**Previous page:**
# BoxSizerFromTheGroundUp/NestedBoxSizers

# BoxSizerFromTheGroundUp

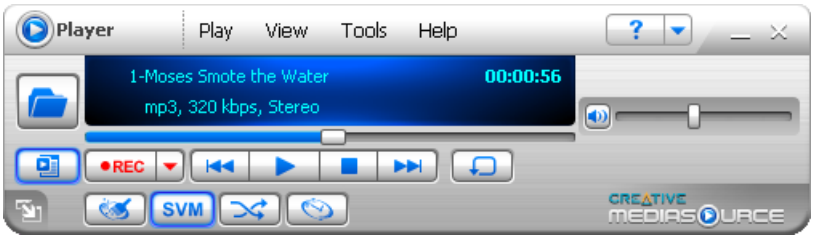
# BoxSizerFromTheGroundUp/ControlsPadding

# BoxSizerFromTheGroundUp/ControlsResizing

# BoxSizerFromTheGroundUp/NestedBoxSizers

BoxSizerFromTheGroundUp/DivideAndConquer

I think we have a good handle on the all the basics needed for nesting sizers with multiple controls. Let's get complicated now. Here's a screen shot of a music file player for MSW from Creative Labs that I received as a freebie when I bought a $25 (USD, shipping included) Creative Audigy II SoundBlaster PCI audio card. The task is: How would you recreate the following control using wxPython ? For the moment, forget about the gradients, shading and such. Approximate facsimiles can easily be added once the page has been laid out properly.

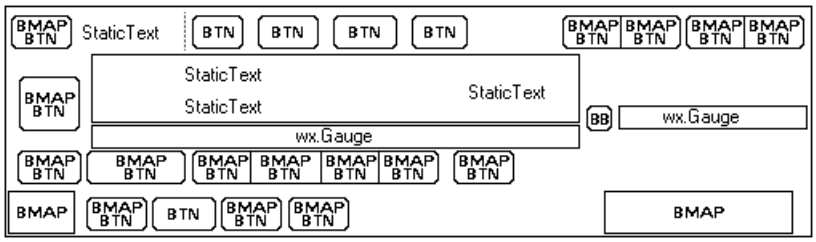
- 


This may seem to be a rather daunting project. Here's where a very fundamental and useful engineering principle known as Divide and Conquer can be used effectively. Simply stated, Divide and Conquer means to break up a large task ~~that is best not to be solved all at once~~ into more manageable sub-tasks. Implement and test each sub-task on its own to make sure it works the way you want it. Then, combine together the lower-level tasks into larger and larger intermediate-sized groups, all the while testing as you go along. Sooner or later everything will have been combined into a single top-level solution. At this point you are done-!

Sounds simple, doesn't it-? While you had been Dividing and Conquering you also had been organizing the project in a hierarchical fashion. This not only makes these projects easier to do, but it also makes it far easier to understand, especially by everyone else. I guess a corollary to "Divide and Conquer" is "A Large Project Is Never Finished". It needs to maintained, not ~~even~~ to mention fixing outright bugs that were missed during the initial design. Other users will find new bugs and deficiencies that the designer never even dreamed would happen. That's just the way it goes on every large development project ~~unless you are a certified genius~~.

Since this task is graphical the best tools to analyze it are graphical, too, such as a tried-and-true pencil and paper. You could get fancy and use a graphical software

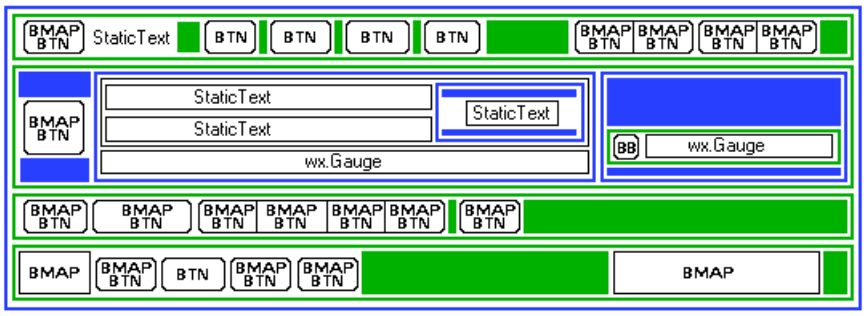
app such as MS Visio to make drawings very quickly and ~~easy~~ easily. Here's a sketch of the control panel reduced to just the actual controls without shading and other eye candy.

- 


This drawing seems to be a bit easier to break down into groups. By simple inspection a list of controls can be compiled:

- 2 StaticBitmaps
- 17 Bitmapped Buttons
- 5 Buttons
- 2 Gauges
- 4 StaticTexts

By observing the groupings of the controls we can choose which groups can be handled by specific sizers:

- 


The blue rectangles represent vertical BoxSizers, the green are horizontal. The solid rectangles are spacers of some sort. The analysis is complete. From here on just ordinary hard work is needed to implement the wxPython page. Then some real work needs to be done to actually create a properly working app using this GUI.

## BoxSizerFromTheGroundUp

## BoxSizerFromTheGroundUp/ControlsPadding

## BoxSizerFromTheGroundUp/ControlsResizing

## BoxSizerFromTheGroundUp/NestedBoxSizers

BoxSizerFromTheGroundUp/DivideAndConquer

BoxSizerFromTheGroundUp/DivideAndConquer (last edited 2010-09-29 16:35:32 by WinCrazy)