

---

Subject: Re: Wrapper for WIDGET\_DRAW

Posted by [Pavel Romashkin](#) on Tue, 29 Jun 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Phil,

I see what you are saying. Maybe the following will work:

- a) Make the DERA\_Draw return the ID of the draw widget, so that no error occurs if the user calls "Widget\_Control, drawID, /MOTION\_EVENTS" - this will then work fine.
- b) Create compound widget, but place some other widget (a table) hiding under the DRAW widget.
- c) Use that other widget's VALUE and/or UVALUE through PRO\_SET\_VALUE and FUNC\_GET\_VALUE to store and retrieve the DRAW widget type. This way, a structure may be placed into VALUE and anything - into UVALUE.

Another way - use the UVALUE of the top base of compound widget (the true parent of the drawID) to store DRAW type, again using PRO\_SET\_VALUE and FUNC\_GET\_VALUE. This UVALUE is not going to be accessible otherwise because the ID of the base is not known to the user: DERA\_Draw returns the ID of the draw widget. Compared to the first way, you lose one location for storing information (VALUE), but you still have UVALUE which is sufficient. Don't let PRO\_SET\_VALUE modify the stored information; let DERA\_Draw do it.

Either way, "widget\_control, drawID, set\_uvalue='arbitrary' " will not cause loss of DRAW type information, and all keywords to that drawID that are valid for the draw widget will work ok. The flaw I see is that cleanup must be well designed, should a pointer be placed into UVALUES of the base or table, because these widgets' IDs and even their existence are not obvious.

Good luck,

Pavel

```
> ;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 compound widget is
> a base, then when the user does
>
> Widget_Control, drawID, /MOTION_EVENTS
```

>  
> .....they will be a little surprised to get an error message 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.

---