
Subject: Re: Lions and tiger and objects, oh my!
Posted by [David Fanning](#) on Fri, 31 Oct 2008 20:45:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Vince Hradil writes:

> So here's the deal - I decided to take the plunge and try to write an
> app using objects. Specifically, I'm using a lot of the Catalyst
> library as a jumping off point. So far, I have crashed IDLDE about a
> dozen times this afternoon, but I have been able to actually create a
> TLB with some widgets on it. Still no event handling, though.

Whoa! I have to admit, it has been a while since I crashed
IDL. Feel free to ask questions. It will get me started on
the documentation. :-)

Remember all widget events go through the CatEventHandler
and get reformatted as an event structure with objects. You
can set the EVENT_OBJECT keyword on any widget-object in
the same way you might have set the EVENT_PRO keyword
previously. The events are then dispatched to the object
you choose, and are (normally) handed by that object's
EventHandler method. (Although you can specify the method,
too.)

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: Lions and tiger and objects, oh my!
Posted by [Mike\[2\]](#) on Mon, 03 Nov 2008 14:19:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Oct 31, 3:45 pm, David Fanning <n...@dfanning.com> wrote:

> Remember all widget events go through the CatEventHandler
> and get reformatted as an event structure with objects. [...]

Hi David,

I haven't looked at your code in much detail yet, but I'm curious
about CatEventHandler. Does it function like IDL's

generic_class_event, generic_class_notify_realize and generic_kill_notify? Is it better/worse/different? Can you provide a comparison? Also, is it really called CatEventHandler? I don't see it anywhere in my copy of the sources...

Mike

Subject: Re: Lions and tiger and objects, oh my!
Posted by [David Fanning](#) on Mon, 03 Nov 2008 14:48:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mike writes:

> I haven't looked at your code in much detail yet, but I'm curious
> about CatEventHandler. Does it function like IDL's
> generic_class_event, generic_class_notify_realize and
> generic_kill_notify? Is it better/worse/different? Can you provide a
> comparison?

I've never used these routines, nor can I find them, but I imagine they are designed to do exactly the same thing: throw widget events and callbacks into the object system.

So, in that sense, generic_class_event is probably similar to CatEventHandler (not it's real name, see below). However, CatEventHandler is **much** more, as it is designed to deal with objects and not just widget events. So the "event" structure is repackaged so that the relevant fields in the structure are objects and not widget identifiers. In other words, event.id is an object reference to the object that caused the event, not a widget identifier of the widget that caused the event. Other fields are similarly modified.

> Also, is it really called CatEventHandler? I don't see
> it anywhere in my copy of the sources...

No, it is called CatEventDispatcher. I call it by some other name to throw would-be users completely off the track :-(

The routines you are looking for are found in the "utilities" directory of the Catalyst distribution:

CatEventDispatcher
CatKillNotify
CatRealizeNotify

There is debugging code in CatEventDispatcher that can be uncommented. This is extremely useful at times for figuring out where your events are going. Since all objects in the Catalyst system have "names" it is easy to see which object is sending, and which object is receiving, an event.

I am getting **very** close to an "official" release of this material, but there are still changes going on almost daily. Last night, for example, I fixed a long-time vexing problem with CatAtom, the fundamental object in the system, that was giving me some trouble with object deep-copying. Now, basically, any object in the system can make an identical copy of itself by simply calling the COPY method on itself. Pretty slick, and used in the right situations, enormously useful. :-)

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: Lions and tiger and objects, oh my!
Posted by [Mike\[2\]](#) on Tue, 04 Nov 2008 21:31:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Nov 3, 9:48 am, David Fanning <n...@dfanning.com> wrote:

> I've never used these routines, nor can I find them, but
> I imagine they are designed to do exactly the same thing:
> throw widget events and callbacks into the object system.

Yep - I got them from the IDL code library. They use a uvalue to get the object and then call the appropriate method. I associate a base with each object (shades of read_interfile here) and set up the event_pro like this:

```
widget_control, self.base, set_uvalue=self,  
event_pro='generic_class_event'  
xmanager, 'generic_class', self.base, /no_block
```

The generic_class_event routine looks like this:

```
; June, 2001 : JLP, RSI
Pro Generic_Class_Event, Event
COMPILE_OPT STRICTARR
Widget_Control, Event.Handler, Get_UValue = oSelf
If (N_elements(oSelf) eq 1) then Begin
  If (Obj_Valid(oSelf)) then Begin

    ;; A class that uses this routine must have a method
    ;; named "::EVENT".
    oSelf->Event, Event
  EndIf
EndIf
End
```

so it calls the event method of the object. And similarly for the notify methods.

```
>> Also, is it really called CatEventHandler? I don't see
>> it anywhere in my copy of the sources...
>
> No, it is called CatEventDispatcher. I call it by some
> other name to throw would-be users completely off the
> track :-(
```

It worked :-) I'll take a look.

Thanks, Mike
