Subject: Re: Widgets and research
Posted by rivers on Sat, 14 Jun 1997 07:00:00 GMT

View Forum Message <> Reply to Message

In article <MPG.e0b229eb4d1d1e79896a6@news.frii.com>, davidf@dfanning.com (David Fanning) writes:

```
> Edward S. Meinel writes:
>
>>  On the other hand, I am reluctant to try IDL 5.beta because of all the
>>  reports of broken widgets. Do _any_ widgets developed with IDL 4 work
>>  under 5.0?
>
> As a matter of fact, *all* of my widgets work under IDL 5.0.
> A few of them don't *look* exactly like they did under IDL 4,
> but the problems are aesthetic rather than functional. So
> far every one of them has worked like I expect it to.
```

I agree with David. I just finished converting a >3,000 line IDL widget application to being "object oriented", using IDL Objects (but not Object Graphics). The widgets worked perfectly under IDL 5.0 (Motif) with exactly the same layout as 4.0 as far as I can see.

The only code which broke under IDL 5.0 was pickfile(). It used to return "path+file" when selecting existing files, and only "file_name" when entering the name of a new file. I think the old behavior was a bug.

IDL Objects really simplify complex widget programs, because so don't have to create a "state" structure and stick in the uvalue of a top level widget. Just put the object reference to "self" there instead. This is not entirely obvious, and the Object manual does not have an example, so here is how I do it.

```
;;  File example__define.pro

function example::init
    base = widget_base(uvalue=self)
    self.widgets.base = base
    self.widgets.exit = widget_button(base, value='Exit')
    widget_control, base, /realize
    xmanager, 'example::init', base, event='example_event', /no_block
    return, 1
end

pro example_event, event
    ; Note: The main event handler CANNOT be an object method, since xmanager
    ; won't know how to call it as such.  However, it only needs to be 4 lines
    ; long: retrieve the object reference and call the object method event
    ; handler, which will know about the objects data structure.
```

```
   widget_control, event.top, get_uvalue=object
   object->event, event
end

pro example::event, event
   ; This is the event handler which knows about the object
   case event.id of
      self.widgets.exit:  begin
         widget_control, event.top, /destroy
      end
      else: print, 'Unknown widget event'
   endcase
end

pro example__define
   widgets={example_widgets, base: 0L, exit: 0L}
   data = fltarr(30)
   t = {example, widgets: widgets, data: data}
end
```

Mark Rivers (773) 702-2279 (office)
CARS (773) 702-9951 (secretary)
Univ. of Chicago (773) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars.uchicago.edu (e-mail)

or:
Argonne National Laboratory (630) 252-0422 (office)
Building 434A (630) 252-0405 (lab)
9700 South Cass Avenue (630) 252-1713 (beamline)
Argonne, IL 60439 (630) 252-0443 (FAX)