
Subject: Re: Dynamically adding and removing widgets
Posted by [Antonio Santiago](#) on Wed, 23 Nov 2005 17:58:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
> 4. One of my event handlers calls a procedure called
> show_phenomenon
>
> pro event_handler Event
> ----
>   WIDGET_CONTROL,Event.TOP,SET_UVALUE=state,/NO_COPY
>   show_phenomenon,,,,,,,,,Event
>   ----
> end
>
```

I think your problem is with the /NO_COPY keyword, this undefines the "state".
See PDF manual:

"However, it has the side effect of causing the source variable to become undefined. On a "set" operation (using the SET_UVALUE keyword to WIDGET_CONTROL), the variable passed as value becomes undefined. On a "get" operation (GET_UVALUE keyword to WIDGET_CONTROL), the user value of the widget in question becomes undefined."

Then inside your "Show_phenomenon" procedure the "state" value is undefined.

Bye :)

--

Subject: Re: Dynamically adding and removing widgets
Posted by [akkiraju](#) on Wed, 23 Nov 2005 18:29:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

hi Antonio,

Thanks. I have cross checked my sate variable value in the "show_phenomenon" procedure.

```
IDL> print, base
```

```
mask_tab
```

```
IDL> print,state
```

```
{      7      8      6      20      21
16      17      18}
```

state is not undefined in my case, I can access the components and their names from the show_phenomenon procedure. But when I use these names to add new widgets it doesn't work, that's the real problem.

Subject: Re: Dynamically adding and removing widgets
Posted by [David Fanning](#) on Wed, 23 Nov 2005 19:23:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

akkiraju writes:

```
>      Thanks. I have cross checked my state variable value in the
> "show_phenomenon" procedure.
> IDL> print, base
> mask_tab
> IDL> print, state
> {      7      8      6      20      21
> 16      17      18}
>
> state is not undefined in my case, I can access the components and
> their names from the show_phenomenon procedure. But when I use these
> names to add new widgets it doesn't work, that's the real problem.
```

I agree with Antonio. When you get into Show_Phenomenon and look for your state variable in the UValue of the top-level base it is not defined. It can't be.

I'm not sure how you think you are checking it, but I am 100% sure that what you did is not what is happening in your program. I would stop the program just before the Show_Phenomenon call, then step into the Show_Phenomenon code. You will see that state is undefined there.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Dynamically adding and removing widgets
Posted by peter.albert@gmx.de on Thu, 24 Nov 2005 07:16:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

well, I'd guess that the state variable **does** exist in the routine "show_phenomenon":

In "pro event_handler", state is assigned to the uvalue of the parent widget :
WIDGET_CONTROL,Event.TOP,SET_UVALUE=state,/NO_COPY

It is correct that the variable "state" itself is undefined afterwards, but it does still exist as the UVALUE of the root widget. Then the routine "show_phenomenon" is called with the "event" variable as a parameter, which is fine, as "event" also still exists. Especially, event.top still points at the parent widget. Therefore, when calling

widget_control,Event.Top,GET_UVALUE=state

within "show_phenomenon", everything works as expected and the "state" is read from the parent widget's uvalue.

So where is the problem coming from?

The real problem is with the "base" variable:

You use

```
mask_disp = WIDGET_DRAW(base,xsize=numCols,ysize=numRows)
```

where "base" is expected to be a valid widget ID. As you showed yourself, however, "base" holds the string "mask_tab":

```
>> IDL> print, base  
>> mask_tab
```

Why?

Because in the line above you use

```
base = widget_info(state.mask_tab,/UNAME)
```

Using /UNAME with WIDGET_INFO returns the user name of the widget.

So probably

```
mask_disp = WIDGET_DRAW(state.mask_tab, xsize=numCols,ysize=numRows)
```

is just what you want?

It will create the new draw widget right in the widget whose ID is
state.mask_tab.

Cheers,

Peter
