

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-

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
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        # 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
28win = ExampleWindow()
29win.connect("destroy", Gtk.main_quit)
30win.show_all()
31Gtk.main()
```

HeaderBar Example

conventional_window_header.py

This is a label

Conventional Gtk Window

This is a label

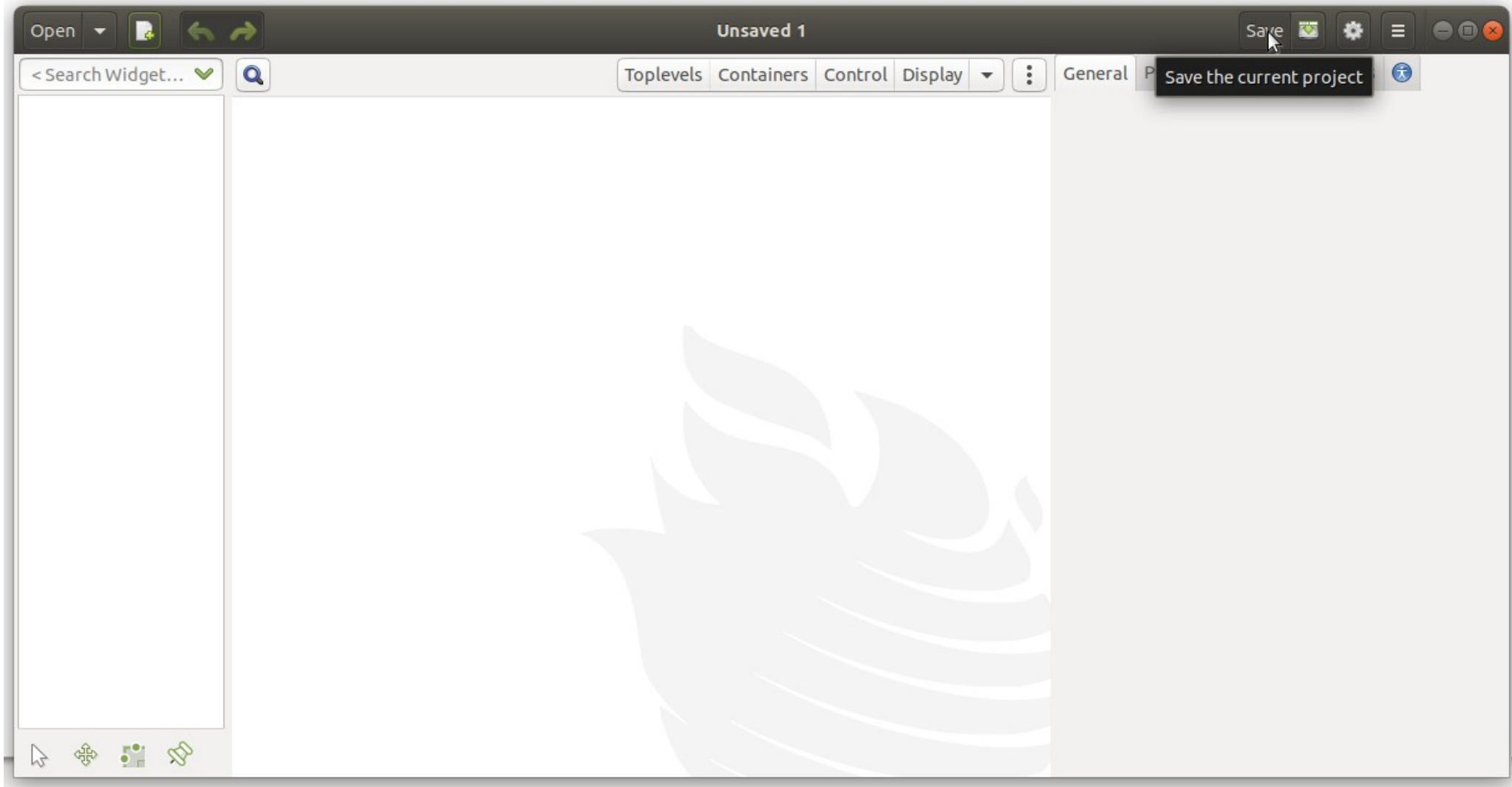
Gtk.Header

- 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'

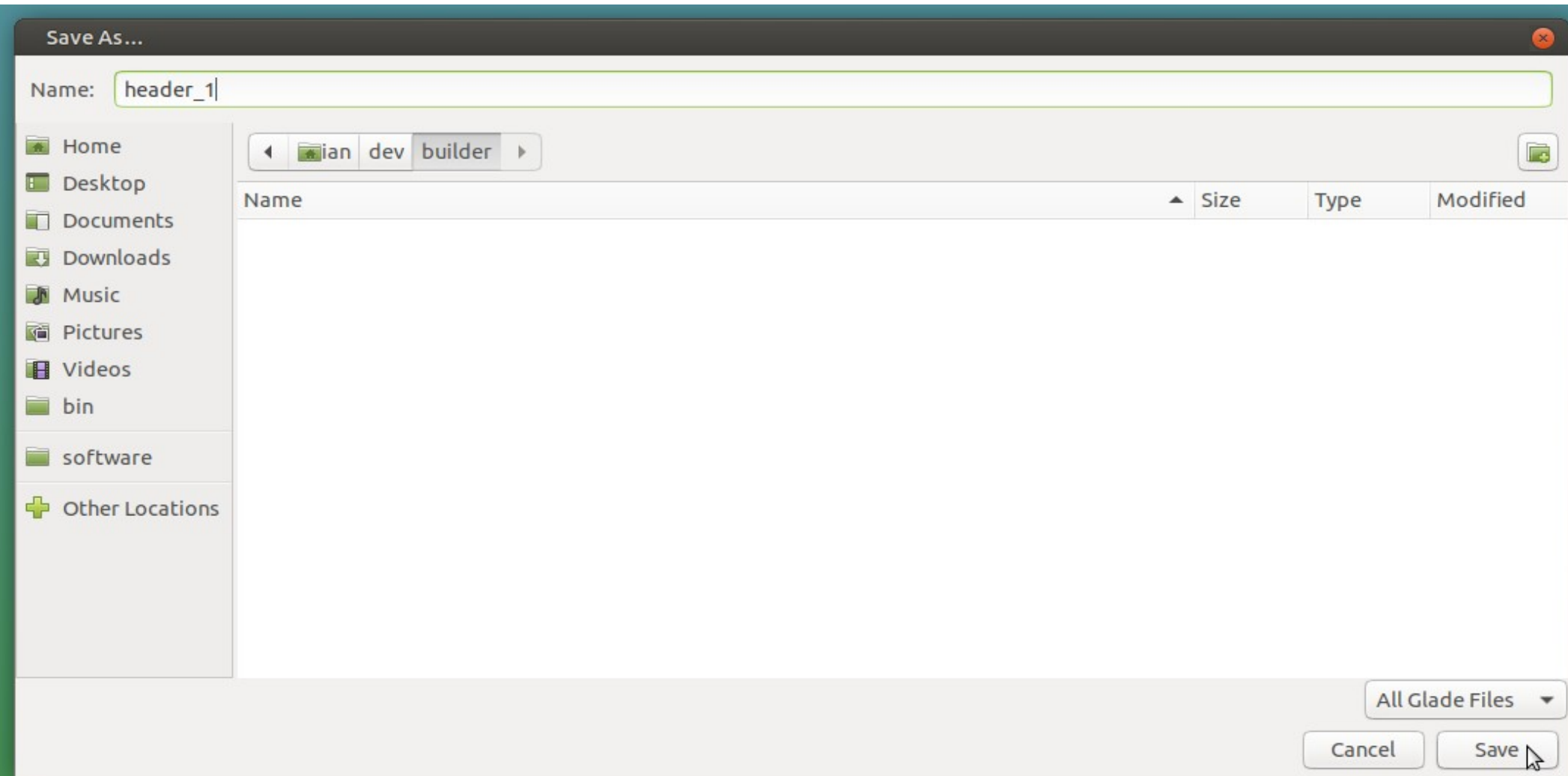
Gtk.Header ~ Create a new project



Gtk.Header ~ Create a new project



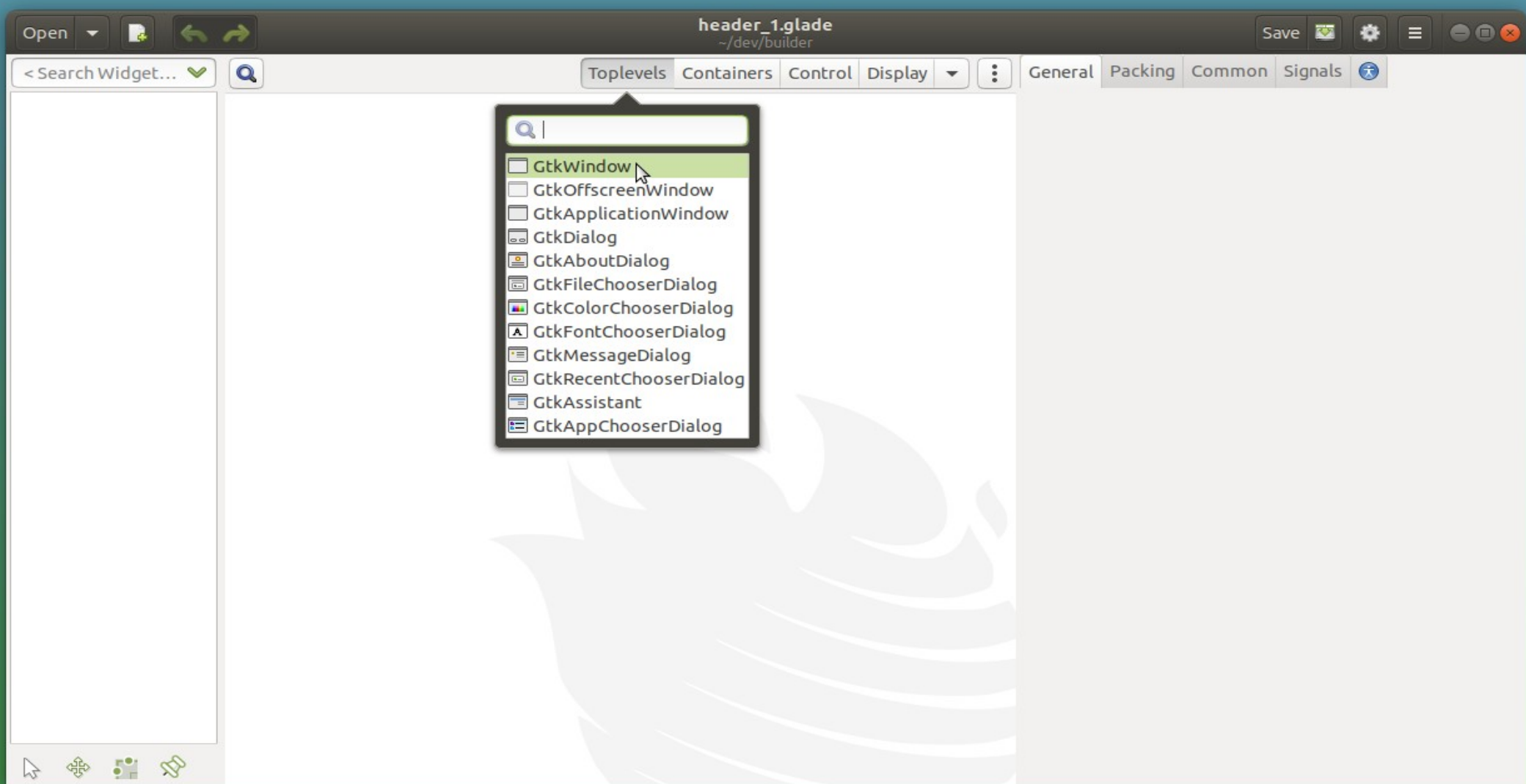
Gtk.Header ~ Create a new project



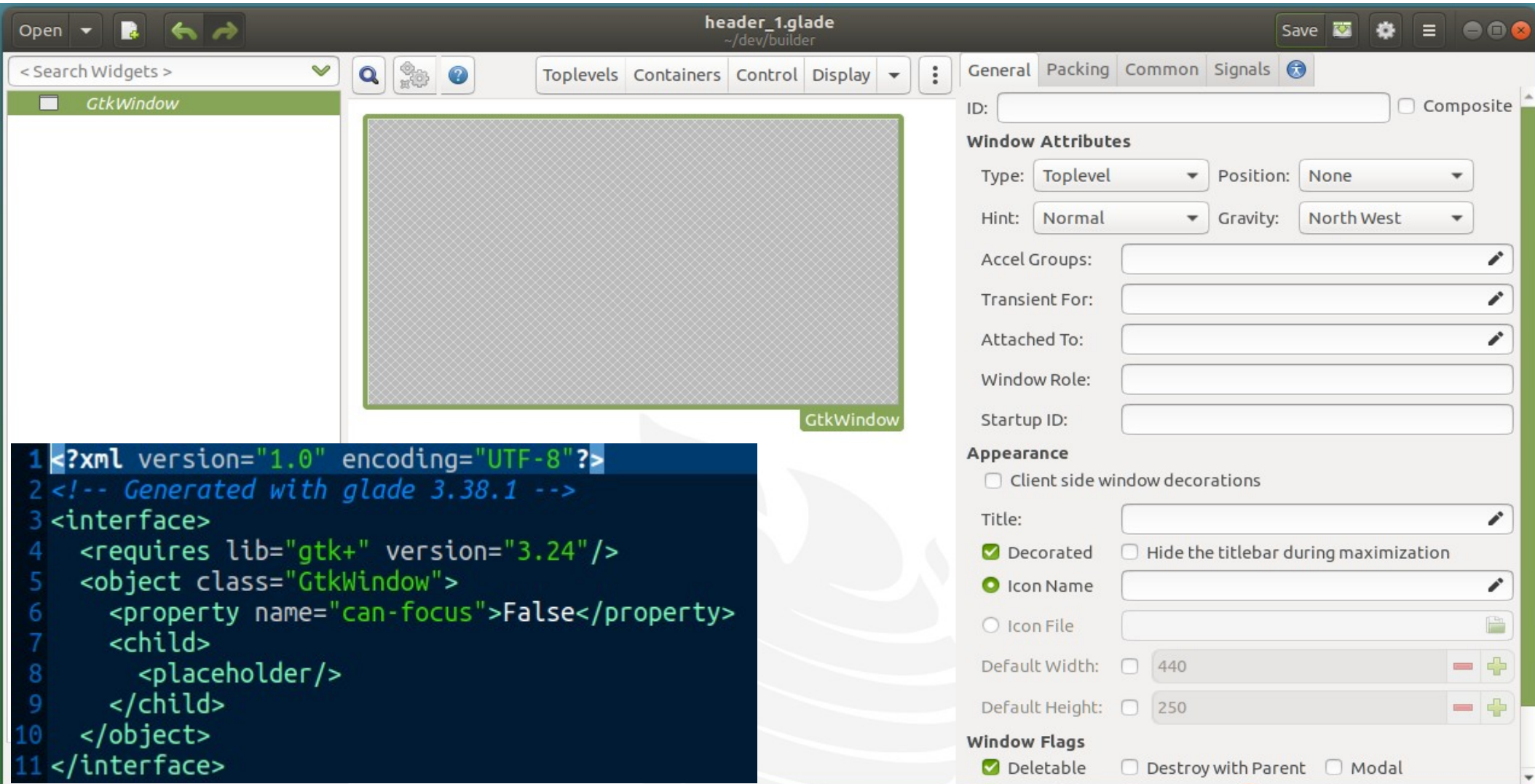
Gtk.Header ~ Contents of UI XML file: header_1.glade

```
*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. The top toolbar includes buttons for 'Open', 'Save', and navigation. The left sidebar shows a search bar and a list of widgets, with 'GtkWindow' selected. The main canvas displays a large, empty window with a green border, labeled 'GtkWindow' at the bottom right. The right sidebar contains configuration tabs: 'General', 'Packing', 'Common', and 'Signals'. The 'General' tab is active, showing various attributes and appearance settings.

General Packing Common Signals

ID: ☐ Composite

Window Attributes

Type: Position:

Hint: Gravity:

Accel Groups:

Transient For:

Attached To:

Window Role:

Startup ID:

Appearance

☐ Client side window decorations

Title:

☒ Decorated ☐ Hide the titlebar during maximization

☒ Icon Name

☐ Icon File

Default Width: ☐ 440

Default Height: ☐ 250

Window Flags

☒ Deletable ☐ Destroy with Parent ☐ 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. The main window is titled `*header_1.glade` and is located at `~/dev/builder`. The left sidebar shows a search bar and a list of widgets, with `window_1` (GtkWindow) selected. The center canvas displays a visual representation of the window, labeled `window_1`. The right sidebar contains the 'Properties' panel, which is currently showing the 'General' tab. The 'ID' field is set to `window_1`. The 'Window Attributes' section includes fields for Type (Toplevel), Position (None), Hint (Normal), and Gravity (North West). The 'Appearance' section has a checked checkbox for 'Client side window decorations'. The 'Window Flags' section has a checked checkbox for 'Deletable'. A code editor window is overlaid on the bottom left, showing the XML code for the window.

```
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>
```

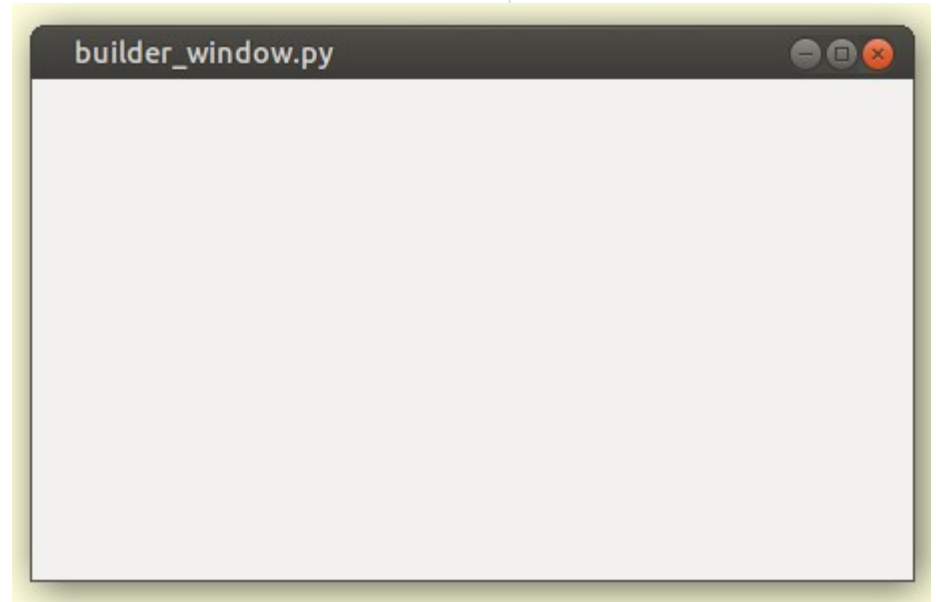
Check: Default Width & Default Height

The screenshot displays the Glade GUI designer interface for a file named `header_1.glade`. The central canvas shows a window widget labeled `window_1`. On the left, the 'Search Widgets' pane lists `window_1` as a `GtkWindow`. On the right, the 'Properties' pane is open, showing the 'General' tab. Under 'Window Attributes', the 'Type' is set to 'Toplevel', 'Position' to 'None', 'Hint' to 'Normal', and 'Gravity' to 'North West'. In the 'Appearance' section, 'Client side window decorations' is checked. Under 'Window Flags', 'Deletable' is checked. The 'Default Width' is set to 440 and 'Default Height' to 250, both with checkboxes and increment/decrement buttons. The bottom-left pane shows the XML code for the window definition.

```
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>
```

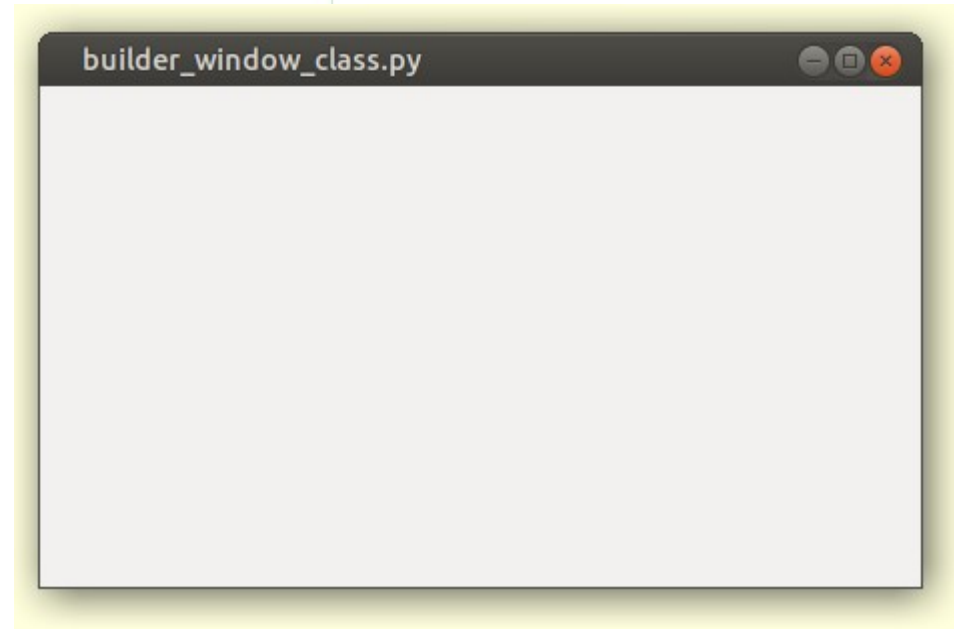

Using Gtk.Builder and the Glade xml data.

```
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 under the `window_1` container.
- Center Panel:** A visual representation of the GUI. A search bar is visible at the top, and a widget palette is open, displaying a list of GTK widgets. The widget `GtkHeaderBar` is highlighted in the list. The palette also includes other widgets like `GtkListBox`, `GtkFlowBox`, `GtkOverlay`, `GtkMenuBar`, `GtkToolBar`, `GtkStatusBar`, `GtkToolPalette`, `GtkPaned`, `GtkButtonBox`, `GtkLayout`, `GtkFixed`, `GtkEventBox`, `GtkExpander`, `GtkViewport`, `GtkScrolledWindow`, `GtkAlignment`, `GtkRevealer`, `GtkSearchBar`, `GtkStack`, `GtkPopover`, `GtkPopoverMenu`, and `GtkActionBar`.
- Right Panel:** A properties panel for the selected widget. It has tabs for `General`, `Packing`, `Common`, and `Signals`. The `General` tab is active, showing the following settings:
 - ID:** `window_1`
 - Window Attributes:**
 - Type: `Toplevel`
 - Position: `None`
 - Hint: `Normal`
 - Gravity: `North West`
 - Appearance:**
 - ☒ Client side window decorations
 - Title: (empty field)
 - ☒ Decorated
 - ☐ Hide the titlebar during maximization
 - ☒ Icon Name
 - ☐ Icon File
 - Default Width: ☒ 440
 - Default Height: ☒ 250
 - Window Flags:**
 - ☒ Deletable
 - ☐ Destroy with Parent
 - ☐ Modal

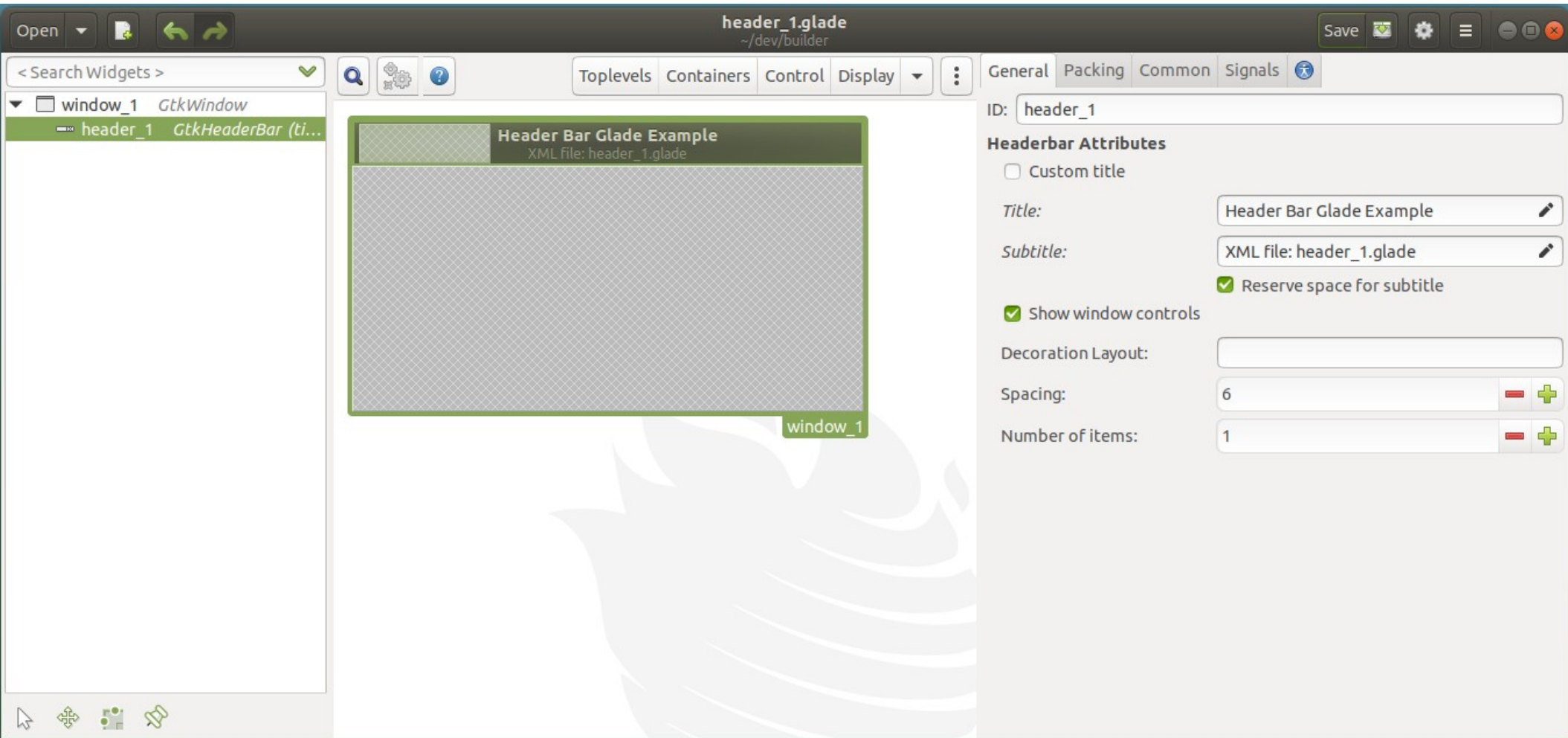
Place the code in a class.

The screenshot displays the Glade GUI designer interface. At the top, the title bar shows the file name `header_1.glade` and the path `~/dev/builder`. The main window is divided into three primary sections:

- Left Panel (Widget Palette):** A search bar at the top is followed by a list of widget categories: `window_1` (GtkWindow), `GtkHeaderBar`, and others. A search icon is visible in the top right of this panel.
- Center Panel (Design Canvas):** A visual representation of the GUI. It shows a `GtkHeaderBar` widget being added to a container. A search bar is also present above the widget list in this panel.
- Right Panel (Properties):** A detailed configuration area for the selected widget. It includes tabs for `General`, `Packing`, `Common`, and `Signals`. The `General` tab is active, showing fields for `ID` (set to `window_1`), `Window Attributes` (Type: `Toplevel`, Position: `None`, Hint: `Normal`, Gravity: `North West`), `Appearance` (Client side window decorations checked, Title: empty, Decorated checked, Hide the titlebar during maximization unchecked, Icon Name: empty, Icon File: empty), and `Window Flags` (Deletable checked, Destroy with Parent unchecked, Modal unchecked).

The bottom of the window features a standard Linux desktop environment with a taskbar containing icons for a file manager, terminal, and other applications.

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.

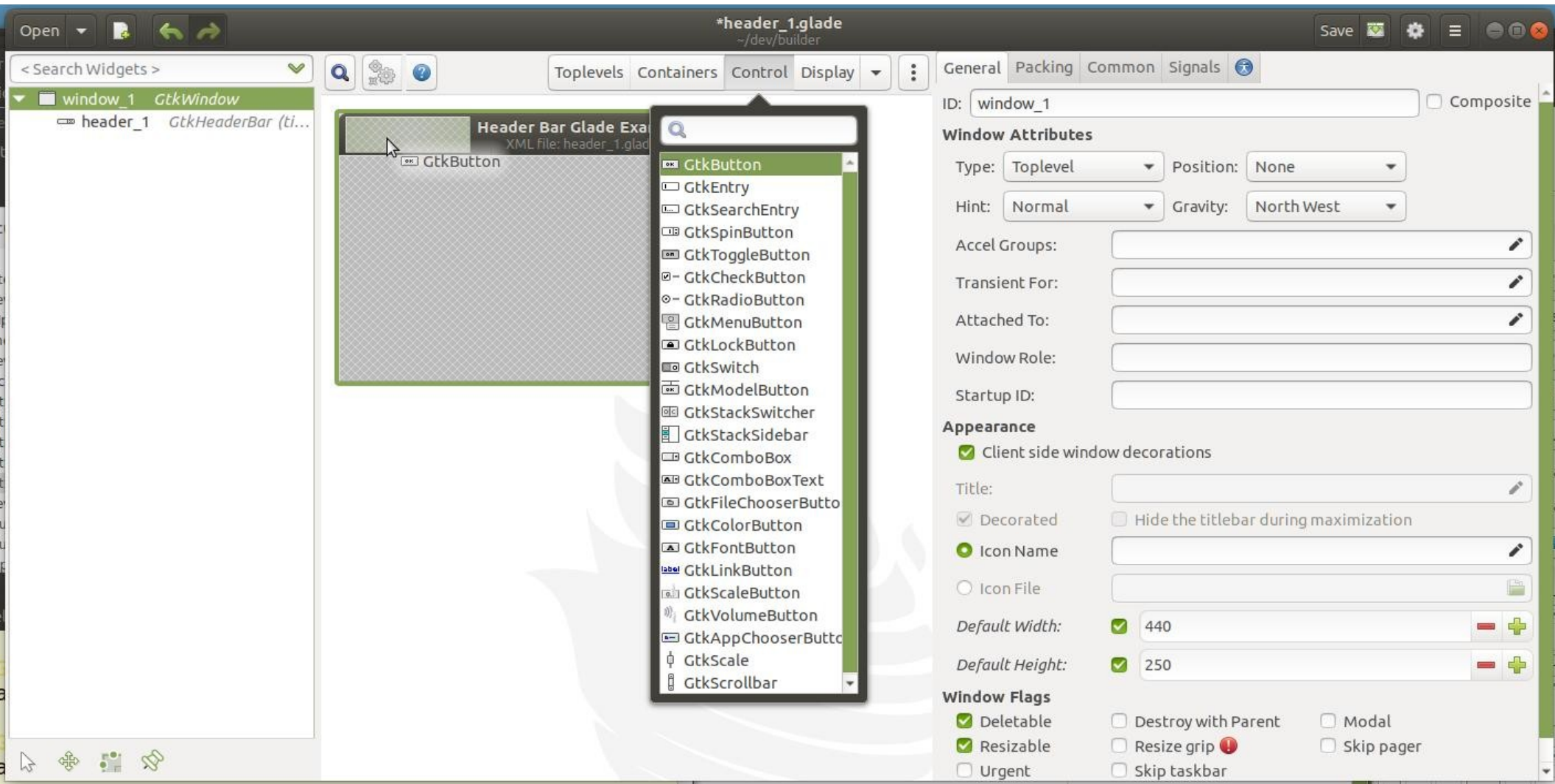
```
8 glade_xml = """
9 <?xml version="1.0" encoding="UTF-8"?>
10 <!-- Generated with glade 3.38.1 -->
11 <interface>
12   <requires lib="gtk+" version="3.24"/>
13   <object class="GtkHeaderBar" id="header_1">
14     <property name="visible">True</property>
15     <property name="can-focus">False</property>
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.



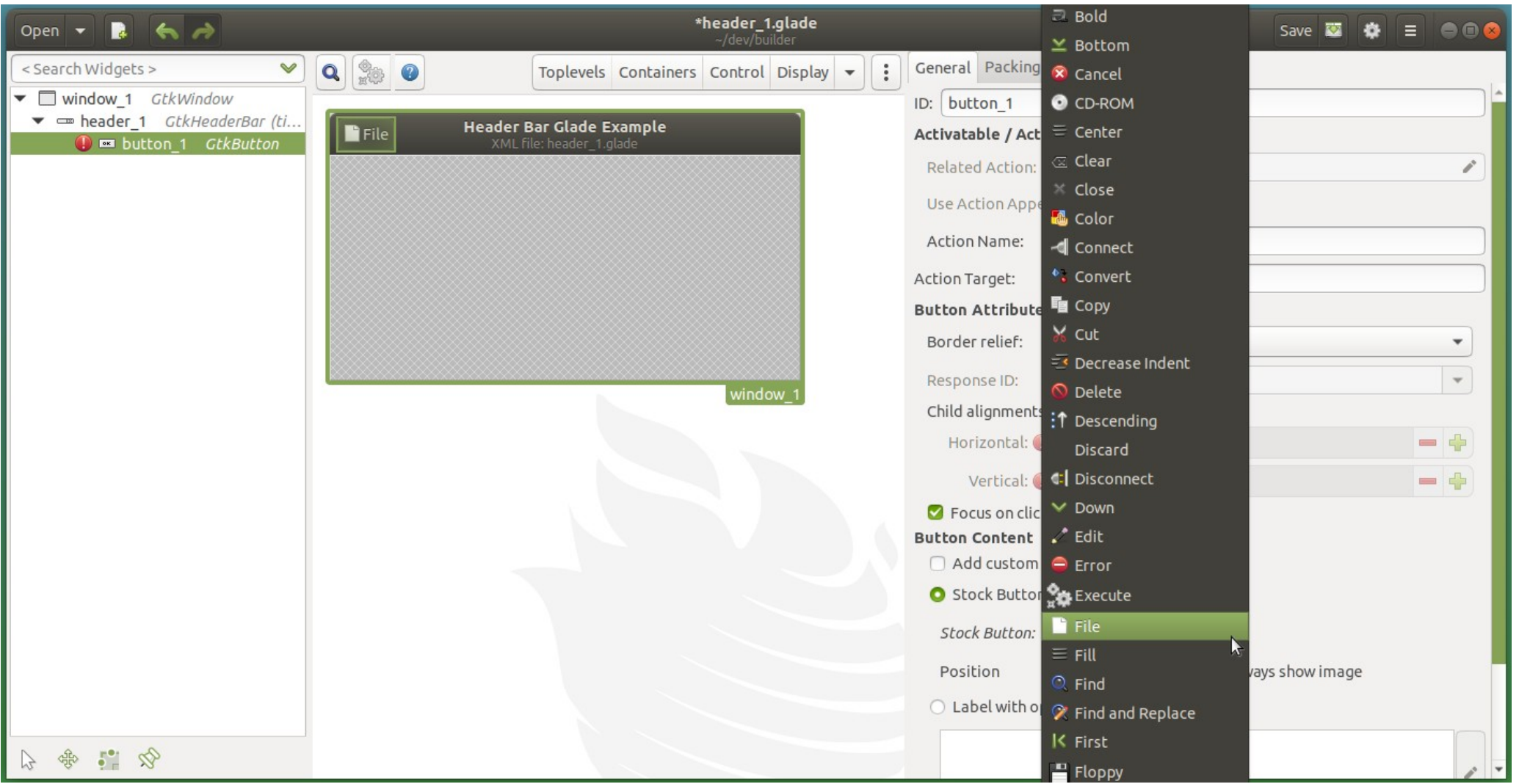
Button

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

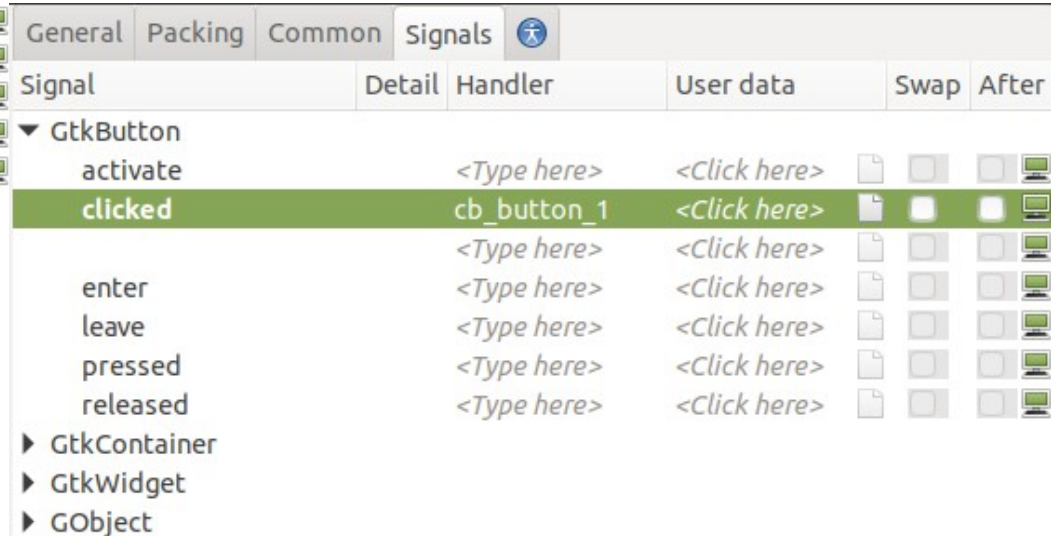
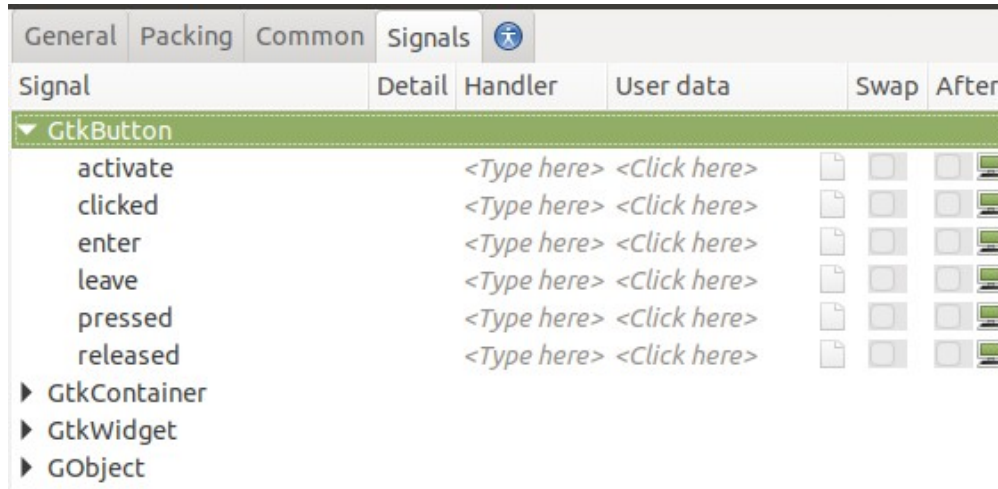
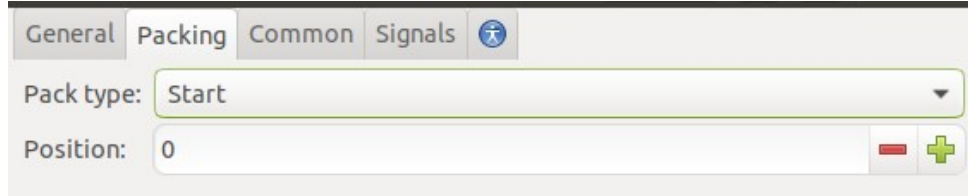
- Left Panel (Widget Hierarchy):** Shows a tree view with the following structure:
 - `window_1` (*GtkWindow*)
 - `header_1` (*GtkHeaderBar* (title bar))
 - `button_1` (*GtkButton*)

- Center Panel (Preview):** Displays a visual representation of the widget hierarchy. A window titled "Header Bar Glade Example" contains a header bar with a button labeled "button". The XML file for this design is `header_1.glade`. The button is highlighted with a green border.
- Right Panel (Properties):** Contains configuration options for the selected widget (`button_1`).
- General:** The ID is set to `button_1`.
- Activatable / Actionable:** Includes fields for "Related Action", "Use Action Appearance", "Action Name", and "Action Target".
- Button Attributes:** Includes a "Border relief" dropdown set to "Normal", a "Response ID" dropdown set to "0", and "Child alignments" for horizontal and vertical (both set to 0.50).
- Focus:** The "Focus on click" checkbox is checked.
- Button Content:** Includes options for "Add custom content", "Stock Button" (selected), and "Label with optional image". The "Stock Button" dropdown is empty, and the "Position" is set to "Left".

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



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

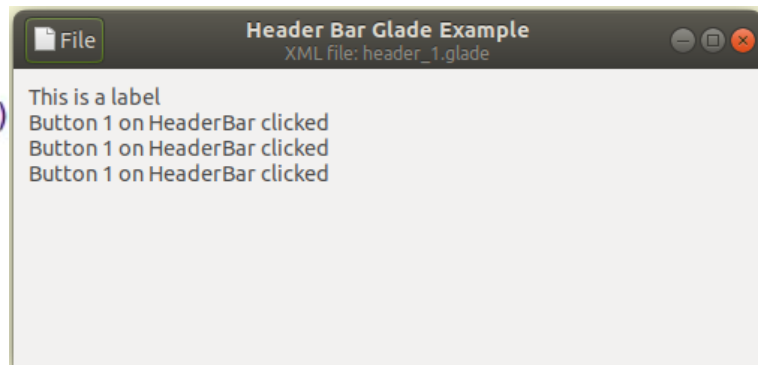


Additional XML for “file” button

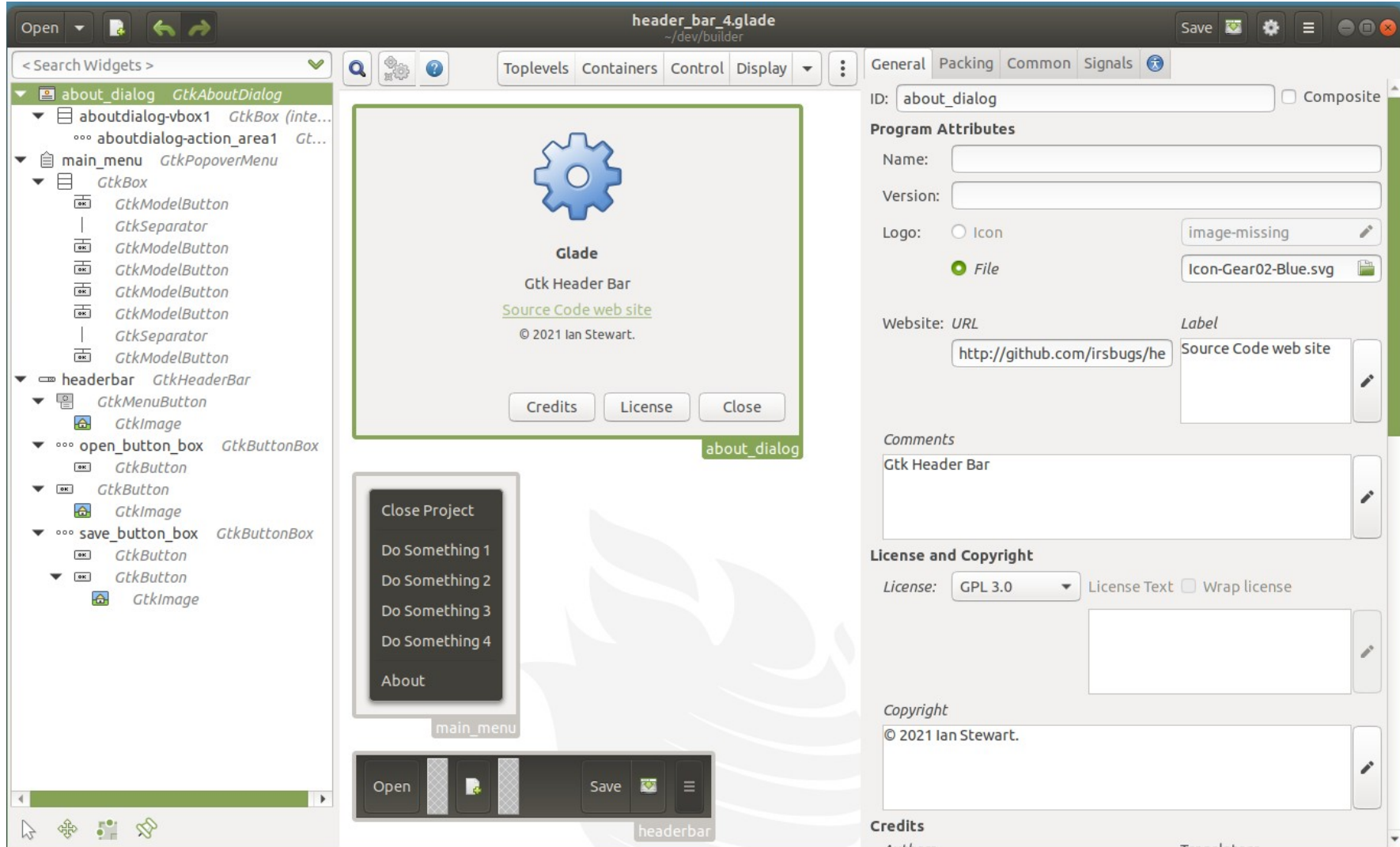
```
13 <object class="GtkHeaderBar" id="header_1">
14   <property name="visible">True</property>
15   <property name="can-focus">False</property>
16   <property name="title" translatable="yes">Header Bar Glade Example</property>
17   <property name="subtitle" translatable="yes">XML file: header_1.glade</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>
```

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()
```



Glade designed components for Gtk.Builder to add.



Glade designed components for Gtk.Builder to add.

- Review source code and Demo program.

End.