Subject: Re: widget_base alignment question
Posted by David Fanning on Thu, 17 Feb 2005 04:25:30 GMT
View Forum Message <> Reply to Message

Marshall Perrin writes:

> Having read through the IDL documentation on widget_base, and
> skimmed through all the widget stuff in Dave Fanning's most excellent
> book, I'm still stuck on this one. This is easy in GTK! Why is it so
> hard (or at least obscure) in IDL?!

Oh, let's not start. :-)

I need this kind of thing from time to time, too.
It usually takes less time to look for a layout that
doesn't have the requirement than it does to code the
damn thing up, but here is what I do.

I calculate the size of the TLB. Then the size of the
things I want to put into that row base. I do some subtraction
to get the size of the "space" I want between the things in
the row base. I have a "spacer" object or widget that I use
to insert itself in the right location. Just a label widget
with nothing written on it. Of course, after doing all the
calculations I have to subtract a fudge factor of 17 or 23
or something like that. It varies with the platform and the
time of the month. (Or maybe it's the phase of the moon, I
can't remember.)

Anyway, when it's done it looks great on my machine and
lousy everywhere else. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: widget_base alignment question
Posted by eddie haskell on Thu, 17 Feb 2005 16:04:35 GMT
View Forum Message <> Reply to Message

> 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
```

Subject: Re: widget_base alignment question
Posted by David Fanning on Thu, 17 Feb 2005 18:58:08 GMT
View Forum Message <> Reply to Message

eddie haskell writes:

> 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)

Oh, yeah, I forgot about that. :-(

Cheers,

David

P.S. I better write that up before I forget it again. :-)

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: widget_base alignment question
Posted by JD Smith on Mon, 21 Feb 2005 20:53:50 GMT
View Forum Message <> Reply to Message

On Thu, 17 Feb 2005 09:04:35 -0700, eddie haskell wrote:

>> 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.
```

Useful layout trick. Does this work for everyone? Why does Linux suffer here? Unless I realize the widgets first, and set the xsize after, the labels are piled up on each other (see turtle.as.arizona.edu/idl/test_pre.png and turtle.as.arizona.edu/idl/test_fix.png). But even that doesn't really fix it, cutting off part of the "t" in "right".

Then when it gets resized, and a TLB_SIZE_EVENTS comes in, it is clearly lying about the "frame-free" size of the usable window area, and the widgets are resized to overfill the window (see turtle.as.arizona.edu/idl/test_post.png). Does anyone at RSI actually go through the layout on all of their systems to see what works?

JD
