Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets

Posted by JD Smith on Fri, 23 Apr 2004 18:21:07 GMT

View Forum Message <> Reply to Message

On Thu, 22 Apr 2004 21:30:40 -0600, David Fanning wrote:

> Sean Dettrick writes:
>
>> The handle_all_events routine inspects the UVALUE of the widget
>> causing the event, and uses CALL_METHOD to call my ReDraw method on
>> the object (David Fanning credits Stein Vidar for this idea)
>
> Really!? I guess I thought I had "invented" it myself.
> Maybe I re-discovered it independently after Stein Vidar
> showed it to me and I forgot all about it. :-(

Wait one minute... I thought *I* invented it. Call the patent
attorneys. Actually, it's a fairly obvious extension of the object
event callback. Another level of abstraction that I find useful is to
give individual widgets a UVALUE that is just a method name, or
"something else", which I call an "action". So then the event handler
can do one of several things:

1. Re-write the action based on some condition.
2. Handle the action directly itself.
3. Let the action fall through to a method call.

The event handlers look in skeleton form like:

```
pro object_event, ev
  widget_control, ev.top, get_uvalue=self
  self->Event,ev
end
```

Notice how the self object is retrieved from the top level base, not the
widget itself...

```
pro Event, ev
 widget_control, ev.id, get_uvalue=action

 ;; Example action rewrite
 if ev.clicks eq 2 then action='viewrecord'

 case action of
   'someaction': print,'Got some action'
   'save-as': self->Save,/AS
   ...
```

```
    else: call_method,action,self ;all others, just call the named method
 endcase
end
```

and when you're setting up the widgets:

```
b1=widget_button(base,VALUE='Save Project...',uvalue='save')
...
widget_control, base,SET_UVALUE=self,/REALIZE
 XManager,'MyObject',base,/NO_BLOCK,EVENT_HANDLER='object_eve nt'
```

The nice thing about this system: you can decide whether to handle the
event in-place, or farm it out to a method, and multiple different
widgets can trivially call the same method. Each widget only needs to
know its "action" and doesn't need a copy of the self object (which is
just stuck in the TLB).

JD