Subject: event_pro for compound widgets
Posted by Benjamin Hornberger on Tue, 07 Sep 2004 17:32:41 GMT
View Forum Message <> Reply to Message

Hi all,

I am quite confused now and can't figure out how to write a compound widget which has an event_pro keyword and behaves like a regular basic widget. Sorry for the lengthy message...

My compound widget is supposed to control a stepping motor and will consist of a non-editable field with the current position, editable text fields for "move to position" and "move by distance", a slider to set a step size and two buttons to move forward and backward in steps whose step size is set by the slider. As text fields, I am using David Fanning's fsc_field.

This compound widget I want to include in other widgets by something like

```
motor1_id = cw_move_motor(parent, ..., event_pro = 'motor1_event')
```

When I change the step size of the slider, this should only affect the CW itself (updating its internal "stepsize" value and a displayed number"). I could manage to do that already.

All other events (hitting enter in the moveto or moveby field, or clicking the forward / backward buttons) should send an event structure to motor1_event.pro (something like { CW_MOVE_MOTOR, id: id, top: top, handler: handler, type: 0L, moveby: 0D, moveto: 0D, stepsize: 0D } where type holds a code for whether it was a moveto or moveby request, etc.).

Then, of course I would want to write

```
PRO motor1_event, event
  ...
  IF event.type EQ 0 ...
```

Now, what do I do? In the CW definition function, I define event_func='cw_move_motor_event' for my fsc_fields, buttons etc., This cw_move_motor_event.pro will create and return the desired event structure. But where is that returned to?

And of course I have to define a keyword event_pro in my CW definition function which will hold a string with the event handler procedure's name. How do I make sure that in case of an event, my event structure is created and passed to that procedure?

Thanks for your help,

Benjamin

Subject: Re: EVENT_PRO
Posted by David Fanning on Mon, 29 Nov 2010 16:46:42 GMT

View Forum Message <> Reply to Message

silje writes:

```
>
> Hey, I'm trying to structure my IDL program into several event
> handlers, e.g. one procedure that opens a file, one that save data
> etc. To do this I have used EVENT_PRO like this:
>
> PRO MotionControl
>
> tlb = WIDGET_BASE(column=1, title='MotionControl', tlb_frame_attr=1,
> MBAR=bar )
>   file_menu = WIDGET_BUTTON(bar, VALUE='File', /MENU)
>   file_bttn1 = WIDGET_BUTTON(file_menu, VALUE='Open dataset',
> UVALUE='OpenDat', EVENT_PRO = 'OpenData')
>   file_bttn2 = WIDGET_BUTTON(file_menu, VALUE='Save dataset',
> UVALUE='SaveDat', EVENT_PRO = 'SaveData')
>
>  ......................................................... ..........................
> etc
>
>
> WIDGET_CONTROL, tlb, /realize
> XMANAGER, 'MotionControl', tlb
>
> end
>
> In this way I have one event handler  that is called
> MotionControl_event that takes care of all the events not handled by
> specified EVENT_PRO. I don't know if this is a really bad way of
> structuring my code, even though it does work. Can somebody give me
> some tips?
```

It is a perfectly OK way to structure your code,
although you will get in trouble sooner or later
with the names of these routines. I would read the
"How to Name Files" article again. :-)

   http://www.dfanning.com/tips/namefiles.html

These event handlers should have names like
"MotionControl_SaveData" and "MotionControl_OpenData".

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: EVENT_PRO
Posted by silje on Mon, 29 Nov 2010 17:22:48 GMT

View Forum Message <> Reply to Message

On 29 Nov, 17:46, David Fanning <n...@dfanning.com> wrote:
> silje writes:
>
>> Hey, I'm trying to structure my IDL program into several event
>> handlers, e.g. one procedure that opens a file, one that save data
>> etc. To do this I have used EVENT_PRO like this:
>
>> PRO MotionControl
>
>> tlb = WIDGET_BASE(column=1, title='MotionControl', tlb_frame_attr=1,
>> MBAR=bar)
>>   file_menu = WIDGET_BUTTON(bar, VALUE='File', /MENU)
>>   file_bttn1 = WIDGET_BUTTON(file_menu, VALUE='Open dataset',
>> UVALUE='OpenDat', EVENT_PRO = 'OpenData')
>>   file_bttn2 = WIDGET_BUTTON(file_menu, VALUE='Save dataset',
>> UVALUE='SaveDat', EVENT_PRO = 'SaveData')
>
>> ....
>> etc
>
>> WIDGET_CONTROL, tlb, /realize
>> XMANAGER, 'MotionControl', tlb
>
>> end
>
>> In this way I have one event handler that is called
>> MotionControl_event that takes care of all the events not handled by
>> specified EVENT_PRO. I don't know if this is a really bad way of
>> structuring my code, even though it does work. Can somebody give me
>> some tips?

>
> It is a perfectly OK way to structure your code,
> although you will get in trouble sooner or later
> with the names of these routines. I would read the
> "How to Name Files" article again. :-)
>
> http://www.dfanning.com/tips/namefiles.html
>
> These event handlers should have names like
> "MotionControl_SaveData" and "MotionControl_OpenData".
>
> 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.")

Changing the names to something more meaningful shouldn't be any
problem :) But it still feels like an easy solution. I also found this
in a previous discussion

"If you do use EVENT_PRO to assign an event handler for a top-level
base being directly managed by XMANAGER, you will find exceedingly
strange things going on in your widget program, if it works at all.
Believe me, you don't want to do this. "

http://www.dfanning.com/widget_tips/multiple_widgets.html

Could you explain what you meant by that?

Silje

Subject: Re: EVENT_PRO
Posted by David Fanning on Mon, 29 Nov 2010 18:38:21 GMT
View Forum Message <> Reply to Message

silje writes:

> Changing the names to something more meaningful shouldn't be any
> problem :) But it still feels like an easy solution. I also found this
> in a previous discussion
>

>   "If you do use EVENT_PRO to assign an event handler for a top-level
> base being directly managed by XMANAGER, you will find exceedingly
> strange things going on in your widget program, if it works at all.
> Believe me, you don't want to do this. "
>
> http://www.dfanning.com/widget_tips/multiple_widgets.html
>
> Could you explain what you meant by that?

In the old days, a widget that was being directly managed by XManager (ie, the widget ID used in the XManager call) could not be assigned an event handler with either EVENT_PRO or EVENT_FUNC keywords. Rather, you had to assign an event handler to this widget with the EVENT_HANDLER keyword to XManager.

I still follow this rule in my own widget programs because, well ..., because I'm old school. :-)

I believe this requirement is no longer in effect, although I haven't really tested it. I just see a lot of programs written by ITTVIS that assign the top-level base an event handler using the EVENT_PRO keyword. So I presume this restriction is no longer valid.

You could test it and let me know. :-)

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: EVENT_PRO
Posted by Paul Van Delst[1] on Mon, 29 Nov 2010 18:43:49 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> silje writes:
>
>> Hey, I'm trying to structure my IDL program into several event

```
>>  handlers, e.g. one procedure that opens a file, one that save data
>>  etc. To do this I have used EVENT_PRO like this:
>>
>>  PRO MotionControl
>>
>>  tlb = WIDGET_BASE(column=1, title='MotionControl', tlb_frame_attr=1,
>>  MBAR=bar )
>>    file_menu = WIDGET_BUTTON(bar, VALUE='File', /MENU)
>>    file_bttn1 = WIDGET_BUTTON(file_menu, VALUE='Open dataset',
>>  UVALUE='OpenDat', EVENT_PRO = 'OpenData')
>>    file_bttn2 = WIDGET_BUTTON(file_menu, VALUE='Save dataset',
>>  UVALUE='SaveDat', EVENT_PRO = 'SaveData')
>>
>>   ........................................................ ..........................
>>  etc
>>
>>
>>  WIDGET_CONTROL, tlb, /realize
>>  XMANAGER, 'MotionControl', tlb
>>
>>  end
>>
>>  In this way I have one event handler  that is called
>>  MotionControl_event that takes care of all the events not handled by
>>  specified EVENT_PRO. I don't know if this is a really bad way of
>>  structuring my code, even though it does work. Can somebody give me
>>  some tips?
>
>  It is a perfectly OK way to structure your code,
>  although you will get in trouble sooner or later
>  with the names of these routines. I would read the
>  "How to Name Files" article again. :-)
>
>     http://www.dfanning.com/tips/namefiles.html
>
>  These event handlers should have names like
>  "MotionControl_SaveData" and "MotionControl_OpenData".
```

Or maybe "MotionControl::SaveData" and "MotionControl::OpenData" ?

Can objects be used this way in widget programming? If not, does the Catalyst library do all the necessary heavy lifting?

I'm at the tipping point of wanting to learn something new in IDL widget-y stuff now that I'm IDL-OO-enthused. Using IDL
objects has made my code design a lot easier to understand (despite the necessary increase in actual LOC) and if there's
one thing my IDL widget programs can greatly benefit from, it's better design!

cheers,

paulv

Subject: Re: EVENT_PRO
Posted by David Fanning on Mon, 29 Nov 2010 19:02:26 GMT
View Forum Message <> Reply to Message

Paul van Delst writes:

> Or maybe "MotionControl::SaveData" and "MotionControl::OpenData" ?
>
> Can objects be used this way in widget programming? If not, does the Catalyst library do all the necessary heavy lifting?
>
> I'm at the tipping point of wanting to learn something new in IDL widget-y stuff now that I'm IDL-OO-enthused. Using IDL
> objects has made my code design a lot easier to understand (despite the necessary increase in actual LOC) and if there's
> one thing my IDL widget programs can greatly benefit from, it's better design!

Well, I can't even remember the details of how
Catalyst does this, so yes, I presume it does
all the necessary heavy lifting. It does just
seem to work. :-)

All widget events in the Catalyst Library (where all
widgets are objects) go to a single "event dispatcher".
(You can find this in the Utilities directory.) The
event dispatcher's job is to determine where the event
should be sent. Each widget has a container with
"event objects" stored there. The event dispatcher
essentially collects the event objects and calls the
EventHandler method on these event objects.

This means that in a Catalyst application a single
"event" might be sent to several different "event
handlers", rather than to a single event handler,
as happens in a strict widget program.

In Catalyst you assign an "event handler" to a widget
with the EVENT_OBJECT keyword. There is an optional
EVENT_METHOD keyword that can be used select the actual event
method for this event object. Otherwise, the event
is always sent to the EventHandler method of the
event object.

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: EVENT_PRO
Posted by silje on Mon, 29 Nov 2010 19:02:51 GMT
View Forum Message <> Reply to Message

On 29 Nov, 19:38, David Fanning <n...@dfanning.com> wrote:
> silje writes:
>> Changing the names to something more meaningful shouldn't be any
>> problem :) But it still feels like an easy solution. I also found this
>> in a previous discussion
>
>> "If you do use EVENT_PRO to assign an event handler for a top-level
>> base being directly managed by XMANAGER, you will find exceedingly
>> strange things going on in your widget program, if it works at all.
>> Believe me, you don't want to do this. "
>
>> http://www.dfanning.com/widget_tips/multiple_widgets.html
>
>> Could you explain what you meant by that?
>
> In the old days, a widget that was being directly managed
> by XManager (ie, the widget ID used in the XManager call)
> could not be assigned an event handler with either EVENT_PRO
> or EVENT_FUNC keywords. Rather, you had to assign an
> event handler to this widget with the EVENT_HANDLER keyword
> to XManager.
>
> I still follow this rule in my own widget programs because,
> well ..., because I'm old school. :-)
>
> I believe this requirement is no longer in effect, although
> I haven't really tested it. I just see a lot of programs
> written by ITTVIS that assign the top-level base an event
> handler using the EVENT_PRO keyword. So I presume this
> restriction is no longer valid.
>
> You could test it and let me know. :-)
>

> 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.")

I have created many separate event handlers using EVENT_PRO and it
works just fine :) I was just a little nervous it was bad
programming..... ;)

Silje

Subject: Re: EVENT_PRO
Posted by David Fanning on Mon, 29 Nov 2010 19:09:25 GMT
View Forum Message <> Reply to Message

silje writes:

> I have created many separate event handlers using EVENT_PRO and it
> works just fine :) I was just a little nervous it was bad
> programming..... ;)

It is not bad programming. It is good modular programming.
And, of course, it leads you, sooner rather than later, to
wonder why widgets aren't objects as they should be, so you
don't have to be copying info or state structures all over
the damn place! :-)

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: EVENT_PRO
Posted by David Fanning on Mon, 29 Nov 2010 19:40:19 GMT

David Fanning writes:

> This means that in a Catalyst application a single
> "event" might be sent to several different "event
> handlers", rather than to a single event handler,
> as happens in a strict widget program.
>
> In Catalyst you assign an "event handler" to a widget
> with the EVENT_OBJECT keyword. There is an optional
> EVENT_METHOD keyword that can be used select the actual event
> method for this event object. Otherwise, the event
> is always sent to the EventHandler method of the
> event object.

I should probably point out that "event objects" don't have to be widgets. In fact, most of the time they are not. If you changed a colorbar object, for example, you might wish to update any number of image objects with the new color information. In this case, the event objects would be these image objects who need to know about the colorbar "event".

There is a parallel system of "messages" in the Catalyst Library that act in a similar way. An object can register an "interest" in the actions of another object and be "notified" when that action occurs. This is really just a more specific instance of "event handling" in objects.

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.")