> I'd like to create a resizable widget app which looks
> kind of like this:
>
> +--+
> | LEFT_ALIGNED_LABEL   RIGHT_ALIGNED_LABEL |
> | |
> | draw widget |
> | some other stuff... |
> +--+
>
> but I can't figure out if this is possible in IDL. The
> top-level base has COLUMN set, and then into that I
> first place a widget_base with ROW set.

The trick is to make the base for your labels a column base instead of a
row base. Yes this sounds odd, but the column keyword also is able to
specify the number of columns in a base, and this helps here. The other
needed part is to make the base gridded. The base definition would look
something like this:

wRow = widget_base(tlb, xsize=xsize, column=2, /grid)

Below is a very simple test case that handles resizing only in the X
direction. You can see that just by re-setting the xsize on the row
base that the labels will jump to where you want them to be. You must
set the xsize in order for this to happen, it will not resize and
reposition automagically, but carrying around an additional widget ID is
not that hard.

Cheers,
eddie

:.......................
;;;;;;;;;;;;;;;;;;;;;;;;

```
PRO test_event, ev

  IF (tag_names(ev, /structure_name) EQ 'WIDGET_BASE') THEN BEGIN
    widget_control, ev.top, get_uvalue=uval
    widget_control, uval.wDraw, xsize=ev.x
    widget_control, uval.wRow, xsize=ev.x
  ENDIF

END
```

:......................
;;;;;;;;;;;;;;;;;;;;;;;

PRO test

```
    xsize = 200
    tlb = widget_base(/column, /tlb_size_events)
    wRow = widget_base(tlb, xsize=xsize, column=2, /grid)
    wLabel = widget_label(wRow, value='align left', /align_left)
    wLabel = widget_label(wRow, value='align right', /align_right)
    wDraw = widget_draw(tlb, xsize=xsize)
    widget_control, tlb, /realize, set_uvalue={wRow:wRow, wDraw:wDraw}
    xmanager, 'test', tlb
```

END