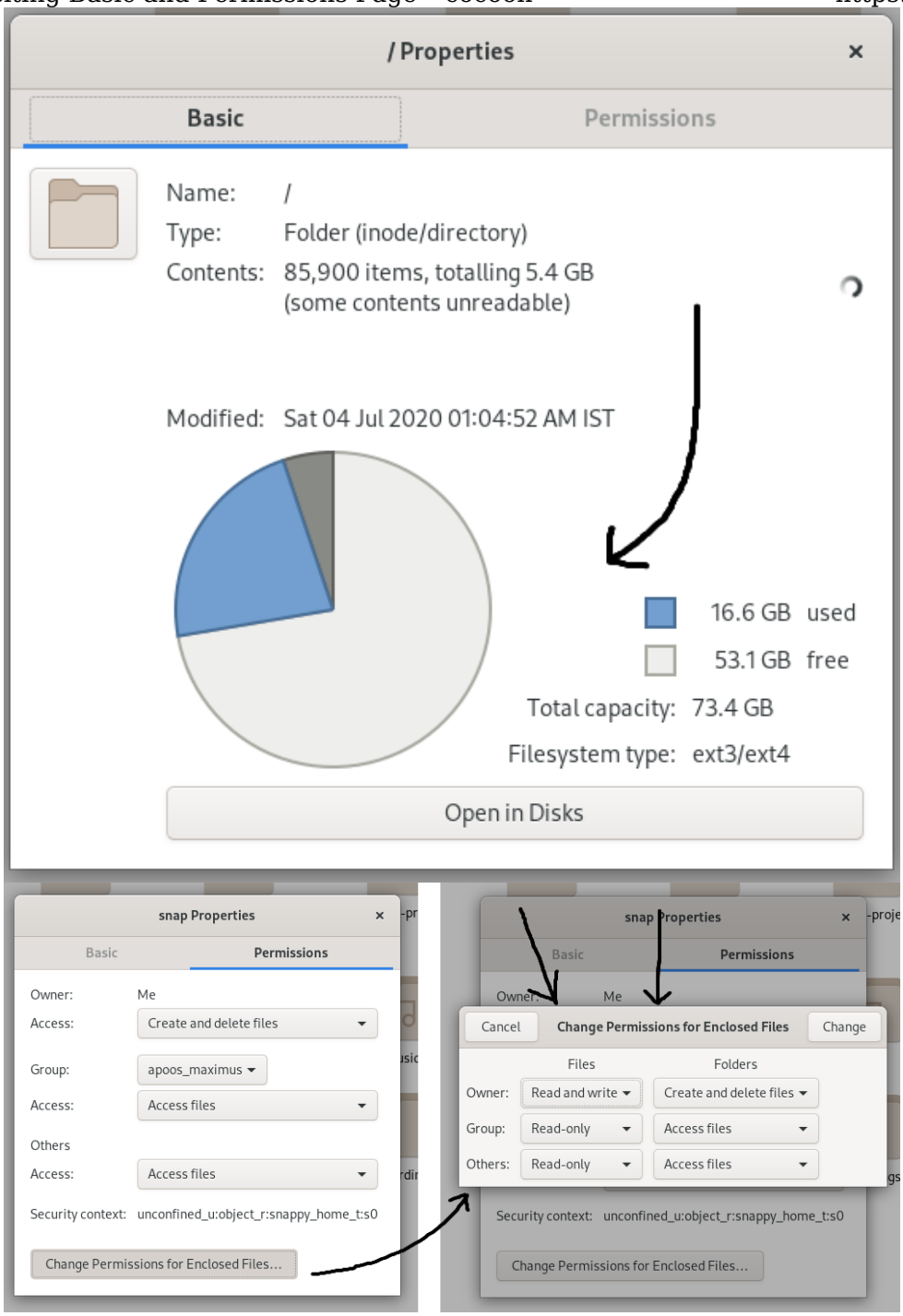
cocoon

let the thoughts fly

Revisiting Basic and Permissions Page

ON JULY 28, 2020JULY 28, 2020 / BY APOOSMAXIMUS / IN TECH /

Porting of Basic and Permissions pages, have been covered in the previous posts, but like the heading suggests there sure was something left. The candidates which remained to be ported were the volume usage widget featuring the pie-chart and the change permissions dialogue which can be used to change permissions of enclosed files in a folder.



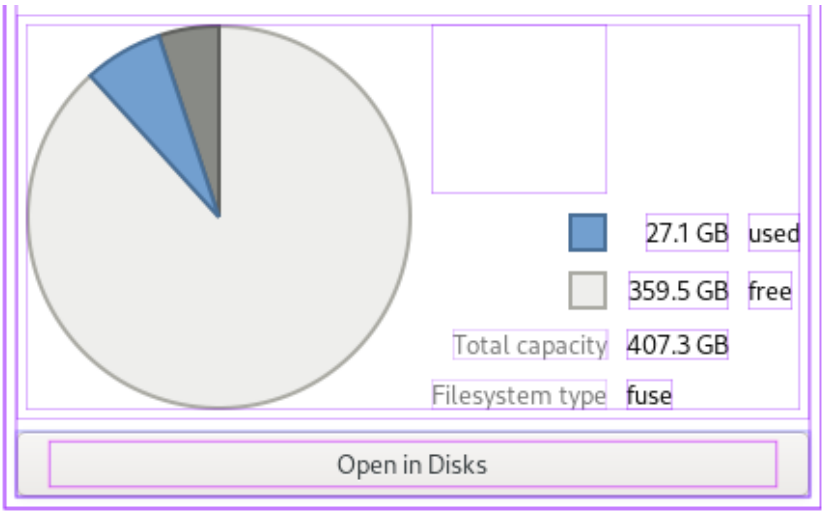
Widgets that remain to be ported

The Volume Widget

The volume widget in itself is a GtkGrid which packs GtkDrawingArea's and GtkLabels, The dimensions of the grid being `5x4` where the first `5x2` worth of room is occupied by the GtkDrawingArea and the rest is occupied by legend which conveys the size and free-space of the selected volume along with their representations in the pie chart. The colored-boxes used to indicate representational colors are also GtkDrawingArea. Styled accordingly with a CSS style class to obtain the required fill-color. Well that leaves us with the pie-chart ! Now, handling that is not as easy as

a pie, but not too hard either. The task of drawing the pie chart was simply left to the preexisting `signals` and `callbacks`

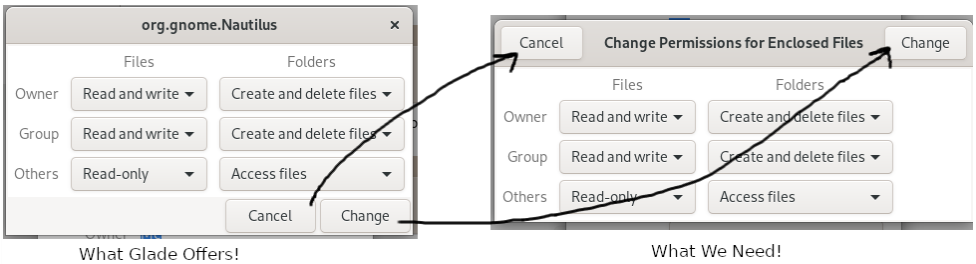
```
     g_signal_connect (window->pie_chart, "draw",
                       G_CALLBACK (paint_pie_chart),
window);
    g_signal_connect (window->used_color, "draw",
                       G_CALLBACK (paint_legend),
window);
    g_signal_connect (window->free_color, "draw",
                       G_CALLBACK (paint_legend),
window);
```



That wraps the volume_widget.

Change Permissions Dialog

This story of porting is old, if you have read the previous posts perhaps you can take a guess and ask me to fire-up glade, pick-up GtkDialog from Toplevel containers/ windows, throw-in a grid, araange labels and buttons and than all thats left is to obtain `GObject` references through the `GtkBuilder API` and wire them together with code. But this is when you realize Glade doesn't offer all variations of a Composite Widget like GtkDialog, lets see:



what glade offers V/s what we need

The reason for this being. Glade doesn't suppoty use-header-bar flag supported by GtkDialog as a result of which, we need to compose the XML for dialog's UI with our own hands, to enable the usage of use-header-bar flag so that buttons could be located there.

Your Dialog Your XML

This is the blog that helped us get our Handwritten XML to produce the outcome-we desired: How do I Dialogs (https://wiki.gnome.org/HowDoI/Dialogs)

```
<?xml version="1.0" encoding="UTF-8"?>
<interface>
  <requires lib="gtk+" version="3.22"/>
  <object class="GtkDialog" id="change_permissio
    <property name="title" translatable="yes">Cha
    <property name="modal">True</property>
    <property name="destroy_with_parent">True</p
    <property name="type_hint">dialog</property>
    <property name="use-header-bar">1</property>
    <child type="action">
      <object class="GtkButton" id="cancel">
        <property name="visible">True</property>
        <property name="label">Cancel</property>
      </object>
    </child>
    <child type="action">
      <object class="GtkButton" id="change">
        <property name="visible">True</property>
        <property name="label">Change</property>
      </object>
    </child>
<child>
 <!-- GtkGrid goes here -->
</child>
    <action-widgets>
      <action-widget response="cancel">cancel</ac
      <action-widget response="ok">change</actio
    </action-widgets>
  </object>
</interface>
```

So this is what we ended up with ! Now this could be used like any other .ui XML file and the old story of porting UI could continue as usual !