
Subject: Re: returning values from widget hierarchy
Posted by [Liam E. Gumley](#) on Fri, 19 May 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Vince Hradil wrote:

```
> First, forgive my ignorance of widget programming ;^)
>
> How can I return values from a "temporary" widget hierarchy. That is:
>
> I have a widget program that creates a new widget for users to adjust
> color settings. When they select a menu item, I do something like,
>
> ...build widgets for color settings...
> widget_control, cs_base, /realize
> xmanager, 'color_settings', cs_base
>
> Then in color_settings_event I handle the various sliders and buttons in
> the cs_base widget. Then when the user clicks on 'Accept' on the
> cs_base widget, I do widget_control, event.top, /destroy
>
> What I'd like to do is NOW be able to see what those settings are back
> in the procedure that created the cs_base in the first place. I would
> like to avoid using COMMON blocks if possible. I'm familiar with using
> widget_control to store a structure to contain all the information
> needed, but can't seem to figure out how to make it work in this
> instance.
```

Here's a minimal example which returns a user's name. Note the use of a pointer to store the information structure in the user value of the top level base.

```
;---
PRO WGETNAME_EVENT, EVENT

;- Get state information
widget_control, event.top, get_uvalue=infoptr
info = *infoptr

;- Identify widget which caused the event
widget_control, event.id, get_uvalue=uvalue

;- Handle the widget events
if (uvalue eq 'Text') or (uvalue eq 'OK') then begin
  widget_control, info.text, get_value=name
  info.name = name
  info.status = 'OK'
endif else begin
  info.name = "
```

```

    info.status = 'Cancel'
endelse

;- Destroy the widget hierarchy
widget_control, event.top, /destroy

;- Save state information
*infoptr = info

END

PRO WGETNAME, NAME, STATUS

;- Create top level base
tlb = widget_base(column=1, xoffset=400, yoffset=300, $
    title='Name', tlb_frame_attr=1)

;- Create text entry widgets
tbase = widget_base(tlb, row=1)
label = widget_label(tbase, value='Enter your name: ')
text = widget_text(tbase, uvalue='Text', /editable, xsize=30)

;- Create button widgets
bbase = widget_base(tlb, row=1, /align_center)
bsize = 75
butta = widget_button(bbase, value='OK', uvalue='OK', xsize=bsize)
buttb = widget_button(bbase, value='Cancel', uvalue='Cancel',
xsize=bsize)

;- Realize widgets
widget_control, tlb, /realize

;- Create and store state information via a pointer
info = {text:text, name:"", status:'Cancel'}
infoptr = ptr_new(info)
widget_control, tlb, set_uvalue=infoptr

;- Manage events
xmanager, 'wgetname', tlb

;- Get state information
info = *infoptr
ptr_free, infoptr

;- Get return variables
name = info.name
status = info.status

```

END

;---

When I execute this procedure, type my name in the dialog box, and click 'OK':

```
IDL> wgetname, name, status
IDL> help, name, status
NAME      STRING   = 'Liam Gumley'
STATUS    STRING   = 'OK'
```

Cheers,
Liam.
<http://cimss.ssec.wisc.edu/~gumley>

Subject: Re: returning values from widget hierarchy
Posted by [Vince Hradil](#) on Fri, 19 May 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sorry about the copies. Having some troubles with 'Preferences' this morning.

Vince Hradil wrote:

```
> First, forgive my ignorance of widget programming ;^)
>
> How can I return values from a "temporary" widget hierarchy. That is:
>
> I have a widget program that creates a new widget for users to adjust
> color settings. When they select a menu item, I do something like,
>
> ...build widgets for color settings...
> widget_control, cs_base, /realize
> xmanager, 'color_settings', cs_base
>
> Then in color_settings_event I handle the various sliders and buttons in
> the cs_base widget. Then when the user clicks on 'Accept' on the
> cs_base widget, I do widget_control, event.top, /destroy
>
> What I'd like to do is NOW be able to see what those settings are back
> in the procedure that created the cs_base in the first place. I would
> like to avoid using COMMON blocks if possible. I'm familiar with using
> widget_control to store a structure to contain all the information
> needed, but can't seem to figure out how to make it work in this
> instance.
>
> IDL> print, !version
```

> { sparc sunos unix 5.1 Apr 13 1998}
>
> Thanks,
>
> Vince
