Subject: Work-around for the Shortcomings of Widget_Tab
Posted by Joe[5] on Tue, 24 Apr 2007 12:08:13 GMT
View Forum Message <> Reply to Message

I have an app that is built up of a set of user-selected/defined tabs in a Widget_Tab. Some of these tabs need to be initialized (e.g., to repopulate droplists) every time their tab is selected. Since the set of tabs is user defined, tabs occur in no predictable order. The .TAB field from the event structure sent to the handler of the Widget_Tab is, therefore, pretty useless as it cannot be used to identify the widget or code associated with that tab. There is no way (that I can see in the documentation) to use either Widget_Info or Widget_Control to obtain the identity of the Widget_Base underlying a given tab. I have a work-around for this that is less than satisfactory. The title, widget id, and the name of an initialization function is stored in a structure array that can be indexed by the TAB field of the event structure. When the a tab is selected, I get the array from the UValue of the Widget_Tab and execute the initialization routine. That works for simple initializations where little or no data is required to initialize the widget. The price I have to pay is maintaining the structure array as tabs are removed or added. Also, for ease of code maintenance, I want to keep the initialization code with the rest of the code it is initializing rather than as part of the event handler for the Widget_Tab.

Has anyone else run into this and if so how did you work around it?

I'm using IDL v6.3. Personally, I find the fact that Widget_Info is unable to retrieve the title of a Widget_Base astounding, especially in light of the ability to retrieve TOOLTIPs! I would have expected that, at a minimum, Widget_Info could obtain the title of a base - especially since there are ways to change the title. Then, in much the same way that Widget_Button events can be processed based on their value, I could process the tab initialization. No the greatest way to do it, but simpler than what I am doing now.

Ideally, Widget_Base should have two new keywords, SELECT_FUNC & SELECT_PRO that identify the name of a function/procedure to call when the Widget_Base is associated with a Widget_Tab and that tab is selected. This would be similar to NOTIFY_REALIZE, but occur each time the associated tab is selected.

Without these, I have to develop the requirements and implement an interface by which new bits of code can register themselves with my app. This is something I really should not have to do.

Subject: Re: Work-around for the Shortcomings of Widget_Tab
Posted by David Fanning on Wed, 02 May 2007 16:46:20 GMT
View Forum Message <> Reply to Message

Joe@Zawodny.com writes:

> Thanks for the idea. It certainly does provide me with the widget_id
> of the base, but that is about it. I still do not know what code that
> widget is associated with nor how to go about initializing it properly
> - or if that is even needed. I could of course hide that info in the
> Base's Value or UValue, but I usually reserve those variables to store
> 'real data' (the way we're taught to use those fields to develop
> compound widgets and/or avoid the use of common blocks).

Humm. I'm still not sure I see the problem here. If you need more information from the widget you found, you could easily write a "GET_VALUE" function for that widget that can return to you any information you feel is relevant and you can get your hands on. Or, you could simply send an event to that widget of your own creation that asks it to initialize itself if it hasn't already done so, etc. The possibilities seem endless. All made immensely easier, of course, if your "widgets" are really "objects", but I assume you will realize this eventually yourself. :-)

> I hope you enjoyed the Iberian Peninsula!

I had a *fabulous* time. I have continued my Spanish studies and I'm looking for another opportunity to travel to a Spanish-speaking place. Tiene ideas? I'm offering substantial discounts on services for the right situation. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Work-around for the Shortcomings of Widget_Tab
Posted by Joe[5] on Wed, 02 May 2007 17:53:37 GMT
View Forum Message <> Reply to Message

On May 2, 12:46 pm, David Fanning <n...@dfanning.com> wrote:
> J...@Zawodny.com writes:

>> Thanks for the idea. It certainly does provide me with the widget_id
>> of the base, but that is about it. I still do not know what code that
>> widget is associated with nor how to go about initializing it properly
>> - or if that is even needed. I could of course hide that info in the
>> Base's Value or UValue, but I usually reserve those variables to store
>> 'real data' (the way we're taught to use those fields to develop
>> compound widgets and/or avoid the use of common blocks).
>
> Humm. I'm still not sure I see the problem here. If you
> need more information from the widget you found, you could
> easily write a "GET_VALUE" function for that widget that
> can return to you any information you feel is relevant and
> you can get your hands on. Or, you could simply send an
> event to that widget of your own creation that asks it to
> initialize itself if it hasn't already done so, etc. The
> possibilities seem endless. All made immensely easier, of
> course, if your "widgets" are really "objects", but I assume
> you will realize this eventually yourself. :-)
>
> Cheers,
>
> David

David,

I considered sending the base widget a special init event, but I could not assure myself that the event would be handled in a timely fashion. There are other events being generated (some of which are time consuming) while all of this is going on. Moreover, even if it is the best option available, it still means I have to design and implement an infrastructure where the Widget_Tab is required to take special actions whether the Widget_Base's underlying each tab require it or not. In that respect, sending a special event to the widget or looking up and calling an init_function are really not that different in principle - only in detail. Only those Widget_Bases really (can and should) know whether they require special attention or services. IDL should provide a transparent mechanism whereby those widgets can request special services (e.g., via SELECT_PRO or SELECT_FUNC keys to Widget_Base) and that all of this be done internally within the code that creates/services that particular widget. I agree wholeheartedly that there are many ways to skin this cat. No disagreements there. I'm saying that the cat should arrive pre-skinned.

Thanks for the ideas, I'll mull over them some more.

Joe

Subject: Re: Work-around for the Shortcomings of Widget_Tab
Posted by David Fanning on Wed, 02 May 2007 18:01:35 GMT
View Forum Message <> Reply to Message

Joe@Zawodny.com writes:

> I agree wholeheartedly
> that there are many ways to skin this cat. No disagreements there. I'm
> saying that the cat should arrive pre-skinned.
>
> Thanks for the ideas, I'll mull over them some more.

Well, I've offered to turn all their widgets into objects
for them (since I've already done it), but no takers yet. :-(

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")