Subject: Re: base widgets growing uncontrollably.... ?
Posted by Paul van Delst on Mon, 30 Jul 2001 13:44:50 GMT
View Forum Message <> Reply to Message

Jim Pendleton wrote:
>
> "JD Smith" <jdsmith@astro.cornell.edu> wrote in message
> news:3B61FE09.765EE529@astro.cornell.edu...
>
> [...stuff]
>
>> Paul van Delst wrote:
>
> [...more stuff]
>>>
>>> JD Smith wrote:
>>>>
>>>> And to point out the obvious, there's no reason you can't make
> compound
>>>> widgets also objects, rather than having an all-in-one object widget
>>>> design. You might then have a larger object interface which
> "composits"
>>>> (i.e. includes) the sub-objects directly, perhaps creating them
> itself.
>>>>
>>
>>
>> Just think how little restful break-time you'd get in a day if not for
>> overly compact programmatic mechanisms.
>>
>> Objectingly(ively?) yours,
>>
>> JD
>>
>> P.S. My one line object tutorial:
>>
>> objects->DoThings, withstuff, LIKE_STRUCTURES=only_smarter

I liked the one line tutorial - it actually made things clearer. Seriously. :o)

> If you're writing applications of more than a couple hundred lines and
> you're not using an object-based event handler described by JD earlier,
> you're missing the revolution.

Uh-oh. Who will be the first against the wall when the revolution comes? :o)

> And if you're writing class functionality that *relies on* the
> IDL widget system, you're missing the next revolution. A GUI is no

> more than an accessor to class data and methods. With sufficient
> forethought in design, IDL classes should be able to interact with Ion, VB,
> C++, perl, python or any other front-end.

I have no doubt all this stuff is very important. But in context. My interest is to look at my data. That's pretty much it. I slapped together my widget program (which was significantly less than a couple hundred lines) solely because I was tired of backspacing and changing an structure array index on the IDL command line to look at a different variable; that is to say, my class functionality consists of PLOT and SHADE_SURF. For what I do, IDL is simply an easy to use (most of the time!), relatively non-specialised tool to let me visualise my data quickly, and in a variety of ways, so I can glean as much info/insight from it as I can.

> Think outside the widgets.

If it's any consolation, I apply object based design principles as much as I can to my software written in my workhorse language, Fortran 90. :o)

paulv

--
Paul van Delst   A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP   Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545   And drinking largely sobers us again.
                                   Alexander Pope.