

---

Subject: Re: IDL compound widgets

Posted by [rmlongfield](#) on Thu, 15 Oct 1998 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

In article <7037o5\$kkq2@f1n1.spenet.wfu.edu>,

"Craig Hamilton" <cah@medeng.wfubmc.edu> wrote:

> Hi all:

>

> I'm trying to build a compound widget for image display. I want the user to

> be

> able to call it as:

> `cw_image(dataset)`

> where dataset is a two or three dimensional array, something like

> `intarr(256,256,30)`.

> I also want to be able to have multiple `cw_image` widgets active at the same  
> time.

> The `cw_image` widget needs access to the values in the dataset, to display  
> values

> as the cursor moves across the image. The compound widget info in the  
> manuals

> suggest stuffing a state variable in the user value of the next-to-the-top  
> widget in

> the tree, but I sure don't want to stuff the dataset in there. I can't use

> a COMMON

> because I want multiple `cw_image` widgets.

>

> So, the event handler in my `cw_image` widget really needs a pointer to  
> dataset, right?

> I can put the pointer in the state variable and get to the image data that  
> way. But, I

> don't want the user to have to mess with pointers; I want the data to come  
> in as

> an array.

>

> IDL doesn't seem to let me use the address of the array as a pointer, right?

> That is

> one difference between C pointers and IDL pointers.

>

> I hope this isn't too confusing, but the bottom line question is: How can a  
> compound

> widget receive a huge dataset and be able to read values from the dataset in  
> its event callback? [And be able to have multiple realizations of the

> widget active

> at the same time with different size data sets]

>

> Any tips most appreciated.

>

> Craig Hamilton

> cah@medeng.wfubmc.edu  
> The Wake Forest Univ. School of Medicine  
> Winston-Salem, NC, USA  
>  
>

Hi Craig, I have also been struggling with creating compound widgets. I have a panel of images from which I choose one or two to do some further operations. It is a work in progress. I have some suggestions. The point of CW's is that they act like a regular widget. I think, if you have your data array declared as a UVALUE for that CW widget, it should be accessible with a WIDGET\_CONTROL. This, an end user should be able to understand, if they understand the widgets.

I also started with a 'state' structure but this tends to get really big. I save this for really important stuff like widget ID's and try to keep it minimal. I have been putting the data pointer in the UVALUE of the Draw widget, rather than the state structure. If I need this data somewhere else, I write my own event structure (pseudo event), in which I can put anything I want AND send it exactly where I want. (The event handler is also defined by the same Draw widget). Finding out where to send it is another story, of course, but at the moment I usually send it to the event.top. It can read the identity of this pseudo event and act accordingly (including sending the data somewhere else). I've used David Fanning's book as a reference for designing the pseudo event.

I really like using pointers, although they are a little confusing when trying to get back the data (all those stars and parentheses). This makes it important to NOT make the state structure too complicated (i.e. having pointers to data structures with pointers). My rule of thumb is, if I find I am typing too many stars, it's time to re-formulate the algorithm. Thus, the pseudo events.

Just a few thoughts. I hope this helps. Also see Doug Larson's CW's which he posted here about two weeks ago.

Good luck.  
Rose

-----== Posted via Deja News, The Discussion Network ==-----  
<http://www.dejanews.com/> Search, Read, Discuss, or Start Your Own

---