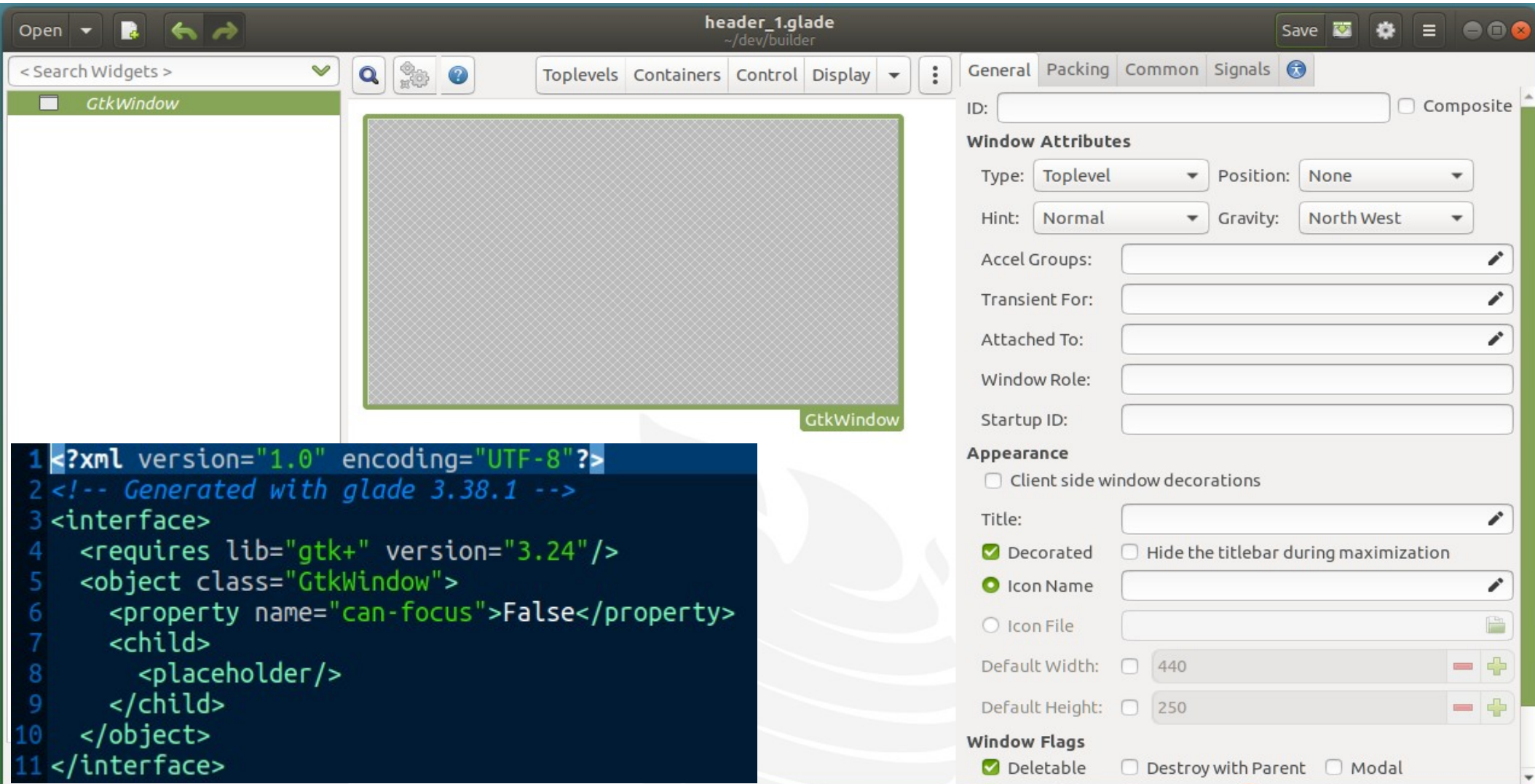
```
*header_1.glade (~/dev/builder) - Pluma
File Edit View Search Tools Documents Help
Open Save Print Undo Redo Cut Copy Paste Find Replace
*header_1.glade ✕
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Generated with glade 3.38.1 -->
3 <interface>
4   <requires lib="gtk+" version="3.24" />
5 </interface>
```



# Gtk.Header ~ Contents of UI XML file: header\_1.glade



# Gtk.Header ~ Add Gtk.Window.



The screenshot shows the Glade GUI designer interface for a file named `header_1.glade`. The top toolbar includes buttons for `Open`, `Save`, and navigation arrows. Below the toolbar is a search bar labeled `< Search Widgets >` and a list of widget categories: `Toplevels`, `Containers`, `Control`, and `Display`. The `GtkWindow` widget is selected in the widget list on the left. The central canvas displays a large, empty rectangular area representing the window, with a green border and a grid pattern. The right-hand panel shows the properties for the selected `GtkWindow` widget, organized into sections: `General`, `Packing`, `Common`, and `Signals`. The `General` section includes fields for `ID`, `Composite`, `Window Attributes` (Type, Position, Hint, Gravity), `Accel Groups`, `Transient For`, `Attached To`, `Window Role`, and `Startup ID`. The `Appearance` section includes checkboxes for `Client side window decorations`, `Decorated`, `Icon Name`, and `Icon File`, along with fields for `Title`, `Default Width`, and `Default Height`. The `Window Flags` section includes checkboxes for `Deletable`, `Destroy with Parent`, and `Modal`.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Generated with glade 3.38.1 -->
3 <interface>
4   <requires lib="gtk+" version="3.24"/>
5   <object class="GtkWindow">
6     <property name="can-focus">False</property>
7     <child>
8       <placeholder/>
9     </child>
10  </object>
11 </interface>
```

# Add ID. Check Client side window decorations

The screenshot shows the Glade GUI designer interface for a window titled `*header_1.glade`. The left pane shows a search for `GtkWindow` with `window_1` selected. The center pane shows a visual representation of the window with a yellow callout bubble saying "Check". The right pane shows the properties for `window_1`, with a yellow callout bubble saying "Provide an ID" pointing to the ID field. The bottom pane shows the XML code for the window, with yellow callout bubbles saying "Provide an ID" pointing to the `id="window_1"` attribute and "Check" pointing to the `Client side window decorations` checkbox.

Annotations:

- Check
- Provide an ID
- Provide an ID
- Check

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Generated with glade 3.38.1 -->
3 <interface>
4   <requires lib="gtk+" version="3.24"/>
5   <object class="GtkWindow" id="window_1">
6     <property name="can-focus">False</property>
7     <child>
8       <placeholder/>
9     </child>
10    <child type="titlebar">
11      <placeholder/>
12    </child>
13  </object>
14 </interface>
```

Properties:

- ID: `window_1`
- Window Attributes:
  - Type: `Toplevel`
  - Position: `None`
  - Hint: `Normal`
  - Gravity: `North West`
- Appearance:
  - ☒ Client side window decorations
  - ☒ Decorated
  - ☐ Hide the titlebar during maximization
  - ☒ Icon Name
  - ☐ Icon File
  - Default Width: `440`
  - Default Height: `250`
- Window Flags:
  - ☒ Deletable
  - ☐ Destroy with Parent
  - ☐ Modal

# Check: Default Width & Default Height

The screenshot shows the Glade GUI designer interface for a window named 'window\_1'. The left pane displays the widget hierarchy, and the right pane shows the configuration options for the selected widget. The 'General' tab is active, showing 'Window Attributes' and 'Appearance' sections. The 'Default Width' and 'Default Height' are set to 440 and 250, respectively, with checkboxes for 'Make visible'.

**Code Snippet (XML):**

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Generated with glade 3.38.1 -->
3 <interface>
4   <requires lib="gtk+" version="3.24"/>
5   <object class="GtkWindow" id="window_1">
6     <property name="can-focus">False</property>
7     <property name="default-width">440</property>
8     <property name="default-height">250</property>
9     <child>
10      <placeholder/>
11    </child>
12    <child type="titlebar">
13      <placeholder/>
14    </child>
15  </object>
16 </interface>
```

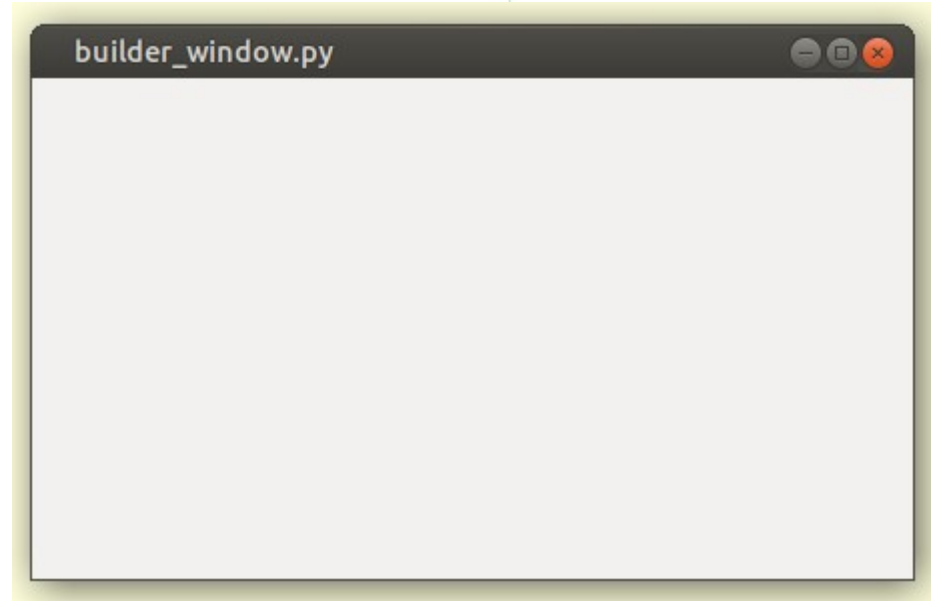
**Configuration Details:**

- Window Attributes:**
  - Type: Toplevel
  - Position: None
  - Hint: Normal
  - Gravity: North West
- Appearance:**
  - Client side window decorations: ☒
  - Decorated: ☒
  - Hide the titlebar during maximization: ☐
  - Icon Name:
  - Icon File:
  - Default Width: ☒ 440
  - Default Height: ☒ 250
- Window Flags:**
  - Deletable: ☒
  - Destroy with Parent: ☐
  - Modal: ☐



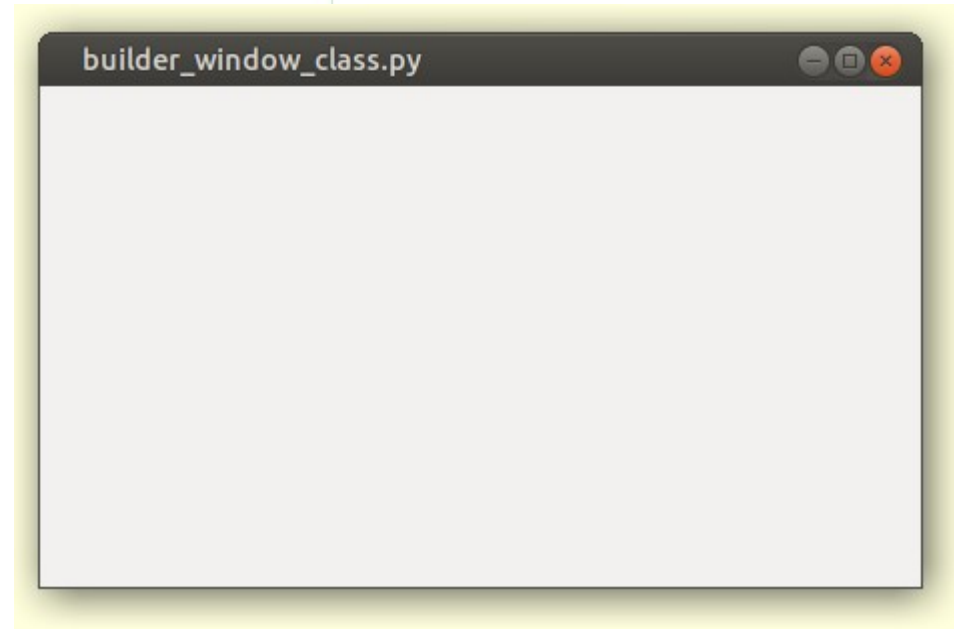
# Using Gtk.Builder and the Glade xml data.

```
1  #!/usr/bin/env python3
2  #!
3  # Glade/Builder Gtk.Window.
4  import gi
5  gi.require_version("Gtk", "3.0")
6  from gi.repository import Gtk
7
8  glade_xml = """
9  <?xml version="1.0" encoding="UTF-8"?>
10 <!-- Generated with glade 3.38.1 -->
11 <interface>
12   <requires lib="gtk+" version="3.24"/>
13   <object class="GtkWindow" id="window_1">
14     <property name="can-focus">False</property>
15     <property name="default-width">440</property>
16     <property name="default-height">250</property>
17     <child>
18       <placeholder/>
19     </child>
20     <child type="titlebar">
21       <placeholder/>
22     </child>
23   </object>
24 </interface>
25 """
26 builder = Gtk.Builder()
27 builder.add_from_string(glade_xml)
28 window = builder.get_object("window_1")
29 window.show_all()
30 Gtk.main()
```



# Place the code in a class.

```
1  #!/usr/bin/env python3
2  #!
3  # Glade/Builder Gtk.Window.
4  import gi
5  gi.require_version("Gtk", "3.0")
6  from gi.repository import Gtk
7
8  glade_xml = """
9  <?xml version="1.0" encoding="UTF-8"?>
10 <!-- Generated with glade 3.38.1 -->
11 <interface>
12   <requires lib="gtk+" version="3.24"/>
13   <object class="GtkWindow" id="window_1">
14     <property name="can-focus">False</property>
15     <property name="default-width">440</property>
16     <property name="default-height">250</property>
17     <child>
18       <placeholder/>
19     </child>
20     <child type="titlebar">
21       <placeholder/>
22     </child>
23   </object>
24 </interface>
25 """
26
27 class Window_Example():
28     def __init__(self):
29         builder = Gtk.Builder()
30         builder.add_from_string(glade_xml)
31         self.window = builder.get_object("window_1")
32         self.window.connect("destroy", Gtk.main_quit)
33         self.window.show_all()
34
35 window = Window_Example()
36 Gtk.main()
```



# Add Gtk.HeaderBar.

The screenshot shows the Glade GUI designer interface for a file named `header_1.glade` located at `~/dev/builder`. The interface is divided into several panels:

- Left Panel:** A search bar labeled "< Search Widgets >" and a list of widgets. The widget `GtkWindow` is currently selected.
- Center Panel:** A visual representation of the GUI. A green box labeled `window_1` is shown. A search bar is active over the widget list, and the `GtkHeaderBar` widget is highlighted in the list.
- Right Panel:** A configuration panel for the selected widget. It includes tabs for `General`, `Packing`, `Common`, and `Signals`.
  - General Tab:** Shows the widget ID as `window_1` and a `Composite` checkbox. Under **Window Attributes**, the `Type` is set to `Toplevel`, `Position` is `None`, `Hint` is `Normal`, and `Gravity` is `North West`. Other attributes like `Accel Groups`, `Transient For`, `Attached To`, `Window Role`, and `Startup ID` are empty.
  - Appearance Tab:** The `Client side window decorations` checkbox is checked. The `Title` field is empty. The `Decorated` checkbox is checked, and `Hide the titlebar during maximization` is unchecked. The `Icon Name` field is empty, and `Icon File` is also empty. The `Default Width` is set to 440 and the `Default Height` is set to 250, both with checkboxes for enabling the defaults.
  - Window Flags Tab:** The `Deletable` checkbox is checked, while `Destroy with Parent` and `Modal` are unchecked.

# Place the code in a class.

The screenshot shows the Glade GUI designer interface. The title bar indicates the file is `header_1.glade` located at `~/dev/builder`. The left sidebar contains a search bar and a list of widgets, with `window_1` (GtkWindow) selected. The main canvas displays a design area with a `GtkHeaderBar` widget and a `window_1` container. A search popup is open, listing various GTK widgets, with `GtkHeaderBar` highlighted. The right sidebar shows the properties panel for the selected widget, with tabs for General, Packing, Common, and Signals. The General tab is active, displaying attributes such as ID, Window Attributes (Type, Position, Hint, Gravity), Appearance (Client side window decorations, Title, Decorated, Icon Name, Icon File), and Window Flags (Deletable, Destroy with Parent, Modal).

Open [icon] [icon] [icon] [icon]

header\_1.glade  
~/dev/builder

Save [icon] [icon] [icon] [icon]

< Search Widgets > [icon]

window\_1 GtkWindow

GtkHeaderBar

Search Widgets

- GtkListBox
- GtkFlowBox
- GtkOverlay
- GtkMenuBar
- GtkToolBar
- GtkStatusbar
- GtkToolPalette
- GtkPaned
- GtkButtonBox
- GtkLayout
- GtkFixed
- GtkEventBox
- GtkExpander
- GtkViewport
- GtkScrolledWindow
- GtkAlignment
- GtkRevealer
- GtkSearchBar
- GtkHeaderBar**
- GtkStack
- GtkPopover
- GtkPopoverMenu
- GtkActionBar

General Packing Common Signals [icon]

ID: window\_1 ☐ Composite

**Window Attributes**

Type: Toplevel Position: None

Hint: Normal Gravity: North West

Accel Groups: [input] [icon]

Transient For: [input] [icon]

Attached To: [input] [icon]

Window Role: [input]

Startup ID: [input]

**Appearance**

☒ Client side window decorations

Title: [input] [icon]

☒ Decorated ☐ Hide the titlebar during maximization

☒ Icon Name [input] [icon]

☐ Icon File [input] [icon]

Default Width: ☒ 440 [icon] [icon]

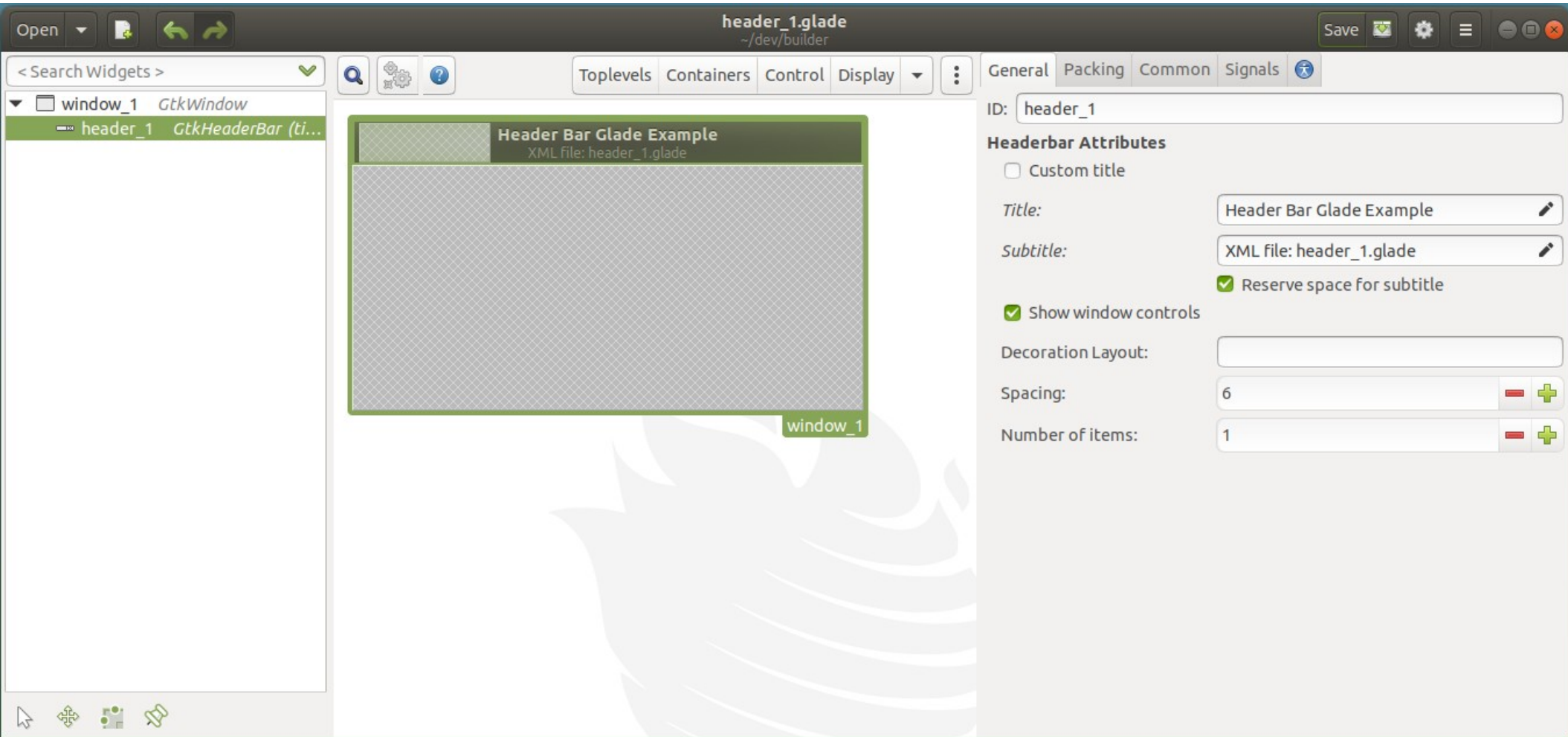
Default Height: ☒ 250 [icon] [icon]

**Window Flags**

☒ Deletable ☐ Destroy with Parent ☐ Modal

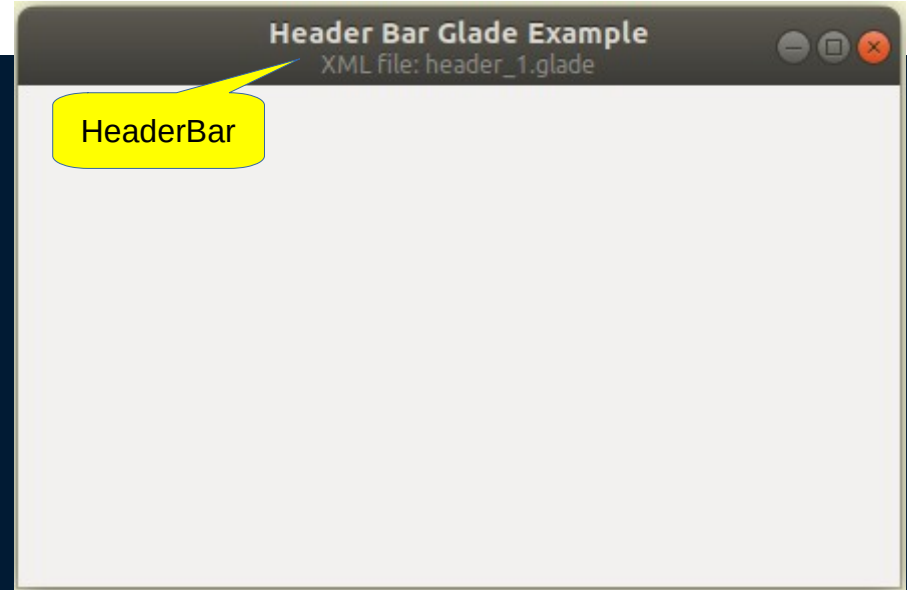


# Place the code in a class.



# Glade XML with HeaderBar displaying title and sub-title.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Generated with glade 3.38.1 -->
3 <interface>
4   <requires lib="gtk+" version="3.24"/>
5   <object class="GtkWindow" id="window_1">
6     <property name="can-focus">False</property>
7     <property name="default-width">440</property>
8     <property name="default-height">250</property>
9     <child>
10      <placeholder/>
11    </child>
12    <child type="titlebar">
13      <object class="GtkHeaderBar" id="header_1">
14        <property name="visible">True</property>
15        <property name="can-focus">False</property>
16        <property name="title" translatable="yes">Header Bar Glade Example</property>
17        <property name="subtitle" translatable="yes">XML file: header_1.glade</property>
18        <property name="show-close-button">True</property>
19      </child>
20      <placeholder/>
21    </child>
22  </object>
23 </child>
24 </object>
25 </interface>
```



# Glade xml. HeaderBar only. Gtk.Window removed.

```
8 glade_xml = ""
9 <?xml version="1.0" encoding="UTF-8"?>
10 <!-- Generated with glade 3.38.1 -->
11 <interface>
12   <requires lib="gtk+" version="3.24"/>
13   <object class="GtkHeaderBar" id="header_1">
14     <property name="visible">True</property>
15     <property name="can-focus">False</property>
16     <property name="title" translatable="yes">Header Bar Glade Example</property>
17     <property name="subtitle" translatable="yes">XML file: header_1.glade</property>
18     <property name="show-close-button">True</property>
19     <child>
20       <placeholder/>
21     </child>
22   </object>
23 </interface>
24 ""
```

# Gtk.Window Grid and Label. Also using builder for header.

```
26 class ExampleWindow(Gtk.Window):
27     def __init__(self):
28         Gtk.Window.__init__(self, title="Menu Example")
29         self.set_default_size(500, 200)
30
31         # Use Builder to read embedded xml string defining HeaderBar
32         builder = Gtk.Builder()
33         builder.add_from_string(glade_xml)
34         header = builder.get_object("header_1")
35         self.set_titlebar(header)
36
37         # Add widgets using traditional method to the Gtk.Window
38         self.grid = Gtk.Grid()
39         self.grid.set_border_width(10)
40         self.label = Gtk.Label(label="This is a label")
41         self.grid.attach(self.label, 0,0,1,1)
42         self.add(self.grid)
43
44 win = ExampleWindow()
45 win.connect("destroy", Gtk.main_quit)
46 win.show_all()
47 Gtk.main()
```



# Add a button to the header bar.

The screenshot shows the Glade GUI designer interface for a file named `*header_1.glade` located at `~/dev/builder`. The interface is divided into several panels:

- Left Panel (Widget Hierarchy):** Shows a tree view with `window_1` (GtkWindow) containing `header_1` (GtkHeaderBar).
- Center Panel (Canvas):** Displays a visual representation of the header bar. A mouse cursor is hovering over a `GtkButton` widget that has been placed on the header bar.
- Right Panel (Properties):** Contains configuration options for the selected `GtkButton` widget.
  - General:** ID is `window_1`. There is a ☐ for `Composite`.
  - Window Attributes:** Type is `Toplevel`, Position is `None`, Hint is `Normal`, and Gravity is `North West`.
  - Appearance:** ☒ `Client side window decorations`. Title is empty. ☒ `Decorated`. ☐ `Hide the titlebar during maximization`. ☒ `Icon Name`. ☐ `Icon File`. `Default Width` is 440 (checked). `Default Height` is 250 (checked).
  - Window Flags:** ☒ `Deletable`, ☒ `Resizable`, ☐ `Urgent`, ☐ `Destroy with Parent`, ☐ `Modal`, ☐ `Skip pager`, and ☐ `Skip taskbar`.



# Button

The screenshot displays the Glade GUI designer interface for editing a button widget. The main window is titled `*header_1.glade` and shows a preview of a window with a header bar and a button. The button is selected, and its properties are shown in the right-hand pane.

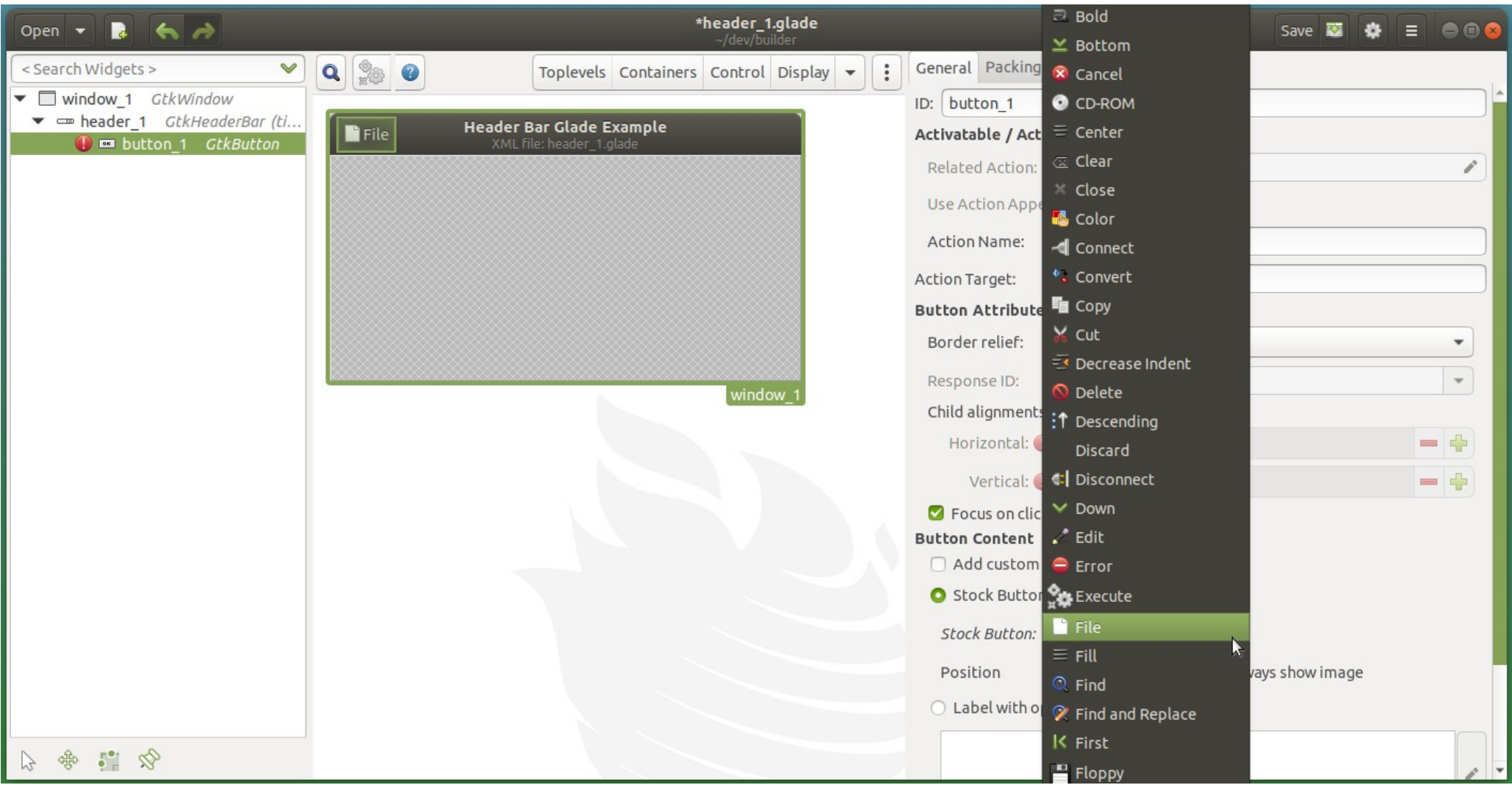
**Left Pane (Widget Hierarchy):**

- `window_1` *GtkWindow*
  - `header_1` *GtkHeaderBar (ti...*
    - `button_1` *GtkButton*

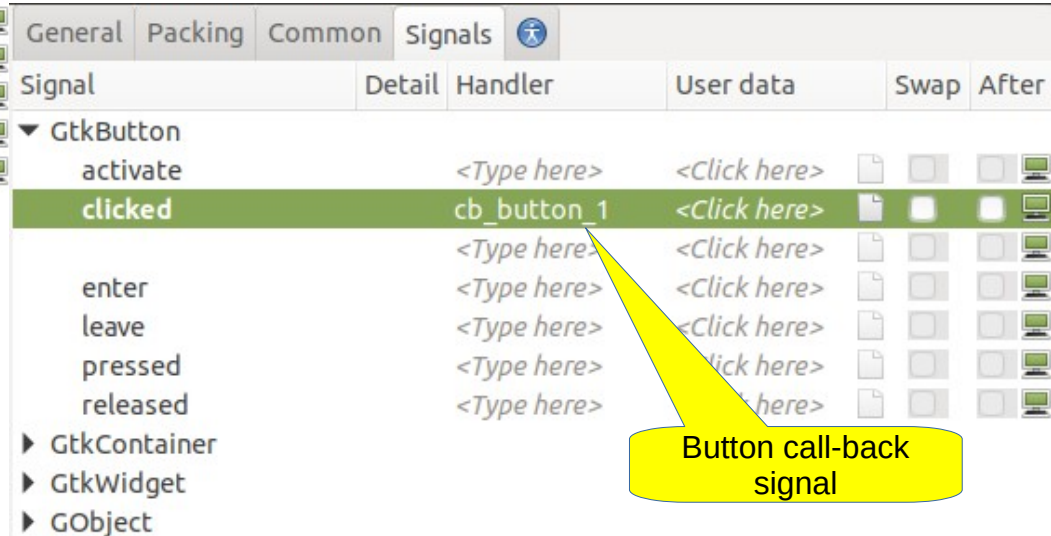
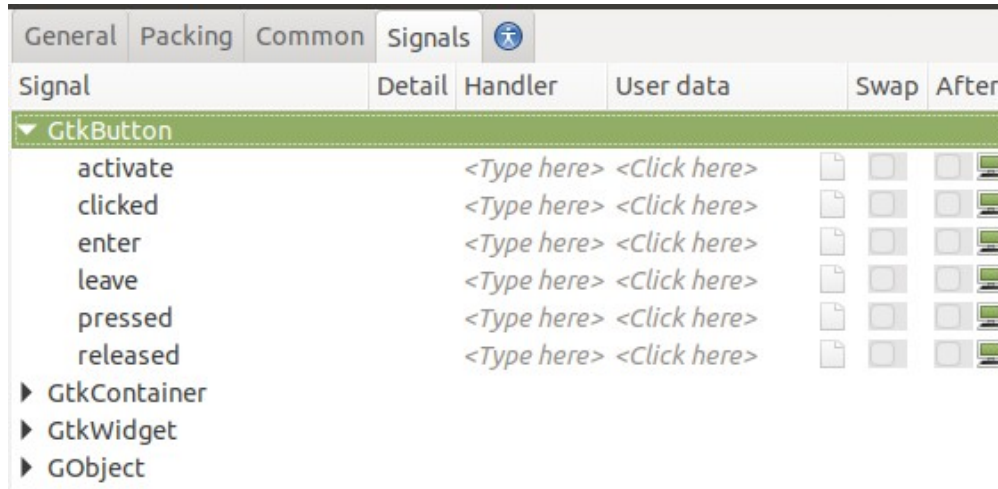
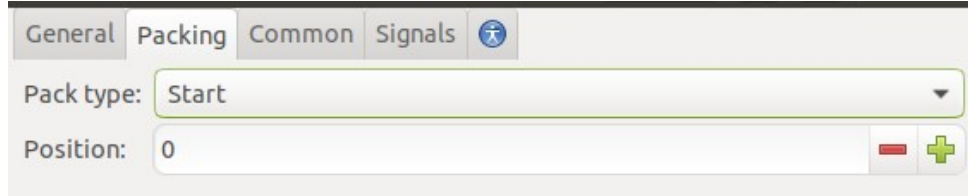
**Right Pane (Properties):**

- General** (selected tab)
- ID: `button_1`
- Activatable / Actionable**
  - Related Action: ☐ [empty]
  - Use Action Appearance: ☐ [radio]
  - Action Name: [empty]
  - Action Target: [empty]
- Button Attributes**
  - Border relief: `Normal`
  - Response ID: ☐ `0`
  - Child alignments:
    - Horizontal: ☐ `0.50`
    - Vertical: ☐ `0.50`
  - ☒ Focus on click
- Button Content**
  - ☐ Add custom content
  - ☒ Stock Button
    - Stock Button: [empty]
    - Position: `Left` ☐ Always show image
  - ☒ Label with optional image
    - button

# Gtk.Window Grid and Label. Also using builder for header.



# Gtk.Window Grid and Label. Also using builder for header.





# Additional XML for “file” button

```
13 <object class="GtkHeaderBar" id="header_1">
14   <property name="visible">True</property>
15   <property name="can-focus">False</property>
16   <property name="title" translatable="yes">Header Bar Glade Example</property>
17   <property name="subtitle" translatable="yes">XML file: header_1.glade</property>
18   <property name="show-close-button">True</property>
19   <child>
20     <object class="GtkButton" id="button_1">
21       <property name="label">gtk-file</property>
22       <property name="visible">True</property>
23       <property name="can-focus">True</property>
24       <property name="receives-default">True</property>
25       <property name="use-stock">True</property>
26       <signal name="clicked" handler="cb_button_1" swapped="no"/>
27     </object>
```

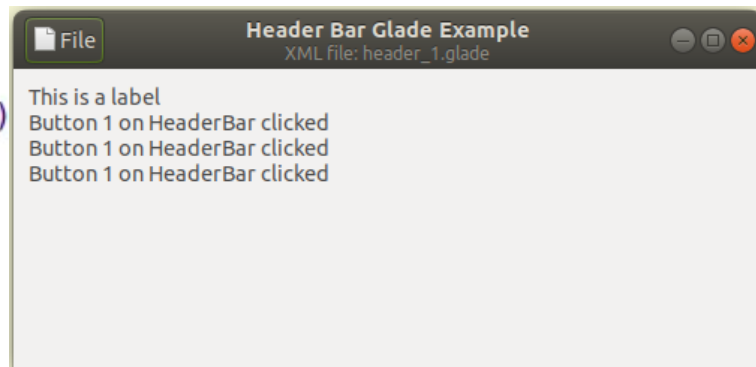
Button call-back  
signal

# Builder connects signals. Provide method for button\_1 “file”.

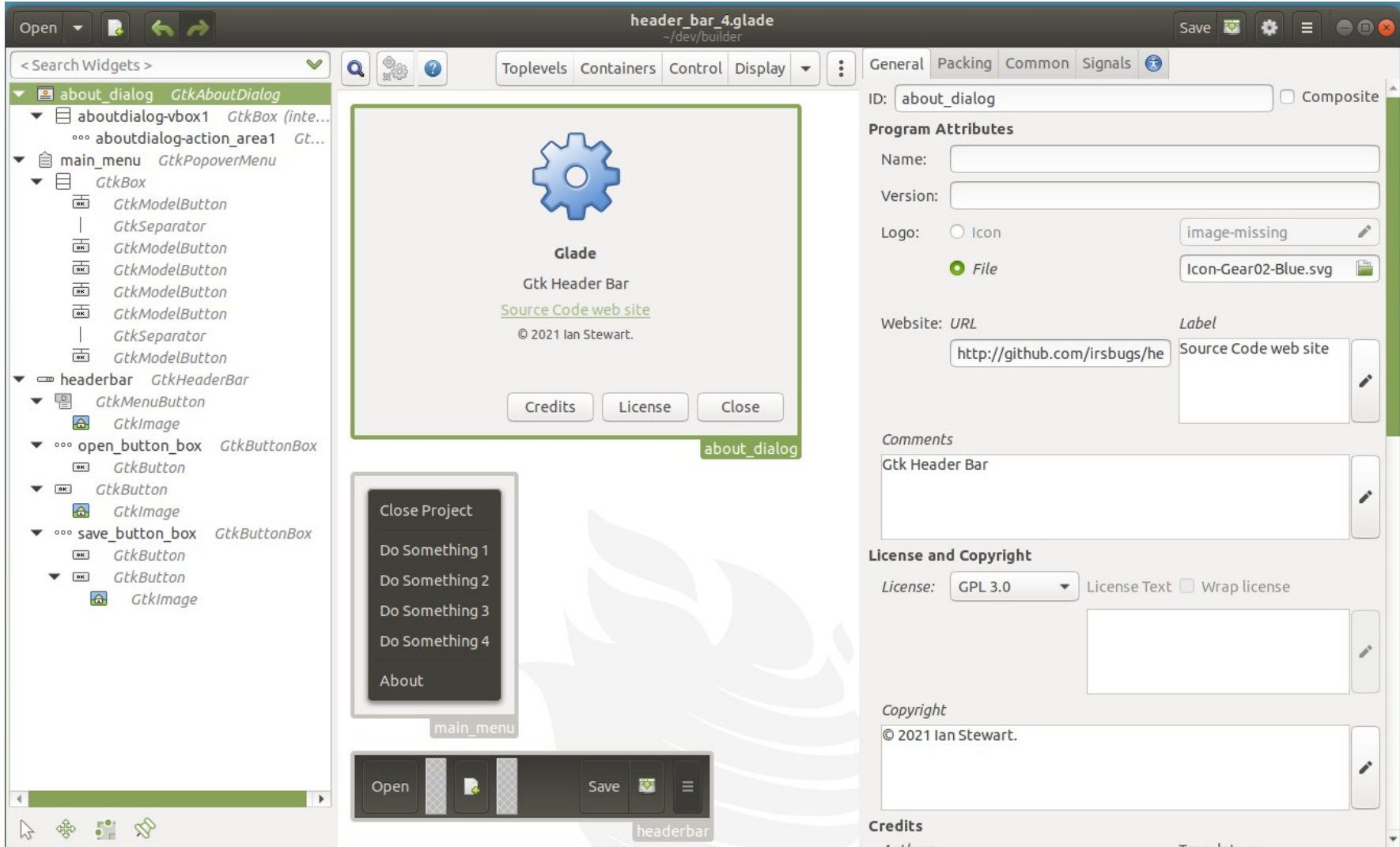
```
34 class ExampleWindow(Gtk.Window):
35     def __init__(self):
36         Gtk.Window.__init__(self, title="Menu Example")
37         self.set_default_size(500, 200)
38
39         # Use Builder to read embedded xml string defining HeaderBar
40         builder = Gtk.Builder()
41         builder.add_from_string(glade_xml)
42         header = builder.get_object("header_1")
43         builder.connect_signals(self)
44         self.set_titlebar(header)
45
46         # Add widgets using traditional method to the Gtk.Window
47         self.grid = Gtk.Grid()
48         self.grid.set_border_width(10)
49         self.label = Gtk.Label(label="This is a label")
50         self.grid.attach(self.label, 0,0,1,1)
51         self.add(self.grid)
52
53     def cb_button_1(self, button):
54         self.label.set_text(self.label.get_text() +
55                             "\nButton 1 on HeaderBar clicked")
56
57 win = ExampleWindow()
58 win.connect("destroy", Gtk.main_quit)
59 win.show_all()
60 Gtk.main()
```

Invoke call-back  
signals

Button call-back  
method



# Glade designed components for Gtk.Builder to add.



# Glade designed components for Gtk.Builder to add.

- Review source code and Demo program.

End.