Subject: Classes and Widget Event Handlers...
Posted by dmorris on Fri, 02 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

I have a problem I am hoping someone on the group can help me with:

I have a class with a lot of variables (a few hundred). The class
works very wel and is a big improvement over the previous
design...with one exception. We use as widget to input numerous
values for generating a plot.

The event-handler for the widget needs to have access to all of the
variables of the class, and it must have direct access to them.
Unfortunately, XMANAGER (as far as I can tell) does not allow the use
of object method-routines as an event handler.

Can anyone tell me a way to give my event handler access to class
variables.

--David

Subject: Re: Classes and Widget Event Handlers...
Posted by davidf on Sat, 03 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Struan Gray (struan.gray@sljus.lu.se) writes:

> When this was originally posted in June '97, Ronn Kling and Bob
> Mallozzi pointed out you could exploit the way that IDL calls and
> defines event handlers. Ronn Kling used the EVENT_PRO keyword when creating
> the top level base to define EXAMPLE::EVENT as an event procedure.

I don't recall now if Ronn actually did this or not, but
if he did it was a BAD idea! Using the EVENT_PRO keyword
to assign the event handler to the top-level base can result
in all kinds of havoc. This event handler should *always* be
assigned with the Event_Handler keyword on the XManager
routine. (By the way, I am explicitly talking about a
top-level base that is being *managed* by XManager.)

> Both techniques look powerful (and, more importantly, cool :-),
> but both seem to rely on the ability to invoke an object method as if
> it were a normal procedure, something that should, formally, be
> impossible.

I think it is impossible (or, more accurately, I haven't
discovered a way to do it). But, by putting the "self"

structure in the user value of the top-level base, each
event handler has access to the self object pretty much
directly. True, you have to interact with it through
methods, but for most programs these methods are super
simple to write. Certainly easier, most of the time,
than writing the requisite event handler code.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Classes and Widget Event Handlers...

Posted by ronn on Mon, 05 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <MPG.11e88590a993852b989810@news.frii.com>,
 davidf@dfanning.com (David Fanning) wrote:
> Struan Gray (struan.gray@sljus.lu.se) writes:
>
>> When this was originally posted in June '97, Ronn Kling and Bob
>> Mallozzi pointed out you could exploit the way that IDL calls and
>> defines event handlers. Ronn Kling used the EVENT_PRO keyword when
creating
>> the top level base to define EXAMPLE::EVENT as an event procedure.
>
> I don't recall now if Ronn actually did this or not, but
> if he did it was a BAD idea! Using the EVENT_PRO keyword
> to assign the event handler to the top-level base can result
> in all kinds of havoc. This event handler should *always* be
> assigned with the Event_Handler keyword on the XManager
> routine. (By the way, I am explicitly talking about a
> top-level base that is being *managed* by XManager.)

I actually used event_pro='example::event' in a button definition
statement as I recall. More importantly, this did work in the very
first version of IDL 5.0. But as I remember it disappeared in the 5.0.a
release.
>
>> Both techniques look powerful (and, more importantly, cool :-),
>> but both seem to rely on the ability to invoke an object method as
if

>> it were a normal procedure, something that should, formally, be
>> impossible.
>
> I think it is impossible (or, more accurately, I haven't
> discovered a way to do it). But, by putting the "self"
> structure in the user value of the top-level base, each
> event handler has access to the self object pretty much
> directly.

As of today it is still impossible, but I keep asking RSI to allow us to
be able to specify object methods in the event_pro and event_func
keywords. We can get around it, but it adds just enough obfuscation to
the code that it makes me uncomfortable. Just think if we could do
this... No more non-object event handling code, no more
widget_control,event.top,get_uvalue=state (or self ,etc.). Almost
everything that we write could be objects!

-Ronn

--
Ronn Kling
Ronn Kling Consulting
www.rlkling.com