

Previous page:

BoxSizerFromTheGroundUp/ControlsResizing

BoxSizerFromTheGroundUp

BoxSizerFromTheGroundUp/ControlsPadding

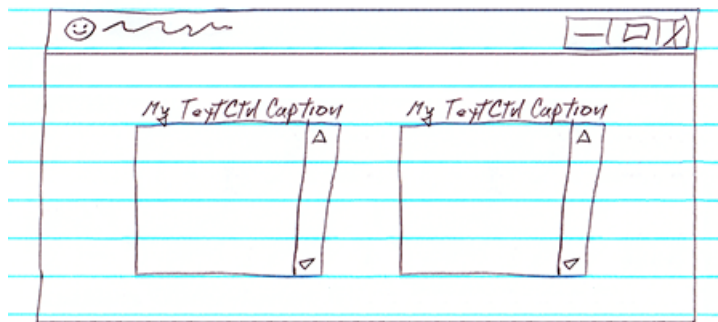
BoxSizerFromTheGroundUp/ControlsResizing

BoxSizerFromTheGroundUp/NestedBoxSizers

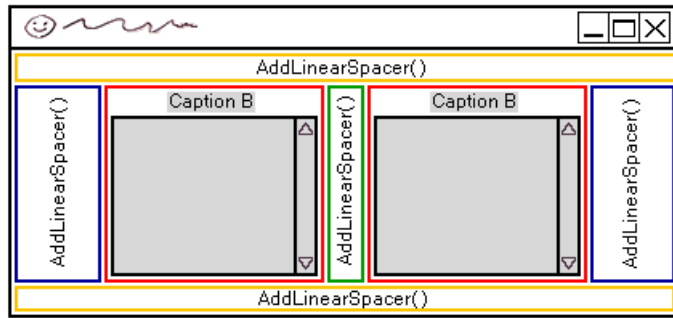
BoxSizerFromTheGroundUp/DivideAndConquer

Multiple BoxSizers: Nesting

Most layouts with even just a few controls require multiple sizers. Consider the following simple layout. It's not possible to group the four controls with a single sizer because they are arranged in a two dimensional pattern. **BoxSizers** can only stack along one axis. This 2-D pattern is "regular" in that a **StaticText** and a **TextCtrl** can be duplicated. This is easily done by subclassing a container control and laying out a **StaticText-over-TextCtrl**. The only "trick" is to create all unique control instantiations.



While the necessary thought process can be done entirely "in your head", it's much more informative to annotate the above sketch to determine exactly all control groups and spacers.



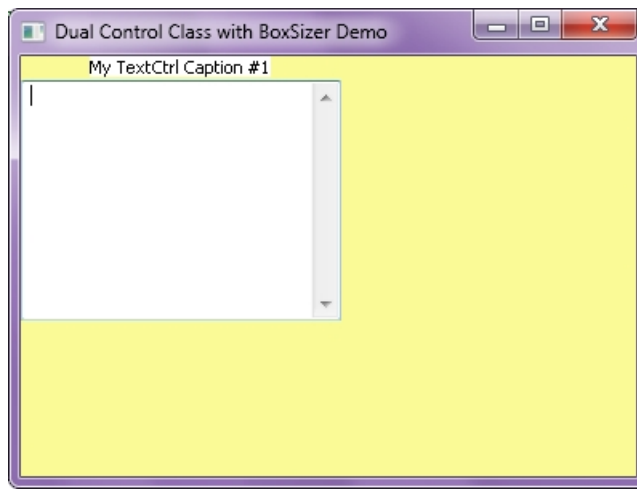
Your own hand sketches don't need to be nearly as fancy as this one, but I want to make sure you understand what I'm trying to do.

Now we're ready to start coding. The most important task is to create a **StaticText-over-TextCtrl** class that can be instantiated twice

```

1 class TxtCtrlPanel( wx.Panel ) :
2
3     def __init__( self, parent, caption_txt ) :
4
5         wx.Panel.__init__( self, parent=parent, id=-1 )
6
7         #-----
8
9         self.caption_stTxt = wx.StaticText( self, -1,
caption_txt )
10        self.caption_stTxt.BackgroundColor = ( 'wHite' )
11
12        self.list_txtCtrl = wx.TextCtrl( self, -1,
style=wx.TE_MULTILINE, size=(200, 150) )
13
14        #-----
15
16        panel_vertSizer = wx.BoxSizer( wx.VERTICAL )
17
18        panel_vertSizer.Add( self.caption_stTxt, proportion=0,
flag=wx.CENTER )
19
20        als.AddLinearSpacer( panel_vertSizer, 2 )
21
22        panel_vertSizer.Add( self.list_txtCtrl, proportion=0,
flag=wx.ALIGN_CENTER )
23
24        self.SetSizer( panel_vertSizer ) # Invoke the sizer.
25        self.Fit() # Make "self" (the panel) shrink to the
minimum size required by the controls.
26
27        #end __init__
28
29        def GetControls( self ) :
30
31            return self.caption_stTxt, self.list_txtCtrl
32
33 #end TxtCtrlPanel class

```



Note that both controls are Add()ed with the flag=wx.CENTER parameter. This ensures that the narrower control is centered by the sizer. It's can't be known ahead of time which of the two will be narrower, even though the control that is wider has no need for this parameter.

The demo app frame is very simple. Line #16 retrieves the instantiated controls for later use when the **TextCtrl** is populated. Even the **StaticText** can be changed if there is a need. The "static" part of the name "**StaticText**" means that this control is not for use as an interactive control such as a **TextCtrl**, a **Button** or a **CheckBox**.

```

1 class MainFrame( wx.Frame ) :
2     """ A very basic single BoxSizer demo """
3
4     def __init__( self, parent ) :
5
6         wx.Frame.__init__( self, parent, -1, title='Simple Single
Sizer Demo',
7                             size=(400, 300), pos=(100, 0),
8                             style=wx.DEFAULT_FRAME_STYLE )
9
10        self.SetBackgroundColour( (215, 150, 250) ) # pale
yellow
11        sacrificial_frmCtrl = wx.Panel( self ).Hide() ** What is
this? Please clarify. **
12
13        #-----
14
15        txtCtl_panel_1 = TxtCtrlPanel( self, 'My TextCtrl Caption
#1' )
16        stTxt_1, txtCtl_1 = txtCtl_panel_1.GetControls()
17
18        #-----
19
20        allCtrls_vertSizer = wx.BoxSizer( wx.HORIZONTAL )
21
22        self.SetSizer( allCtrls_vertSizer )
23        self.Layout() # self.Fit() #
always use one or the other

```

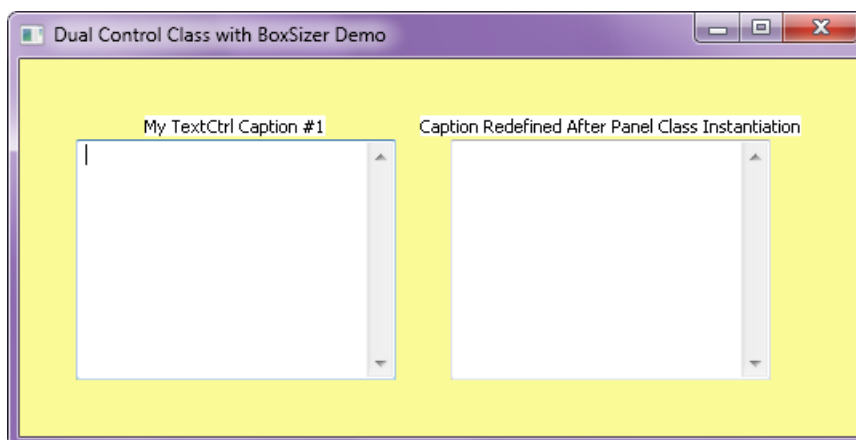
```

24
25     #end __init__
26
27 #end MainFrame class
28
29 #=====
30
31 if __name__ == '__main__' :
32
33     app = wx.PySimpleApp( redirect=False )
34     appFrame = MainFrame( None ).Show()
35     app.MainLoop()
36
37 #end if

```

DUAL_CONTROL_CLASS_1.PY

The demo app can now be modified to add a second **TxtCtrlPanel** instantiation now that its class is proven to be correct . To align the two panels horizontally there needs to be a horizontal sizer added in the Frame **class object** to hold them. Three spacers will also be added to separate the panels from each other and the edges of the page. In addition, the entire page needs to be "sized" (positioned) by an overall vertical sizer. It's all very straight-forward.



Note that the second panel's caption was changed after the panel was already created. Also note that the width of the **StaticText** is wider than its **TextCtrl**. This demonstrates why the parameter **wx.CENTER** needs to be in every **.Add()** statement for each control in the sizer. Also be aware of the three other synonyms for **wx.CENTER** !

```

1 class MainFrame( wx.Frame ) :
2     """ A 2-control class with a BoxSizer demo """
3
4     def __init__( self, parent ) :
5
6         wx.Frame.__init__( self, parent, -1, title='Dual Control
Class with BoxSizer Demo',

```

```

7             size=(500, 300), pos=(100, 0),
8             style=wx.DEFAULT_FRAME_STYLE )
9
10          self.SetBackgroundColour( (250, 250, 150) )      # pale
yellow
11          sacrificial_frmCtrl = wx.Panel( self ).Hide()
12
13          #-----
14
15          # Create all the page's controls either individually or
using classes.
16
17          ctrlsPanel_1 = TxtCtrlPanel( self, 'My TextCtrl Caption
#1' )
18          stTxt_1, txtCtl_1 = ctrlsPanel_1.GetControls() # Retrieve
the control objects.
19
20          ctrlsPanel_2 = TxtCtrlPanel( self, 'My TextCtrl Caption
#2' )
21          stTxt_2, txtCtl_2 = ctrlsPanel_2.GetControls()
22
23          # Let's change the original caption "after the fact".
24          stTxt_2.Label = 'Caption Redefined After Panel Class
Instantiation'
25          ctrlsPanel_2.LayoutAgain()
26
27          #-----
28
29          panelCtrls_horzSizer = wx.BoxSizer( wx.HORIZONTAL )
30
31          als.AddLinearSpacer( panelCtrls_horzSizer, 35 )
32          panelCtrls_horzSizer.Add( ctrlsPanel_1 )
33          als.AddLinearSpacer( panelCtrls_horzSizer, 15 )
34          panelCtrls_horzSizer.Add( ctrlsPanel_2 )
35          als.AddLinearSpacer( panelCtrls_horzSizer, 35 )
36
37          page_vertSizer = wx.BoxSizer( wx.VERTICAL )
38
39          als.AddLinearSpacer( page_vertSizer, 35 )
40          page_vertSizer.Add( panelCtrls_horzSizer )
41          als.AddLinearSpacer( page_vertSizer, 35 )
42
43          # "There can be only one.", Duncan MacLeod from the clan
MacLeod.
44          # That is, a page can have exactly 1 or none top level
sizer.
45          # Sizer nesting can effectively have infinite levels.
46          self.SetSizer( page_vertSizer )
47          self.Fit()      # or self.Layout() to keep the pages
programatically-set size.
48
49          #end __init__
50
51          #end MainFrame class
52
53
=====
=====
54
55 if __name__ == '__main__' :
56
57     app = wx.PySimpleApp( redirect=False )
58     appFrame = MainFrame( None ).Show()

```

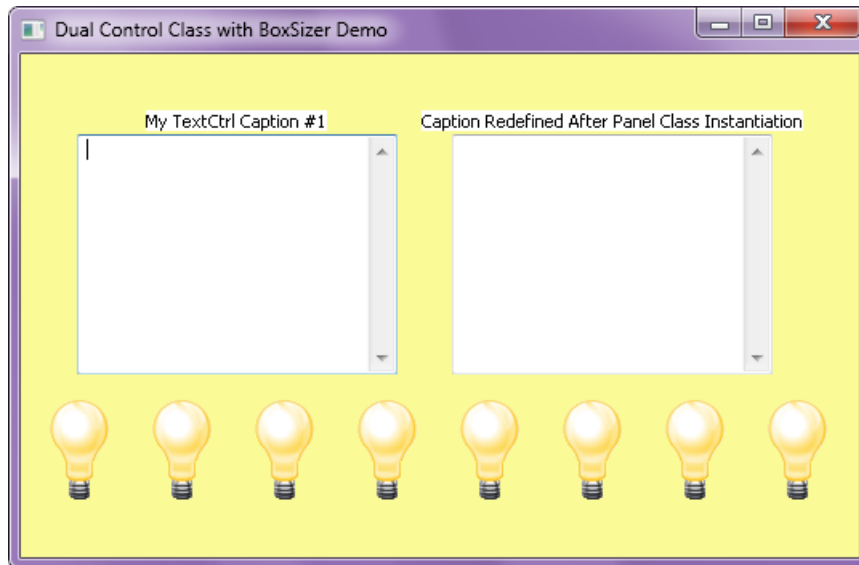
```

59     app.MainLoop( )
60
61 #end if

```

- DUAL_CONTROL_CLASS_2.PY
- AddLinearSpacer.py

Just for fun, I'll throw in a graphic image.



```

1         #-----
2
3         lightbulb_image = wx.Image( 'LIGHTBULB.64x64.PNG',
wx.BITMAP_TYPE_ANY )
4         lightbulb_bmap = lightbulb_image.ConvertToBitmap()
5
6         lightbulb_horzSizer = wx.BoxSizer( wx.HORIZONTAL )
7         for idx in xrange( 8 ) :
8             aLightbulb_stBmap = wx.StaticBitmap( self, -1,
lightbulb_bmap )
9             lightbulb_horzSizer.Add( aLightbulb_stBmap,
flag=wx.CENTER )
10
11         page_vertSizer.Add( lightbulb_horzSizer,
flag=wx.ALIGN_CENTRE ) # note synonym
12
13         #-----

```

- DUAL_CONTROL_CLASS_3.PNG
- LIGHTBULB.64x64.PNG

Adding controls to the page sizer is very easy when it's planned out, especially

when control classes for multiple grouped controls are used. I think you can see by now that **laying out pages can very quickly become very, very complicated**. This brings us up to the next topic :

Next page:

BoxSizerFromTheGroundUp/DivideAndConquer

BoxSizerFromTheGroundUp

BoxSizerFromTheGroundUp/ControlsPadding

BoxSizerFromTheGroundUp/ControlsResizing

BoxSizerFromTheGroundUp/NestedBoxSizers

BoxSizerFromTheGroundUp/DivideAndConquer

BoxSizerFromTheGroundUp/NestedBoxSizers (last edited 2010-09-29 16:36:10 by WinCrazy)