Subject: how to resize a top level base widget
Posted by markb77 on Sat, 06 Apr 2013 19:48:32 GMT
View Forum Message <> Reply to Message

I've been messing around with resizing top level base widgets under Windows (IDL 8.0.1). After some experimentation, I've figured out one or two things that are worth noting for future reference.

As far as I can ascertain, the correct way to resize a TLB is to resize the widgets contained within it.

Let's say you have a graphics window which you want the user to be able to resize. You have set the TLB_SIZE_EVENTS keyword for the base. Now the user resizes the base and your event handler receives a resize event. What happens next?

If you use WIDGET_INFO to check the geometry of the TLB at this point, you get back the new size of the TLB.

The first time you use WIDGET_CONTROL to resize one of the child widgets of the TLB, however, the size of the TLB will "snap back" to the largest extents of its child widgets.

Therefore, the correct procedure for resizing is, after receiving the resize event, to use WIDGET_CONTROL to appropriately resize the widgets within your TLB in order to fill up the new space. This will result in a TLB that has the same size that the user selected.

Warning: DO NOT use WIDGET_CONTROL to directly resize the top level base widget. This results in strange behavior and will mess up your widget layout. If you need to programmatically resize the TLB, simply adjust the size of the widgets contained within, and the size of the TLB will change automatically.

I should note that In my case, the TLB specifies the COLUMN keyword, so the layout of the child widgets is managed by the top level base widget. I'm not sure if what I've described above applies when this isn't the case, or on platforms other than Windows.

Some of this is already covered in David's article about resizeable widget programs, but I got stuck on this a couple of times so I thought I'd add this to the list.

Mark