Subject: Re: Work-around for the Shortcomings of Widget_Tab
Posted by David Fanning on Tue, 24 Apr 2007 14:40:57 GMT
View Forum Message <> Reply to Message

Joe@Zawodny.com writes:

> 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.

This workaround feels a little too complicated for my tastes.
I think what I would do is use the event.tab field, which
you know is the index number of the child base, to fish
out the required base. You could do something like this
to identify the base widget:

```
CASE event.tab OF
   0: theBase = Widget_Info(event.id, /Child)
   ELSE: BEGIN
      theChild = Widget_Info(event.id, /Child)
      FOR j=1,event.tab DO BEGIN
         theChild = Widget_Info(theChild, /Sibling)
      ENDFOR
      theBase = theChild
      END
 ENDCASE
```

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 16:29:02 GMT
View Forum Message <> Reply to Message

On Apr 24, 10:40 am, David Fanning <n...@dfanning.com> wrote:
> J...@Zawodny.com writes:
>> 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.
>
> This workaround feels a little too complicated for my tastes.
> I think what I would do is use the event.tab field, which
> you know is the index number of the child base, to fish
> out the required base. You could do something like this
> to identify the base widget:
>
> CASE event.tab OF
> 0: theBase = Widget_Info(event.id, /Child)
> ELSE: BEGIN
> theChild = Widget_Info(event.id, /Child)
> FOR j=1,event.tab DO BEGIN
> theChild = Widget_Info(theChild, /Sibling)
> ENDFOR
> theBase = theChild
> END
> ENDCASE
>
> 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.")

David,

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). So even with this widget_id, I'm still stuck with the need to develop a framework where these bases register with the mother application. I guess that I'll stick with my brute force method while holding my breath for the next version of IDL to include the SELECT_FUNC and SELECT_PRO keywords I asked for originally. I really think this is the only clean, self-contained, maintainable way to build these sorts of widget apps. Any other approach requires too much interconnection between the levels of the widget hierarchy and likely violates good object oriented practices. I'll need to think about this a bit more to see if there is

a way to semi-transparently hide this info in the Value and trigger the initialization by via the Func_Get_Value funtion I assign to the base. It still smacks of a kluge though and results in code that is not standalone or really reusable.

I hope you enjoyed the Iberian Peninsula!

Joe
