
Subject: Event handler as an object method ??

Posted by [Antonio Santiago](#) on Tue, 09 Nov 2004 09:55:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi, is there a reason why an object method can't be used as an event handler?

I just created an object that "contains" a WIDGET_DRAW. I'd like that the event occurred on WIDGET_DRAW was handled by a method of my class, but EVENT_FUNCTION and EVENT_PRO keywords do not accept a method.

The solution I have adopted is to catch the event of WIDGET_DRAW outside of the object (on a WIDGET_BASE where I put the object (really de WIDGET_DRAW)) and redirect it to the method I want (MyObject->EventHandler), but this is an ugly solution.

Any ideas ??

Thanks.

Subject: Re: Event handler as an object method ??

Posted by [David Fanning](#) on Tue, 09 Nov 2004 12:30:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

Antonio Santiago writes:

> Hi, is there a reason why an object method can't be used as an event handler?

No reason at all. In fact, it is only a short step now before you realize widgets should be objects. :-)

>

> I just created an object that "contains" a WIDGET_DRAW. I'd like that
> the event occurred on WIDGET_DRAW was handled by a method of my class,
> but EVENT_FUNCTION and EVENT_PRO keywords do not accept a method.

>

> The solution I have adopted is to catch the event of WIDGET_DRAW outside
> of the object (on a WIDGET_BASE where I put the object (really de
> WIDGET_DRAW)) and redirect it to the method I want
> (MyObject->EventHandler), but this is an ugly solution.

In the absence of help from RSI, most solutions are more or less ugly. The solution I usually adopt is to have all events go to a generic event handler. In the event handler, a "command" structure is extracted from the user value of the widget that caused the event. The "command" is an anonymous structure containing an "object" and a "method" field. All the event handler does is extract the command structure and use CALL_METHOD to call the method field on the object field, passing the event structure

as a parameter to the "event handler method".

To make this work, every widget that is going to generate an event has a "command" structure stored in its user value:

```
drawID = Widget_Draw(tlb, XSize=400, YSize=400, $
    UValue={object:myobject, method:'MyEventHandlerMethod'})
```

The generic widget event handler does this:

```
Widget_Control, event.id, Get_UValue=cmd
Call_Method, cmd.method, cmd.object, event
```

In my Catalyst Library, all widgets are objects and the "fields" of the "event structure" all point to objects as well, so this slight of hand feels a little more natural. Instead of using Event_Pro or Event_Func to direct events, you can use an Event_Method keyword to define the appropriate method to handle the event. If an event method is not specified, we use a EventHandler method that is always associated with an widget object.

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Phone: 970-221-0438, IDL Book Orders: 1-888-461-0155

Subject: Re: Event handler as an object method ??

Posted by [Benjamin Hornberger](#) on Thu, 11 Nov 2004 14:26:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Antonio Santiago writes:

>

>

>> Hi, is there a reason why an object method can't be used as an event handler?

>

>

> No reason at all. In fact, it is only a short step

> now before you realize widgets should be objects. :-)

>

>> I just created an object that "contains" a WIDGET_DRAW. I'd like that

```

>> the event occurred on WIDGET_DRAW was handled by a method of my class,
>> but EVENT_FUNCTION and EVENT_PRO keywords not accpet a method.
>>
>> The solution i have adopted is to catch the event of WIDGET_DRAW outside
>> of the object (on a WIDGET_BASE where i put the object (realy de
>> WIDGET_DRAW)) and redirect it to the method i want
>> (MyObject->EventHandler), but this is an ugly solution.
>
>
> In the absence of help from RSI, most solutions are more
> or less ugly. The solution I usually adopt is to have
> all events go to a generic event handler. In the event
> handler, a "command" structure is extracted from the user
> value of the widget that caused the event. The "command"
> is an anonymous structure containing an "object" and
> a "method" field. All the event handler does is extract
> the command structure and use CALL_METHOD to call the
> method field on the object field, passing the event structure
> as a parameter to the "event handler method".
>
> To make this work, every widget that is going to generate
> an event has a "command" structure stored in its user value:
>
> drawID = Widget_Draw(tlb, XSize=400, YSize=400, $
>   UValue={object:myobject, method:'MyEventHandlerMethod'})
>
> The generic widget event handler does this:
>
> Widget_Control, event.id, Get_UValue=cmd
> Call_Method, cmd.method, cmd.object, event
>
> In my Catalyst Library, all widgets are objects and the
> "fields" of the "event structure" all point to objects
> as well, so this slight of hand feels a little more
> natural. Instead of using Event_Pro or Event_Func
> to direct events, you can use an Event_Method keyword
> to define the appropriate method to handle the event
> If an event method is not specified, we use a EventHandler
> method that is always associated with an widget object.
>
> Cheers,
>
> David
>

```

Is it right that in this case (as David proposes), you don't need the

widget_control, event.top, get/set_uvalue=info, /no_copy

any more to pass the widget's (now object's) internal variables around?
Instead, you can just access the members of the self structures in the
"event handler"? That would be nice...

Benjamin

Subject: Re: Event handler as an object method ??
Posted by [David Fanning](#) on Thu, 11 Nov 2004 14:40:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Benjamin Hornberger writes:

> Is it right that in this case (as David proposes), you don't need the
>
> widget_control, event.top, get/set_uvalue=info, /no_copy
>
> any more to pass the widget's (now object's) internal variables around?
> Instead, you can just access the members of the self structures in the
> "event handler"? That would be nice...

Oh, not just "nice", but the way it *should* be! :-)

Cheers,

David

--

David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Phone: 970-221-0438, IDL Book Orders: 1-888-461-0155
