Subject: Re: Wrapper for WIDGET_DRAW
Posted by philaldis on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

On Tue, 29 Jun 1999 02:16:39 GMT, davidf7203@my-deja.com wrote:

> Phil Aldis writes:
>
>> In this object/direct graphics thing that I'm doing, I've created all
>> my own direct graphics destination objects. The window object takes a
>> draw widget ID and is then bound to that. I would really like it
>> though if I could have a wrapper for WIDGET_DRAW, whereby if
>> GRPHICS_LEVEL is set to 0, then direct graphics and GET_VALUE returns
>> a window number, 1, object graphics and GET_VALUE retruns an
>> IDLgrWindow object and 2, obj/direct graphics and GET_VALUE returns an
>> DERAgrWindow object.
>>
>> The problems I am having is that the user still wants to be able to
>> use Widget_Control to what they think is a draw widget. They don't
>> want what they think is a draw widget to be in fact a base. I can't
>> think of any way to store the object I create and the information
>> about it.
>>
>> Can anyone think of a way to do this?
>
> I don't see the problem here. A draw widget is a draw widget whether
> is in direct graphics or object graphics. The same keywords still
> apply and you still interact with it in the same way. The "thingy"
> (my own object-oriented terminology, what do you think?) that you
> draw into is the only thing that is different and is set by the
> GRAPHICS_LEVEL keyword.
>
> Cheers,
>
> David
>
>
> Sent via Deja.com http://www.deja.com/
> Share what you know. Learn what you don't.

I didn't make myself clear.Here's an example of how I would like it to
work:

```
 ;Create a draw widget
drawID = DERA_Draw(parent, XSIZE=400, YSIZE=400, GRAPHICS_LEVEL = 2)
 ;Get its value which will be a window object of class
 ;DERAgrWindow
Widget_Control, drawID, GET_VALUE=thingyWindow
```

```
 ;Now draw a view into it
thingyWindow->draw, thisView
```

That's what I would like, so obviously you could have a compound widget, with a FUNC_GET_VALUE, which returned either the window number, the IDLgrWindow class or the DERAgrWindow class.

However if its a compound widget and the top of the compund widget is a base, then when the user does

```
Widget_Control, drawID, /MOTION_EVENTS
```

......they will be a little surprised to get an error mesage along the lines of

```
Keyword MOTION_EVENTS not allowed in call to Widget_Control for a base
widget
```

However if I have just a draw widget then I have no way to store what type it is, except by using the UVALUE. However, if the UVALUE was set to something different by the user using Widget_Control, then that would be game over for my widget.

I hope that that explains it better.

Cheers,
Phil