Previous page:
BoxSizerFromTheGroundUp/ControlsPadding

BoxSizerFromTheGroundUp

BoxSizerFromTheGroundUp/ControlsPadding

BoxSizerFromTheGroundUp/ControlsResizing

BoxSizerFromTheGroundUp/NestedBoxSizers

BoxSizerFromTheGroundUp/DivideAndConquer

Resizing Controls

~~The~~ BoxSizer ~~can be used to expand a control "to fill the space allotted to the item". See~~ http://www.wxpython.org/docs/api/wx.Sizer-class.html~~. That's probably as glib as humanly possible. Exactly how much space will be "allotted to the item" ? For that matter, what the heck is an "item" ?! Well, what the~~ WxWidget~~'s documentors didn't bother to write is that the control will be expanded to take up all the space in the container client area excluding the space taken up by fixed-sized controls and fixed-sized spacers. If there are multiple controls and/or spacers that will be **wx.EXPAND**ed then the free space will divided among them. Oh yes: an "item" is simply a control, not a list item, a dictionary item, etc. etc.~~ ** Rewrite - less sarcasm please. Better yet, omit; I'm not sure it's savable. **

To demonstrate BoxSizer control expansion I want to show a frame with just two **wx.Panel**s side-by-side horizontally. The panel colors will be different to make it obvious which panel is which. To a horizontal BoxSizer the two panels will be **.Add()**ed. First, just create the **Frame**, the panels and the sizer. Keep in mind that **wx.Frame**'s have a unique, usually annoying "feature" in that ~~**the first control of any kind to be instantiated in the Frame**~~ ~~***will be automatically expanded along both axes to fill the Frame's client area***~~ the first control of any kind to be instantiated in the Frame *will be automatically expanded along both axes to fill the Frame's client area* . ** Illegal use of bold, underscore and italic at the same time. Bold is already taken. I picked italic.** This "feature" is usually unwelcome, ~~by~~ but wxPython is stuck with it.

```
1 #!/usr/bin/python
```

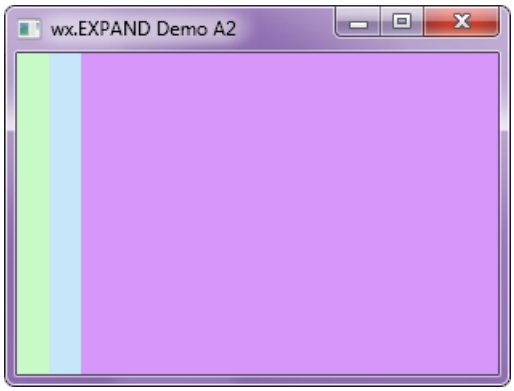
```
   2
   3 # wxEXPAND_DEMO_A_1.PY
   4
   5 import wx
   6
   7
#-----------------------------------------------------------------------
--------
   8
   9 class AppFrame( wx.Frame ) :
  10
  11     def __init__( self ) :
  12
  13         wx.Frame.__init__( self, parent=None, id=-1,
title='wx.EXPAND Demo A1' )
  14
  15         self.Position = (200, 0)
  16         self.ClientSize = (300, 200)        # Set client area.
Frame size is unimportant.
  17
  18         # The first control instantiated in a Frame automatically
expands
  19         #   to the size Frame.ClientSize.  This is unique to
Frames and wxPython.
  20         frame_panel = wx.Panel( self )
  21         # If any portion of this panel is seen this color will
make it stand out.
  22         frame_panel.BackgroundColour = ( 'pUrPLe' )
  23
  24         #-----
  25
  26         left_panel  = wx.Panel( frame_panel )
  27         left_panel.BackgroundColour  = (200, 250, 200)       #
green
  28
  29         right_panel = wx.Panel( frame_panel )
  30         right_panel.BackgroundColour = (200, 230, 250)      # blue
  31
  32         frame_horzSizer = wx.BoxSizer( wx.HORIZONTAL )
  33         frame_horzSizer.Add( left_panel  )      # , flag=wx.EXPAND
)
  34         frame_horzSizer.Add( right_panel )     # ,
flag=wx.EXPAND )
  35         frame_panel.SetSizer( frame_horzSizer )
  36
  37     #end __init__ def
  38
  39     #--------------
  40
  41     def ExitApp( self, event ) :
  42         self.Close()
  43
  44 #end AppFrame class
  45
  46
#=======================================================================
========
  47
  48 MyApp = wx.App( redirect=False )
  49 AppFrame().Show()
  50 MyApp.MainLoop()
```

No size or position parameters are given because it's the job of the sizer to set these.The flag=wx.EXPAND parameter has intentionally been left out to show what happens.

- 

This is definitely not what is wanted. But, the two panels are there and the right-side blue one is not on top of the left-side green. Most controls do not have a "best minimum" size, unlike **wx.Button**s and **wx.StaticText**s. On **MS Windows** platforms a control such as a **wx.Panel** which has not had its size defined will be reduced to a 20 pixel square. That's much better for debugging purposes than it being reduced to a 0 pixel square !

Now let the **wx.EXPAND** parameters be put in. Remember that **wx.EXPAND** affects only the minor axis and the sizer is w**x.HORIZONTAL**.

- 

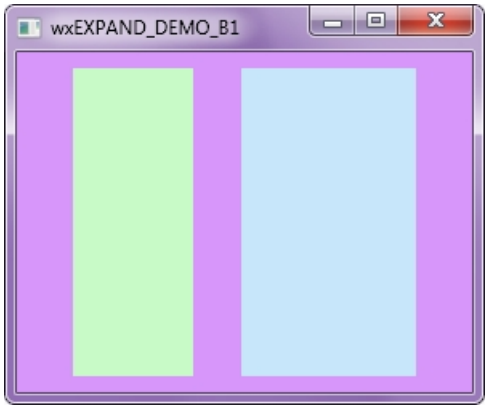
And there we have it ! Well, um... not quite. The panels did get **wx.EXPAND**ed along the sizer's minor axis just as they should have. But, what about expansion along the major axis ? This is the function of the **proportion=** argument that we have already seen. Since the panels are supposed to end up the same size then the integer values for **proportion=** need to be the same positive value. But, *which* integer value should be chosen ? **Any value >= 1 will do just fine.** However, in an effort to promote clarity the value **1** should be used so that anyone, including yourself, who reads the code at a later time will be able to immediately understand what is the intention of the code. ** I don't get how the value 1 tells future generations the intention of the code. Please clarify. **

- 

That's better ! Try resizing the frame horizontally to see how the sizer constantly maintains the panels' width ratio 1:1.

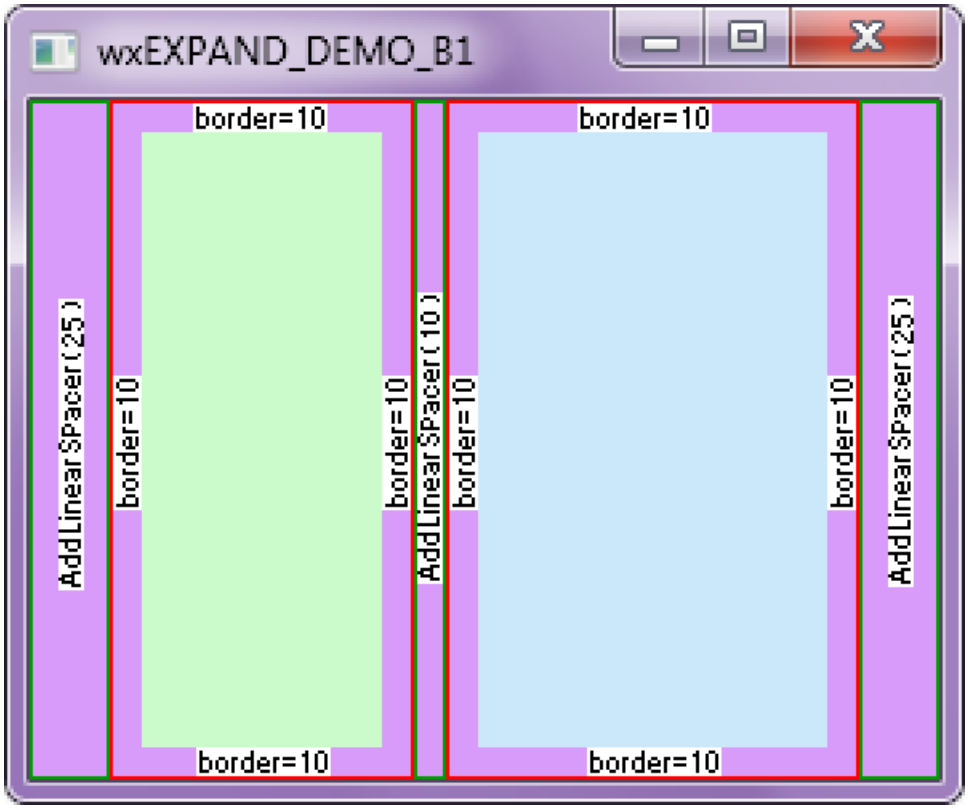
- wxEXPAND_DEMO_A_3.PY

It's time to apply everything we've learned so far to the single-sizer/two-panel frame above. Borders will be put around the panels as well as inserting Linear spacers to achieve a pleasant, though arbitrary appearance. No magic is involved.

- 

- wxEXPAND_DEMO_B_1.PY
- AddLinearSpacer.py

You can match up my annotations added in the following screen shot to the actual demo code.Shown 2X actual size.



Of course, many page layouts need to be more complicated than this. They can be very, very complicated when it makes sense to make an intricate display. This requires using multiple sizers.

Next page: BoxSizerFromTheGroundUp/NestedBoxSizers

BoxSizerFromTheGroundUp

BoxSizerFromTheGroundUp/ControlsPadding

BoxSizerFromTheGroundUp/ControlsResizing

BoxSizerFromTheGroundUp/NestedBoxSizers

BoxSizerFromTheGroundUp/DivideAndConquer

BoxSizerFromTheGroundUp/ControlsResizing (last edited 2010-09-29 16:36:42 by WinCrazy)