
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
