
Subject: IDL widgets: comments and questions

Posted by [knight](#) on Wed, 15 Mar 1995 02:30:57 GMT

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

I have recently written a widget-based image viewer and have some comments about IDL widgets. I mention some problems or lack of features I wanted. If anybody knows how to solve them, please speak up.

1.

I wanted to add a title to each draw widget. The title keyword appears only in `widget_base`. I finally used the title keyword on my plots. I'd still like to add titles to other widgets.

2.

Control of the increment on the slider is alluded to (in The IDL Reference Guide) for Openwindows but appears to not be available under Motif. I wanted to bump a slider value by 1 by pressing the left mouse button, but, whenever the slider range was large, the increment was also large. Does anybody know how to set the increment on a slider? I know you can do it with Motif widgets.

3.

I wanted to change the limits on a slider after it was realized. It doesn't seem possible.

4.

As somebody mentioned recently, it would be nice to be able to know what mouse button was pushed on an event. The `widget_draw` allows this with the `button_events` keyword, and I used it with success. Per the recent suggestions, one could use the different mouse clicks to access help or do alternative actions for the current widget.

5.

On some occasions drawing a new image on a `widget_draw` window was funny. Before drawing the new image, the window was erased and the previous image was redrawn. Then the new image was put up. This was particularly noticeable when the display was a `show3` or `shade_surf`. I don't understand the circumstances here; it may be due to changing windows (with `wset`). Has anybody seen this?

6.

I wanted to make some up/down or left/right arrows for incrementing and decrementing quantities or up/down/left/right arrows for moving about on an image. I tried a bitmap for a button but there seemed to be no control over the actual width or height of the button even though I specified a 16x16 pattern.

Are there ways to build bitmaps that take on a specified size. I really haven't experimented here. I wonder if anybody else has.

7.

Some recent comments have praised the availability of the uvalue. I, too, use it for controlling actions in widget_control. However, I still use common blocks to pass most information. You can't pass the id of a widget via the uvalue, and I need ids whenever I call widget_control for set_value or get_value. So common still seems necessary, even though most information could be passed in the uvalue.

I'd appreciate answers to any of the questions above.

Thanks,
Fred
Fred Knight (knight@ll.mit.edu)

--

=Fred Knight (knight@ll.mit.edu) (617) 981-2027
C-483\\MIT Lincoln Laboratory\\244 Wood Street\\Lexington, MA 02173

Subject: Re: IDL widgets: comments and questions
Posted by [steinhh](#) on Wed, 15 Mar 1995 16:05:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'll only comment on the points that I'm qualified for:

In article <3k5jh1\$7rb@testnews.ll.mit.edu>, knight@ll.mit.edu (Fred Knight) writes:

|> 1.

|> I wanted to add a title to each draw widget. The title keyword appears only
|> in widget_base. I finally used the title keyword on my plots. I'd still
|> like to add titles to other widgets.

Well, you could always have a Widget_LABEL centered on top of any draw window you like. You could even select a special font to make sure it looks like a "title". It would not be erased by erasing the plot, but you could change it whenever you like.

|> 7.

|> Some recent comments have praised the availability of the uvalue. I, too,
|> use it for controlling actions in widget_control. However, I still use
|> common blocks to pass most information. You can't pass the id of a widget
|> via the uvalue, and I need ids whenever I call widget_control for set_value
|> or get_value. So common still seems necessary, even though most information
|> could be passed in the uvalue.

|>

If you set it up correctly using a tree structure, you only need one widget ID in order to find your information no matter how complex your application is. The places where you have a legitimate need for that ID is inside the event handler (or routines called by the event handler). And in the event handler, you do have EVENT.TOP (and even EVENT.HANDLER/EVENT.ID!). This, in combination with structure variables as UVALUES, removes any need for common blocks in order to pass data anywhere.

I'm still using some common blocks in my programs, but those are only there to insure that information is never lost no matter how the application crashes. Still, with the NO_COPY keyword, no data (of any proportion) is ever stored twice. I never use common blocks in order to pass information.

Stein Vidar

Subject: Re: IDL widgets: comments and questions
Posted by [cannon](#) on Fri, 17 Mar 1995 09:27:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3k7396\$6qi@hermod.uio.no>, steinh@amon.uio.no (Stein Vidar Hagfors Haugan) writes:
|>

|> If you set it up correctly using a tree structure, you only need
|> one widget ID in order to find your information no matter how complex
|> your application is. The places where you have a legitimate
|> need for that ID is inside the event handler (or routines called by the
|> event handler). And in the event handler, you do have EVENT.TOP
|> (and even EVENT.HANDLER/EVENT.ID!). This, in combination with structure
|> variables as UVALUES, removes any need for common blocks in order
|> to pass data anywhere.
|>
.... cut

I'm trying to adopt your philosophy, but how do you handle a function which uses a widget to return a value (like the pickfile routine say, which uses a common block)? You call the function which creates the widgets and it is all fine until you want to exit, with the value you want the function to return in a uvalue somewhere. But to exit you have to destroy the widgets so control gets handed back to the function which made them - and then it cant get at the uvalue any longer to return the thing you want??

The only examples i've seen use common blocks, but is there any other way round it?

Robert Cannon

Subject: Re: IDL widgets: comments and questions
Posted by [rep2857](#) on Mon, 20 Mar 1995 17:45:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3k5jh1\$7rb@testnews.ll.mit.edu>,
Fred Knight <knight@ll.mit.edu> wrote:

>

> 3.

> I wanted to change the limits on a slider after it was realized. It doesn't
> seem possible.

The following will work:

Widget_Control, slider_name, Set_Slider_Max=max_value

Where slider_name is the name of the slider widget to be changed, and
max_value is the new maximum value to be set.

> 5.

> On some occasions drawing a new image on a widget_draw window was funny.
> Before drawing the new image, the window was erased and the previous image
> was redrawn. Then the new image was put up. This was particularly
> noticeable when the display was a show3 or shade_surf. I don't understand
> the circumstances here; it may be due to changing windows (with wset). Has
> anybody seen this?

I don't recall seeing this in several applications I've created. However,
"erase, 0" ought to take care of the problem.

> 7.

> Some recent comments have praised the availability of the uvalue. I, too,
> use it for controlling actions in widget_control. However, I still use
> common blocks to pass most information. You can't pass the id of a widget
> via the uvalue, and I need ids whenever I call widget_control for set_value
> or get_value. So common still seems necessary, even though most information
> could be passed in the uvalue.

You can do without COMMON blocks completely in your applications and
pass around the same information with easier traceability and less
confusion using UValues. Unfortunately, the explanation is not easily

handled by email or posts. I think the best way to learn about this is by taking one of RSI's training courses. RSI teaches the technique during their Widget Programming classes and use several example programs to illustrate it. Even so, I had to go back and review what happened during class to make good use of it in my first applications afterwards. They offer good example programs, and relevant real-world knowledge of the language. I was very impressed with the information that I gained from the class and the knowledge of the instructors teaching it (Hi David F. and Molly).

Mike Schienle Hughes Santa Barbara Research Center
rep2857@sbsun0010.sbrc.hac.com 75 Coromar Drive, M/S B28/87
Voice: (805)562-7466 Fax: (805)562-7881 Goleta, CA 93117

Subject: Re: IDL widgets: comments and questions
Posted by [Fergus Gallagher](#) on Fri, 24 Mar 1995 09:52:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

>
> -----
> Subject: IDL widgets: comments and questions
> Date: 15 Mar 1995 02:30:57 GMT
> From: knight@ll.mit.edu (Fred Knight)
> Organization: MIT Lincoln Laboratory
> Newsgroups: comp.lang.idl-pvwave
>
> -----
>
> I have recently written a widget-based image viewer and have some comments
> about IDL widgets. I mention some problems or lack of features I wanted.
> If anybody knows how to solve them, please speak up.
>
> 1.
> I wanted to add a title to each draw widget. The title keyword appears only
> in widget_base. I finally used the title keyword on my plots. I'd still
> like to add titles to other widgets.
>
> -----
>

Put the draw widget in its own base:

Instead of

```
base=widget_base(/column) ; say  
draw=widget_draw(base,.....)
```

do

```
base=widget_base(/column)
draw_base=widget_base(base,title='Draw Widget',/column)
draw=widget_draw(draw_base,.....)
```

Fergus

```
=====
| Fergus Gallagher          |
| Remote Sensing Applications Development Unit |
| British National Space Centre      |
| Monks Wood                      |
| Huntingdon PE17 2LS / UK          |
|                               |
| F.Gallagher@nerc.ac.uk          |
| http://uh.nmt.ac.uk/bnsc/fgg.html |
=====
```
