Subject: Re: widget layout
Posted by Mark Hadfield on Sun, 14 Jul 2002 21:49:04 GMT
View Forum Message <> Reply to Message

"Matt Feinstein" <mfein@clark.net> wrote in message
news:140720021309014241%mfein@clark.net...
> In article <3D30B1DA.E0041F96@yahoo.com>, Ted Cary
> <tedcary@yahoo.com> wrote:
>
>> Thanks, but that's what I used to do in this kind of situation.
>> The routine was conceived in the first place so that I would not
>> have to resort to storing widget geometry information in base
>> uvalues, since keeping track of such details during TLB resizing
>> can be a pain for all but the most simple GUIs. I was hoping for
>> something more general that worked along the lines of the
>> WIDGET_INFO function.
>
> I admit-- I'm new to IDL, so let me continue in the 'why doesn't the
> easy way work?' mode. Why can't you define a 'descendent' function
> of widget_info() that wraps around widget_info(), adds a keyword,
> and does what you want by looking at the base uvalue (where, I'm
> assuming, the appropriate state information is stored)?

Isn't it a pity that widgets are not objects? Then one could simply
subclass the widget base, add an appropriate tag to the class
structure and add corresponding keywords to the Init, GetProperty &
SetProperty methods.

Actually, several people have implemented the "widgets-as-objects"
idea. I have a widget-base class in my Motley library. It doesn't
currently store the row/column info, but it would be straightforward
to extend it so it did.

Final comment: if the base is a child then the base's UVALUE is not an
ideal place to store info relating to the base's own structure,
because the UVALUE may be needed by other widgets further up in the
heirarchy. In IDL 5.5 the RSI developers inadvertently provided a
solution: create an invisible, unrealised context menu as the first
child of the base and store data there!

--
Mark Hadfield   "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)