

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.

Hamilton Python Users Group

Ian Stewart

2021-05-10

Gtk.Header ~ Gtk Window

conventional_window.py ✕

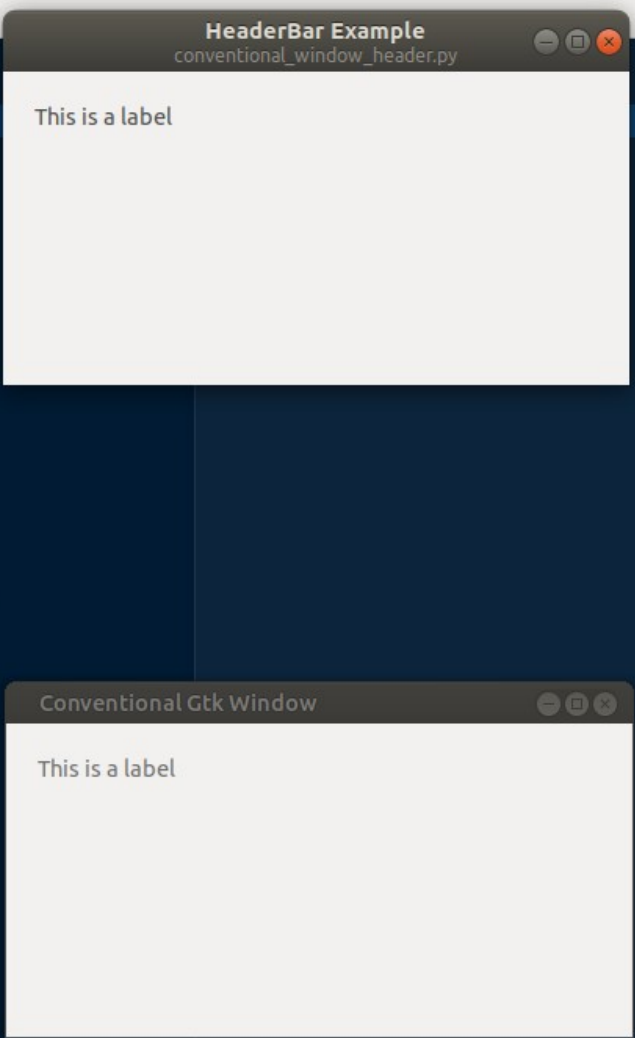
```
1#!/usr/bin/env python3
2#!
3# Conventional Gtk.Window
4import gi
5gi.require_version("Gtk", "3.0")
6from gi.repository import Gtk
7
8
9class ExampleWindow(Gtk.Window):
10    def __init__(self):
11        Gtk.Window.__init__(self, title="Conventional Gtk Window")
12        self.set_default_size(400, 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 titlebar. The window contains a single label 'This is a label'.

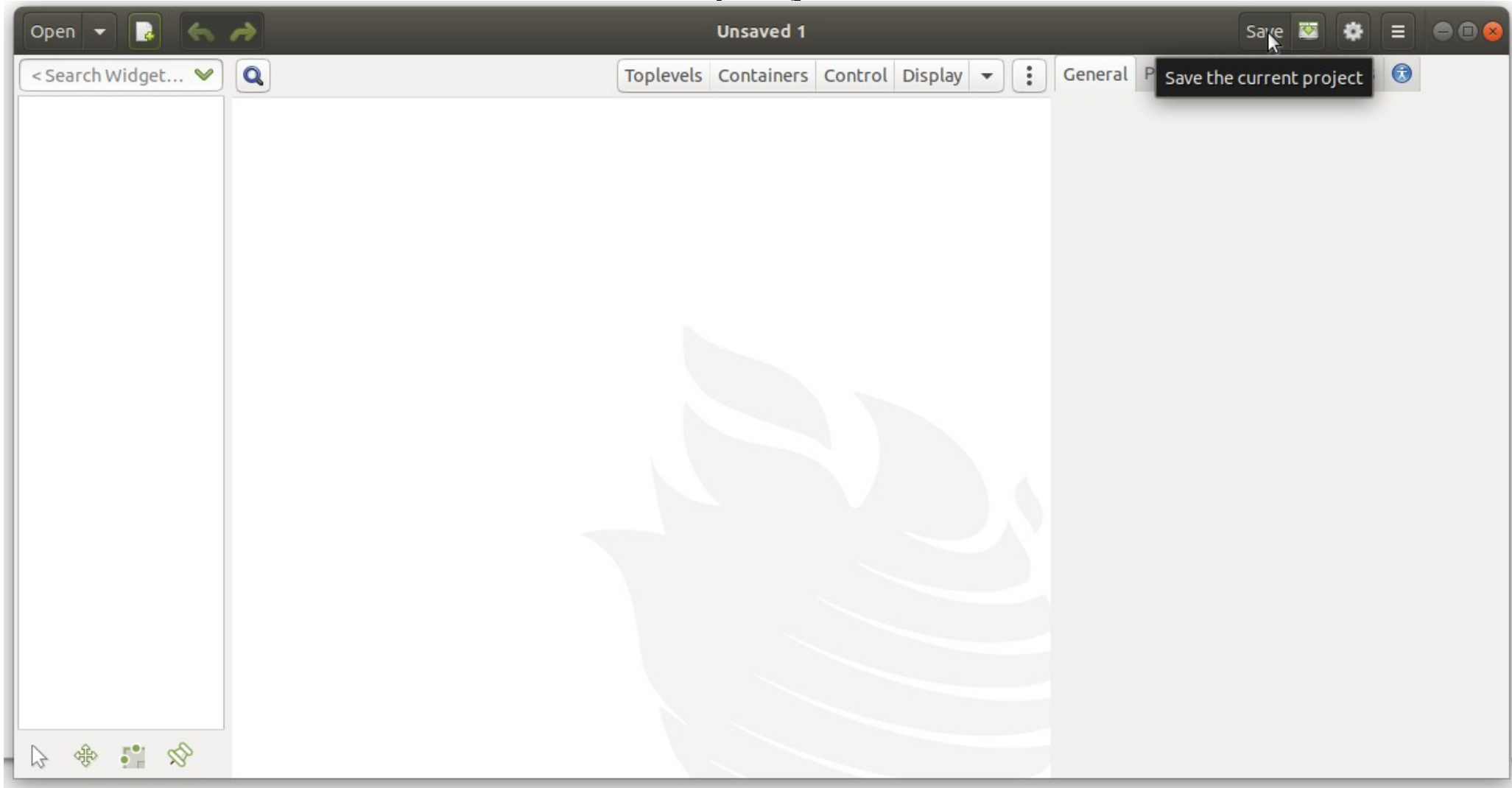
Gtk.Header

- 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'

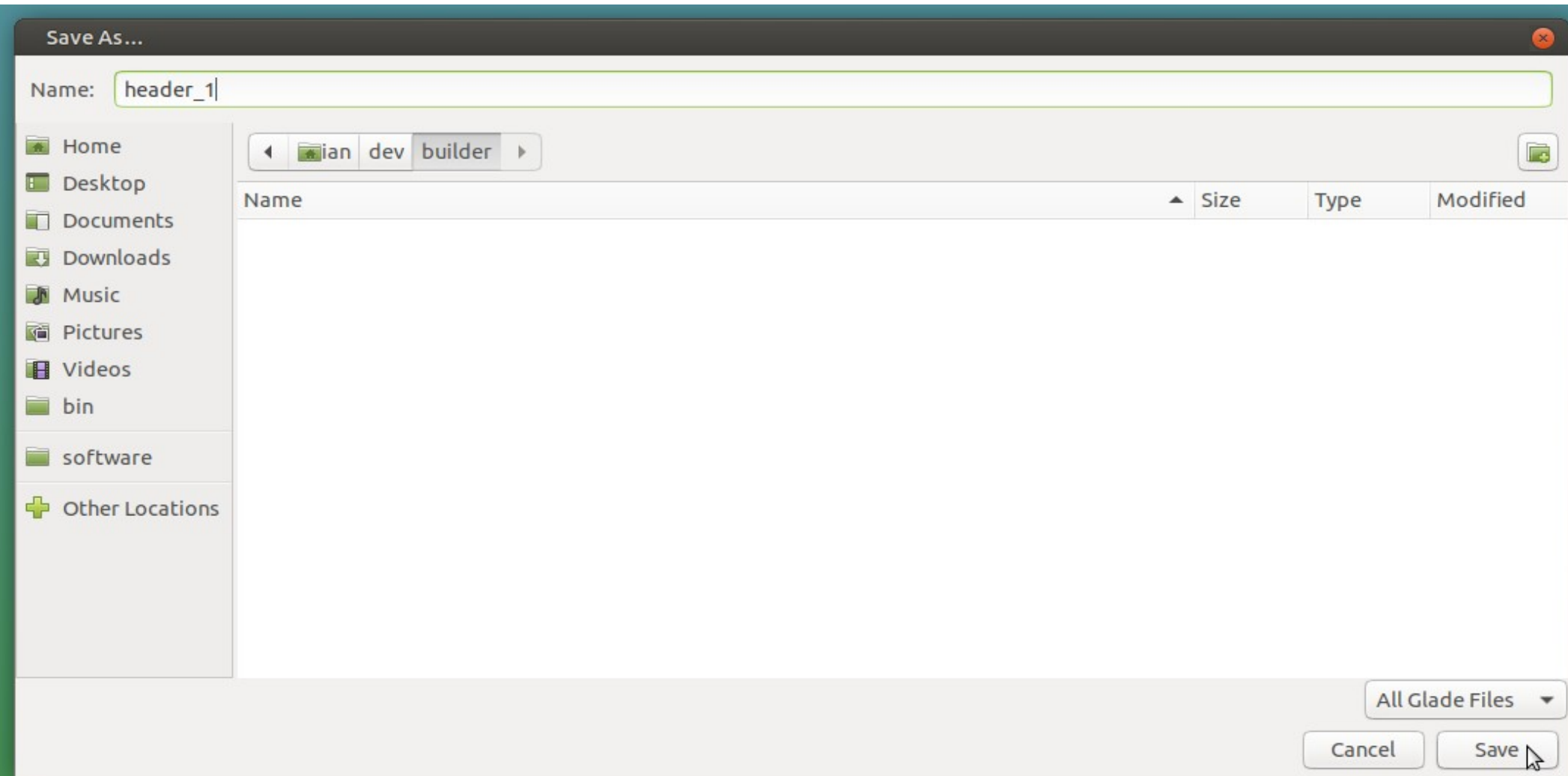
Gtk.Header ~ Create a new project



Gtk.Header ~ Create a new project



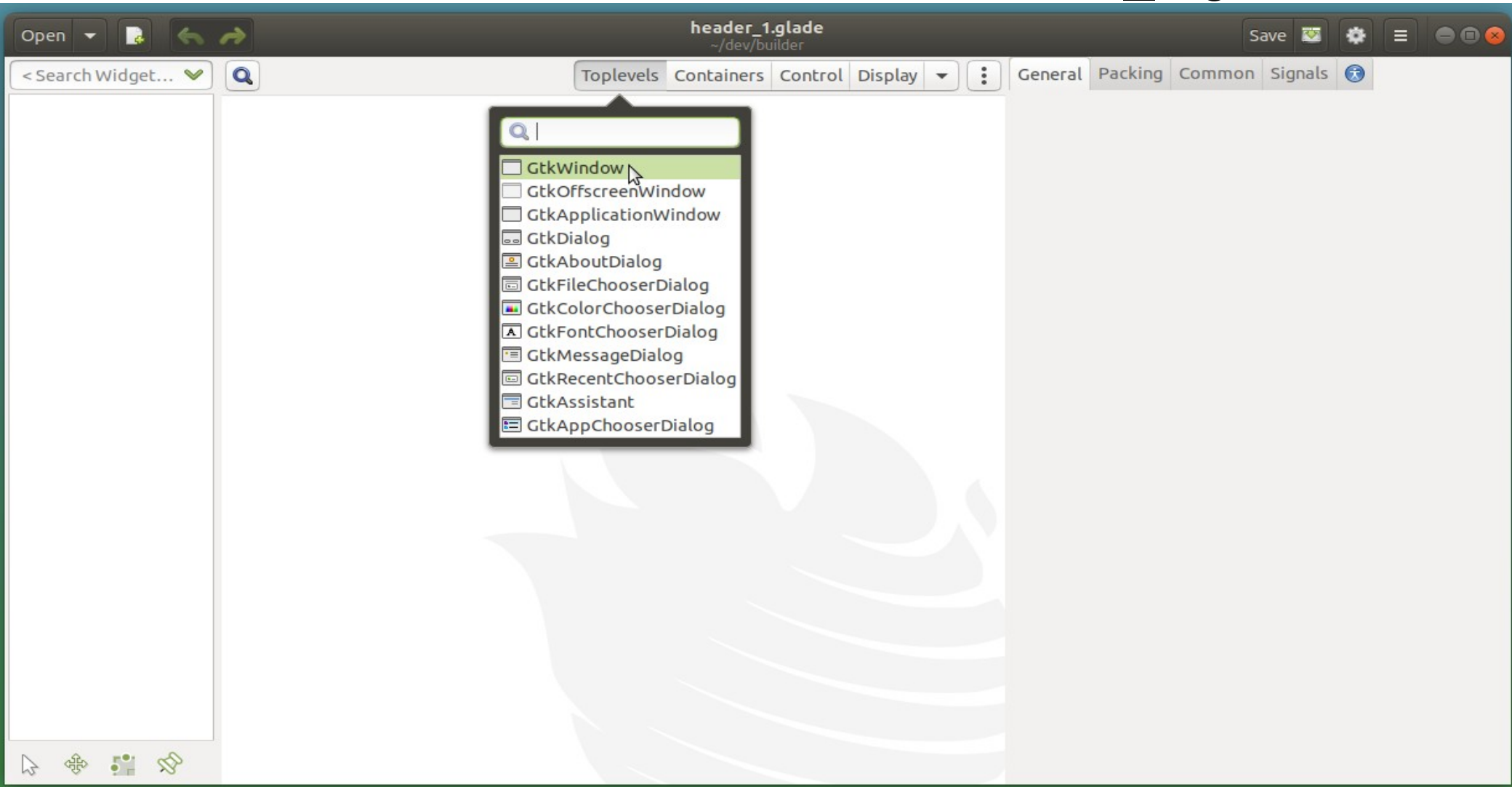
Gtk.Header ~ Create a new project



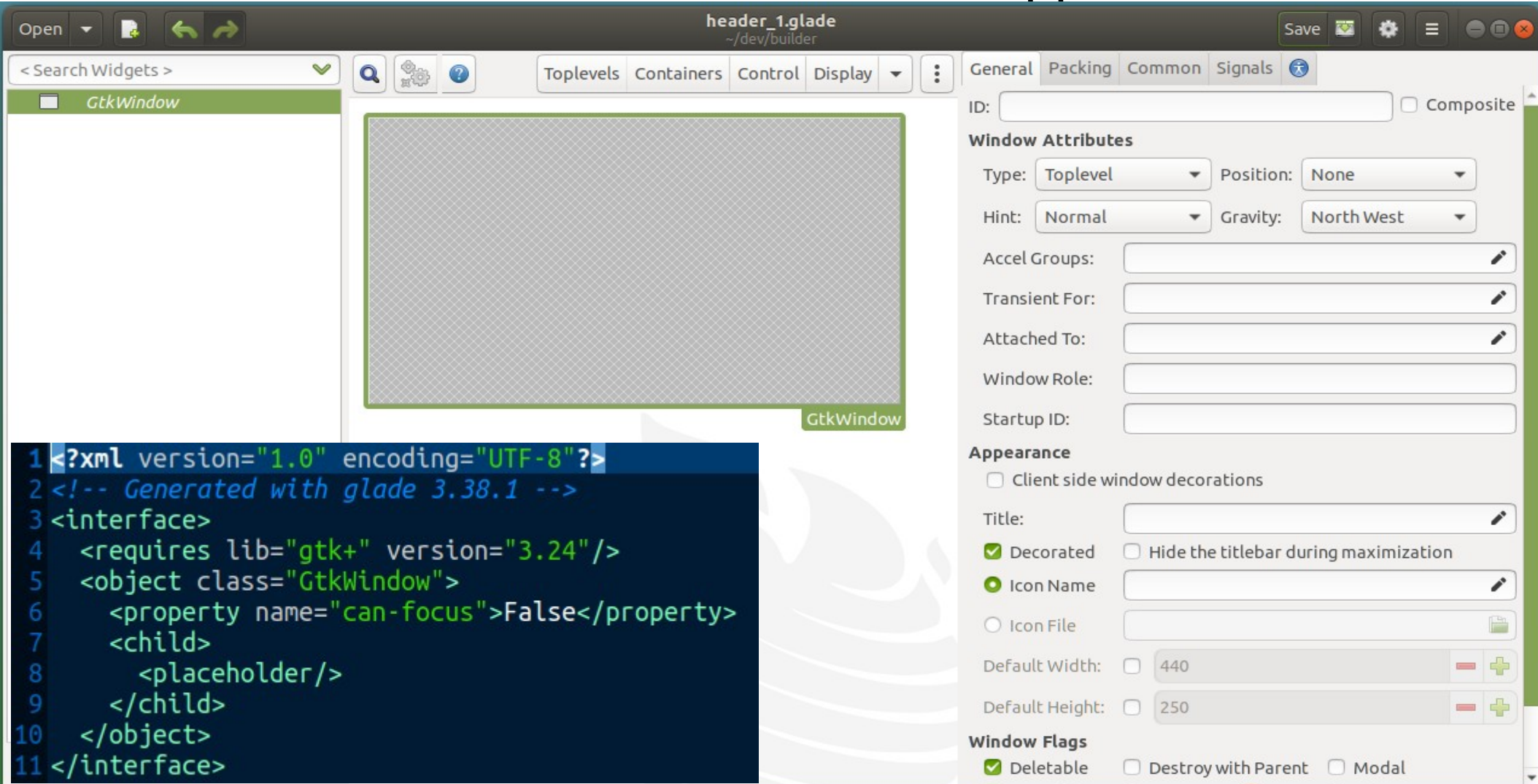
Gtk.Header ~ Contents of UI XML file: header_1.glade

```
*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. XML is appended



The screenshot displays 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 central canvas shows a large rectangular widget 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 'Window Attributes' and 'Appearance' sections. The 'Window Attributes' section includes fields for 'Type' (set to 'Toplevel'), 'Position' (set to 'None'), 'Hint' (set to 'Normal'), and 'Gravity' (set to 'North West'). The 'Appearance' section includes a checkbox for 'Client side window decorations' (unchecked), a 'Title' field, a checked checkbox for 'Decorated', a 'Hide the titlebar during maximization' checkbox (unchecked), an 'Icon Name' field, an 'Icon File' field, 'Default Width' (set to 440), and 'Default Height' (set to 250). The 'Window Flags' section includes a checked checkbox for 'Deletable', and unchecked checkboxes for 'Destroy with Parent' and 'Modal'. A code editor window is overlaid in the bottom left corner, showing the XML code for the widget.

```
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 interface is divided into several panes:

- Left Pane (Search Widgets):** Shows a search bar with `< Search Widgets >` and a list of widgets. `window_1` is selected, showing it is a `GtkWindow`.
- Center Pane (Design View):** Displays a visual representation of the window. A yellow callout bubble with the text "Check" points to the window's border.
- Right Pane (Properties):** Contains various configuration options for the selected widget.
 - General Tab:** The `ID` field is set to `window_1`. A yellow callout bubble with the text "Provide an ID" points to this field.
 - Window Attributes:** Includes settings for Type (Toplevel), Position (None), Hint (Normal), and Gravity (North West).
 - Appearance:** The `Client side window decorations` checkbox is checked. A yellow callout bubble with the text "Check" points to this checkbox.
 - Window Flags:** Includes checkboxes for Deletable, Destroy with Parent, and Modal.
- Bottom Pane (Code View):** Displays the XML code for the window. A yellow callout bubble with the text "Provide an ID" points to the `id="window_1"` attribute in the `<object>` tag.

```
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 window named 'window_1'. The left pane shows the widget hierarchy with 'window_1' selected. The center pane shows a visual representation of the window. The right pane contains the configuration settings for 'window_1', with the 'General' tab active. The 'Window Attributes' section shows 'Type' set to 'Toplevel', 'Position' to 'None', 'Hint' to 'Normal', and 'Gravity' to 'North West'. The 'Appearance' section has 'Client side window decorations' checked. The 'Window Flags' section has 'Deletable' checked. The 'Default Width' is set to 440 and 'Default Height' is set to 250, both with checkboxes and adjustment buttons. Two yellow callout boxes with the text 'Make visible' point to the 'default-width' and 'default-height' properties in the XML code on the left.

```
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>
```

Make visible

Make visible

General Packing Common Signals

ID: window_1 ☐ Composite

Window Attributes

Type: Toplevel Position: None

Hint: Normal Gravity: North West

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

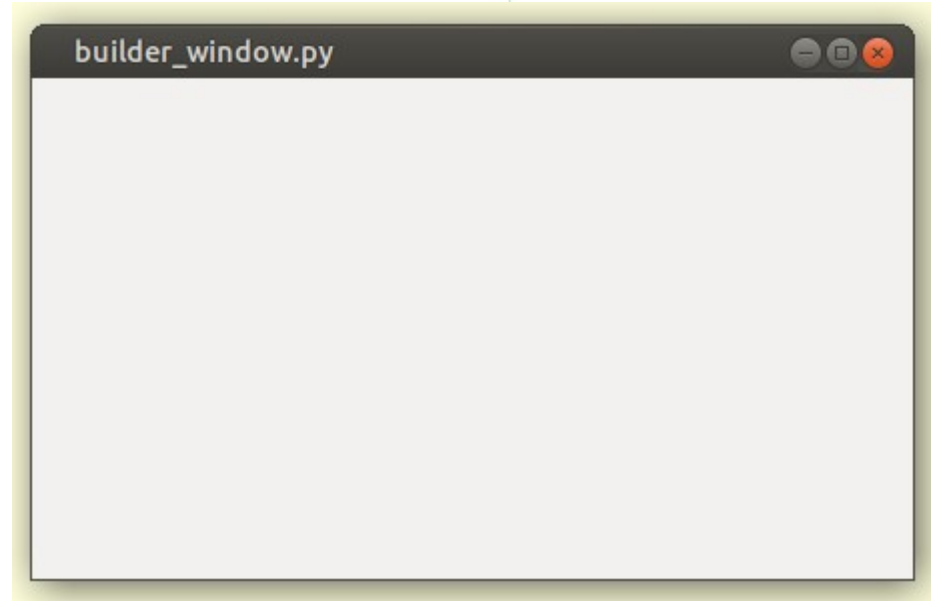
Default Height: ☒ 250

Window Flags

☒ Deletable ☐ Destroy with Parent ☐ Modal

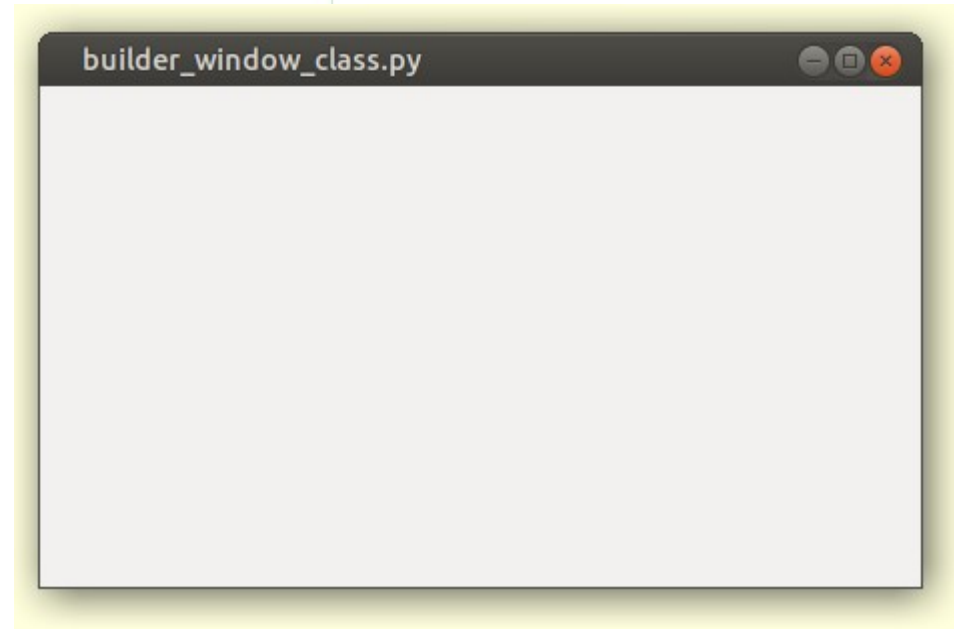
Using Gtk.Builder and the Glade xml data.

```
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()
```

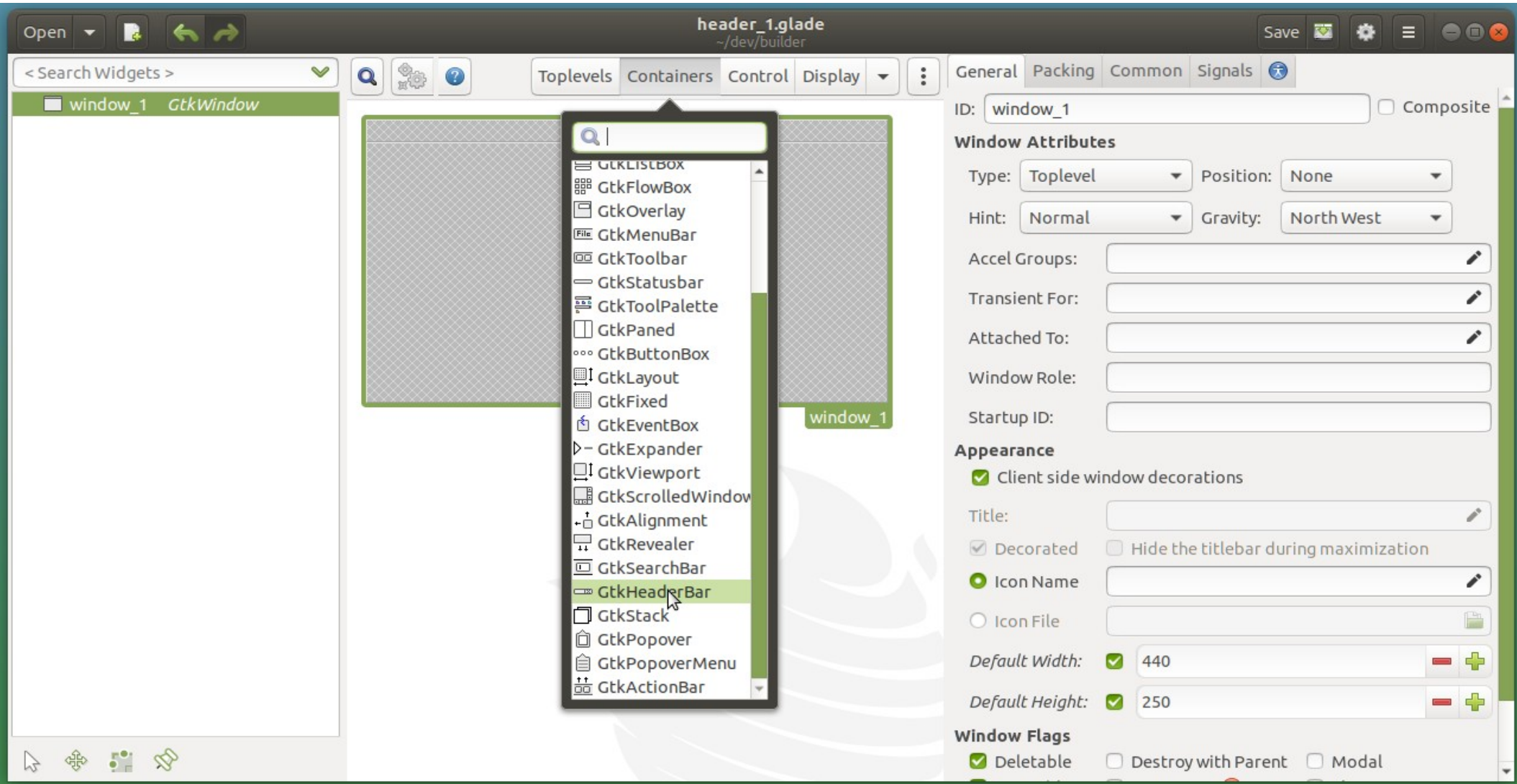


Place the code in a class.

```
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.

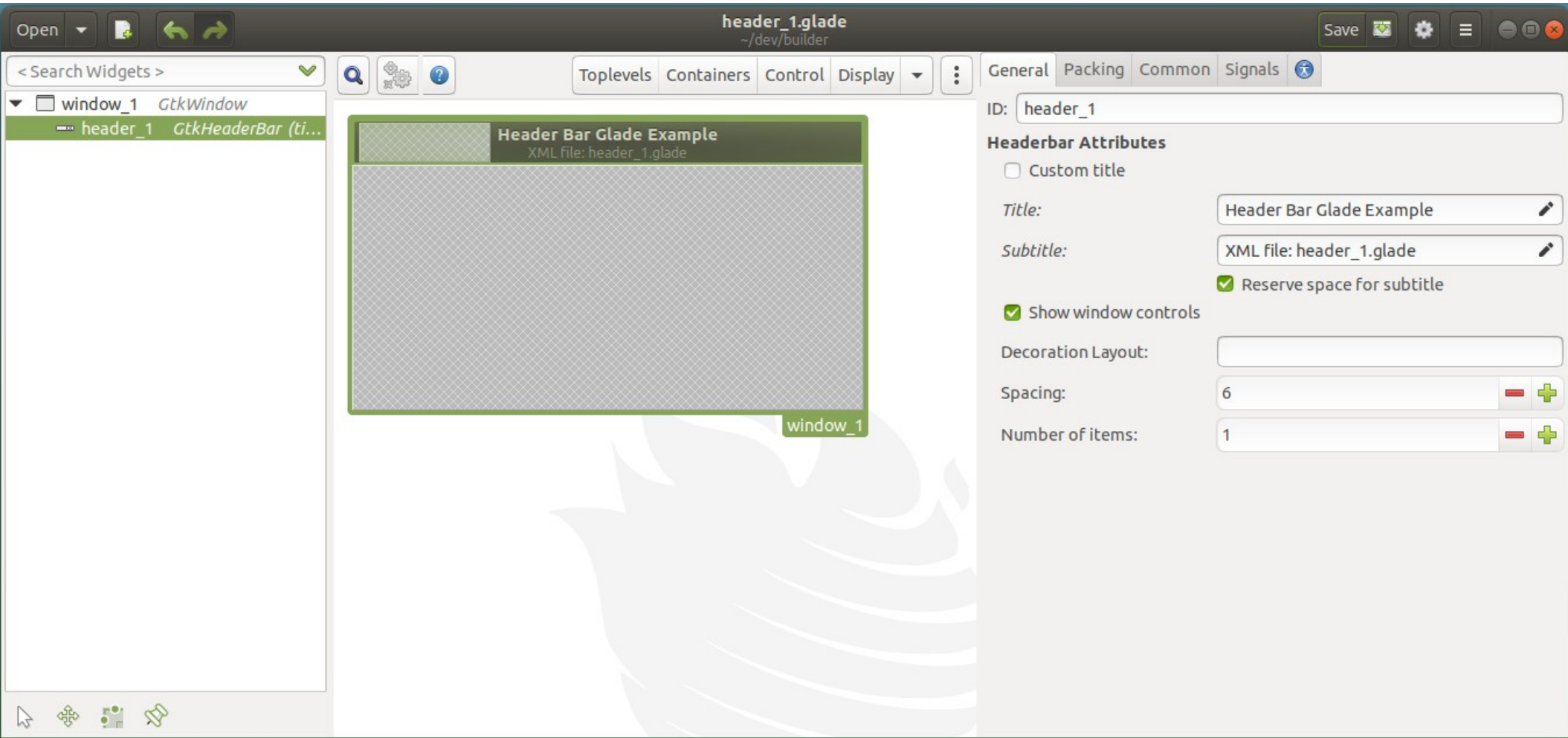


Place the code in a class.

The screenshot shows the Glade GUI designer interface. The title bar indicates the file is `header_1.glade` located at `~/dev/builder`. The interface is divided into several sections:

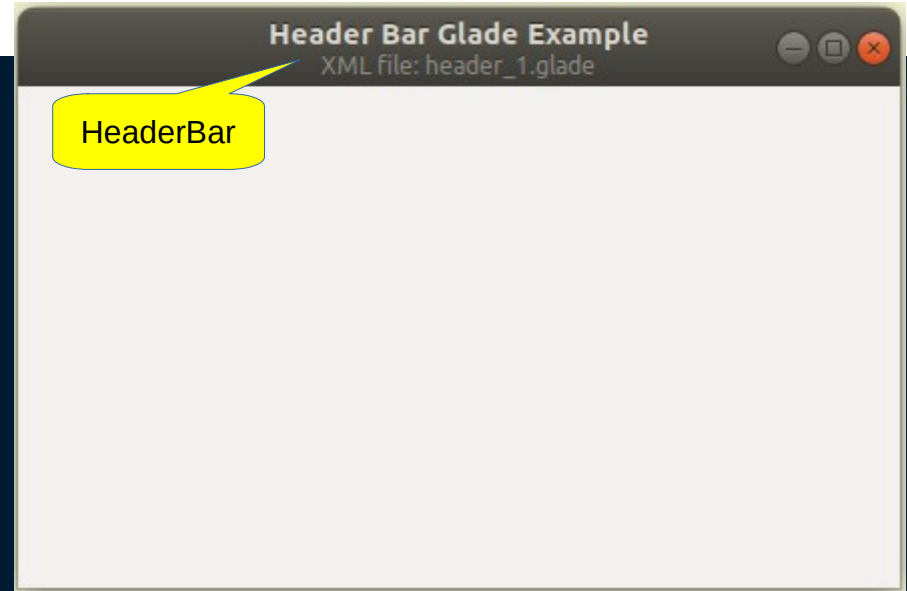
- Left Panel:** A search bar labeled "< Search Widgets >" and a list of widgets. The widget `window_1` of type `GtkWindow` is selected.
- Center Canvas:** A visual representation of the GUI. It shows a `GtkHeaderBar` widget being placed onto a canvas labeled `window_1`. A search bar and a list of widget types are overlaid on the canvas, with `GtkHeaderBar` highlighted.
- Right Panel:** The Properties panel for the selected widget (`window_1`). It contains several tabs: `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.



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.

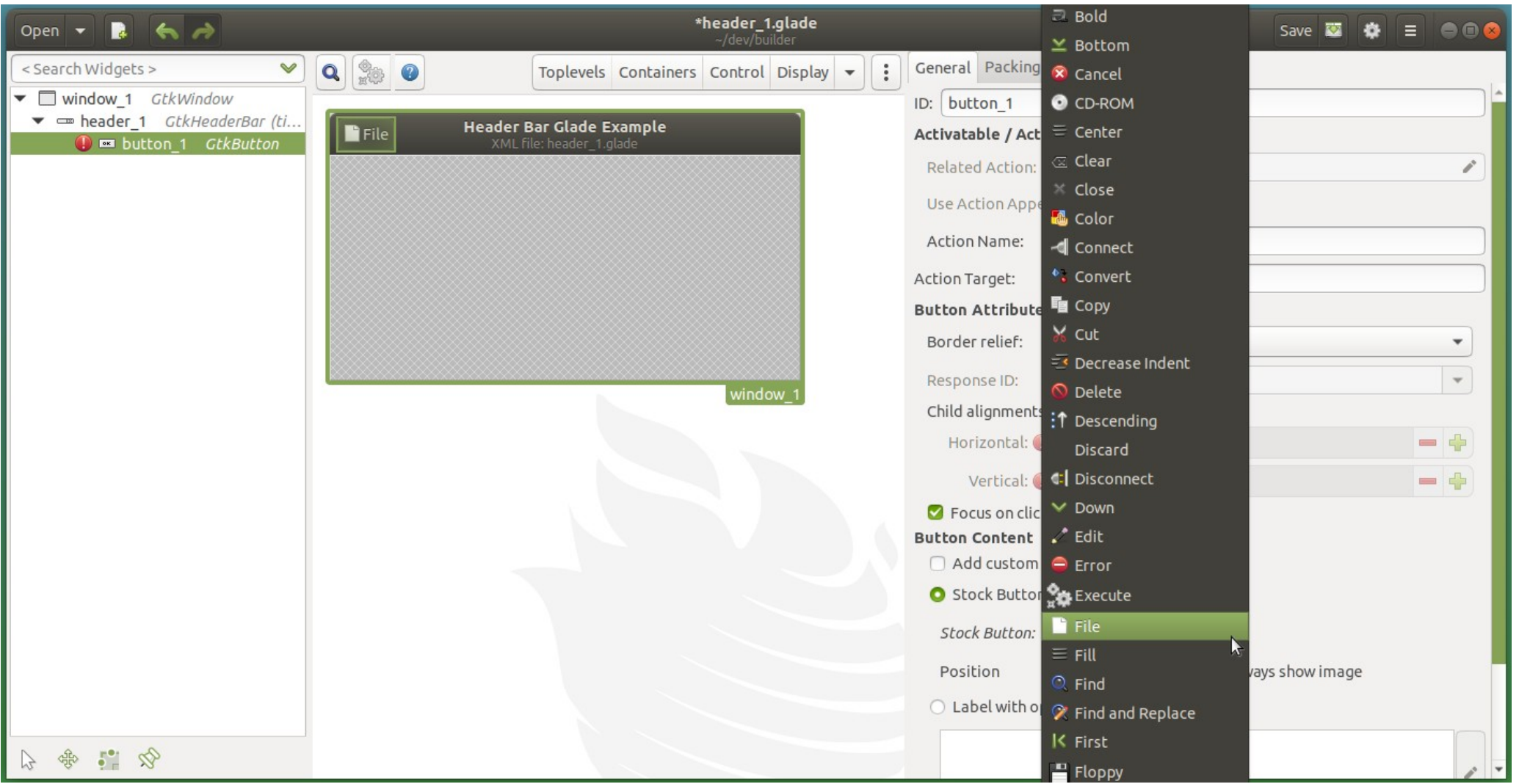
Left Pane (Widget Hierarchy):

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

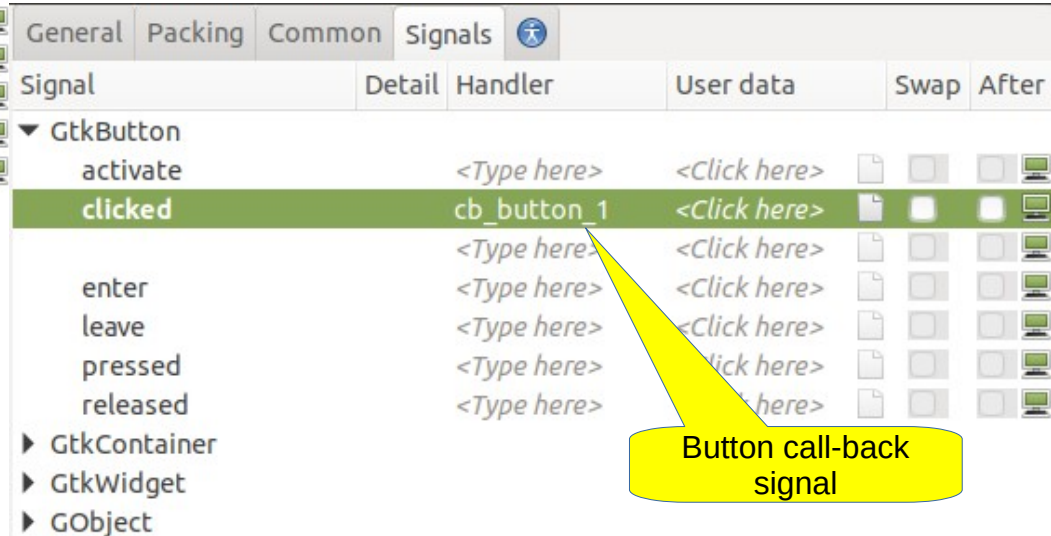
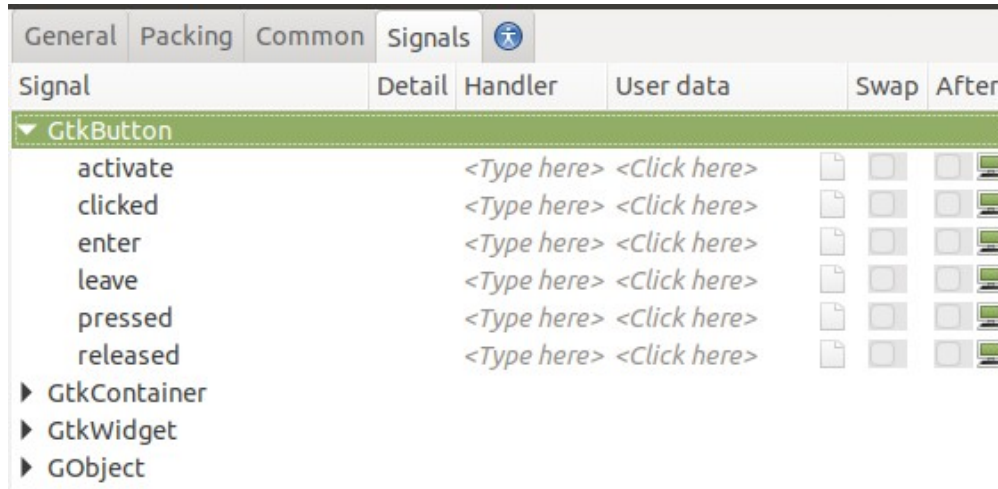
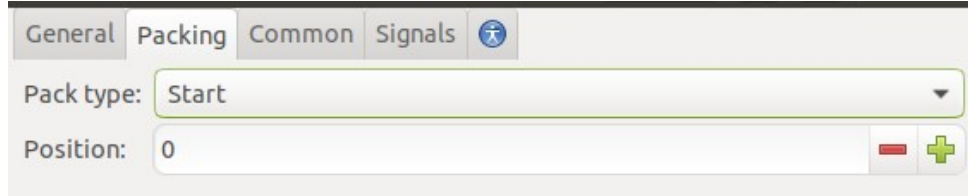
Right Pane (Properties):

- General** (selected tab)
- ID: `button_1`
- Activatable / Actionable**
 - Related Action: ☐ [Empty field]
 - Use Action Appearance: ☐ [Toggle]
 - Action Name: [Empty field]
 - Action Target: [Empty field]
- 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: [Dropdown menu]
 - 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>
```

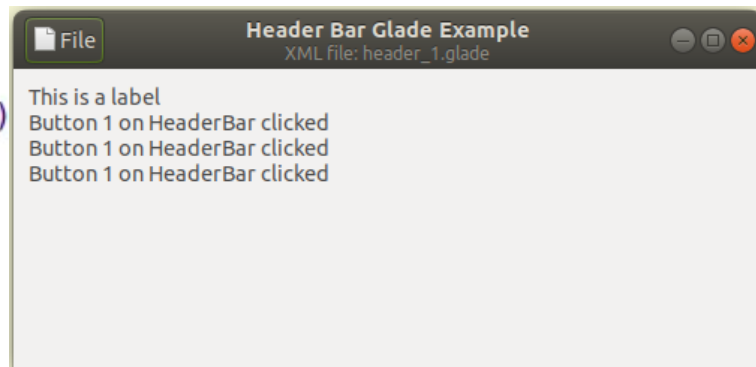
Button call-back
signal

Builder connects signals. Provide method for button_1 “file”.

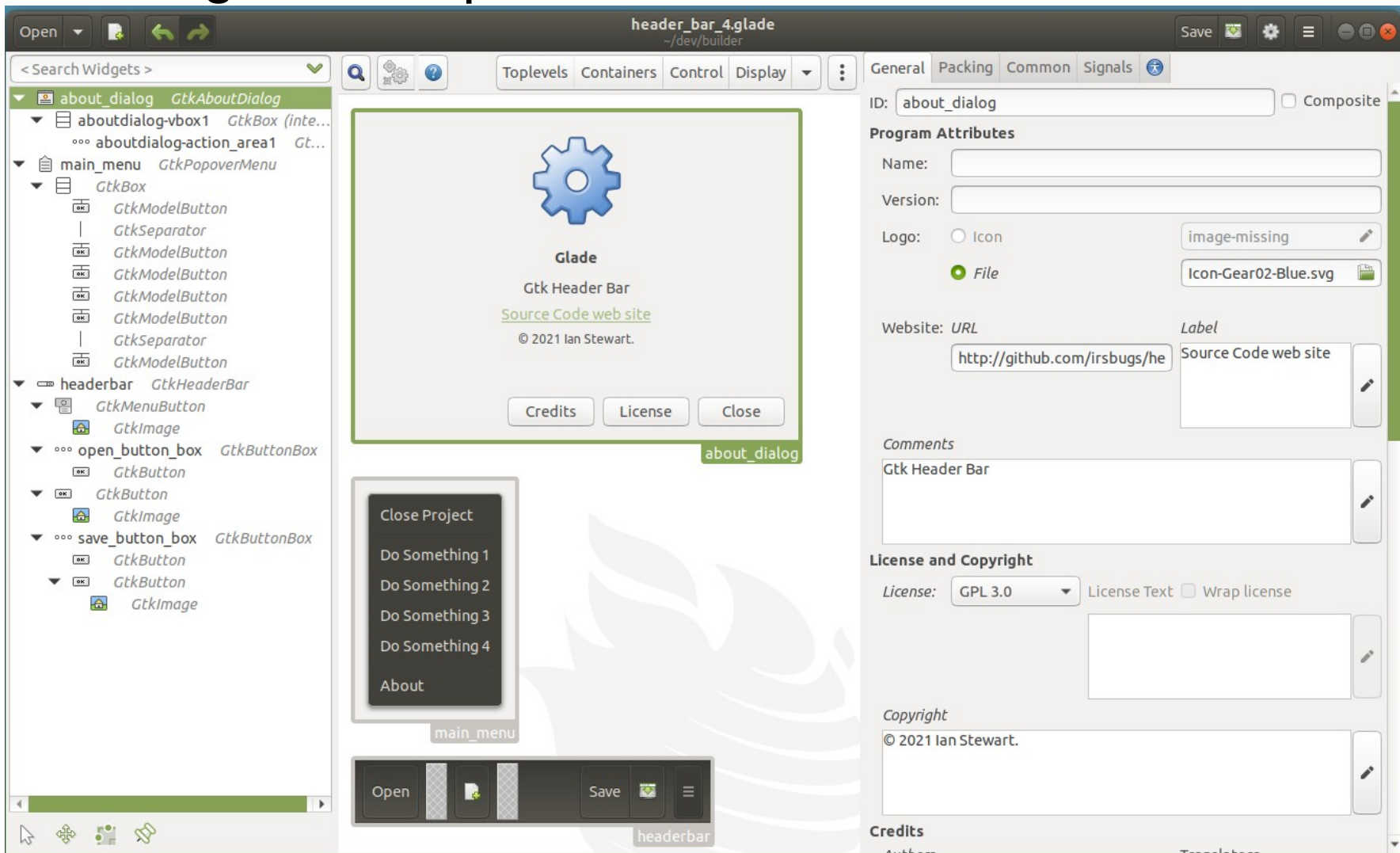
```
34 class ExampleWindow(Gtk.Window):
35     def __init__(self):
36         Gtk.Window.__init__(self, title="Menu Example")
37         self.set_default_size(500, 200)
38
39         # Use Builder to read embedded xml string defining HeaderBar
40         builder = Gtk.Builder()
41         builder.add_from_string(glade_xml)
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 programs...

conventional_window_with_header_bar.py

conventional_window_with_glade_components.py

double_precision.py

End.