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.