

---

Subject: Re: Linking buttons to plot windows

Posted by [Robert Barnett](#) on Wed, 22 Dec 2004 23:09:49 GMT

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

---

And look ... it's possible to write an entire application demonstrating this in little over half the code it took to write the event loop for GUI builder. Apologies, the excess documentation makes it a little hard to read.

David Fanning wrote:

> Clive Cook writes:

>  
>

>> Thanks David, i'm pretty new to widget programming so i am not really sure  
>> what to do. This is an example of the code that the GUIbuilder generates. I  
>> assume that the first part of the code (WID\_BASE\_0\_event) handles the  
>> events, so when i press wid\_button\_1 the function contour1 is called. As i  
>> mentioned before this plots in the last area designated for plots  
>> (wid\_draw\_1). How can i specifically plot it in the first area (wid\_draw\_0)?  
>>

>  
> It is my humble (and mostly ignored) opinion that it  
> is \*impossible\* to use the GUI\_BUILDER to create widget  
> programs. At least programs you can understand. And this  
> code is like running your fingernails across the blackboard  
> to me. But, your code will look something like this (assuming  
> the CONTOUR event handlers aren't doing something weird):

>  
> pro WID\_BASE\_0\_event, Event  
>  
> wWidget = Event.top  
>  
> case Event.id of  
>  
> Widget\_Info(wWidget, FIND\_BY\_UNAME='wid\_button\_0'): begin  
> if( Tag\_Names(Event, /STRUCTURE\_NAME) eq 'WIDGET\_BUTTON' )then \$  
> BEGIN  
> drawID = WIDGET\_INFO(wWidget, FIND\_BY\_UNAME='WID\_DRAW\_O')  
> Widget\_Control, drawID, Get\_Value=wid  
> WSET, wid  
> contour1, Event  
> ENDIF  
> end  
> Widget\_Info(wWidget, FIND\_BY\_UNAME='WID\_BUTTON\_1'): begin  
> if( Tag\_Names(Event, /STRUCTURE\_NAME) eq 'WIDGET\_BUTTON' )then \$

```
> BEGIN
> drawID = WIDGET_INFO(wWidget, FIND_BY_UNAME='WID_DRAW_1')
> Widget_Control, drawID, Get_Value=wid
> WSET, wid
> contour2, Event
> ENDIF
> end
> else:
> endcase
>
> end
>
> If that is not the ugliest code I've ever posted in a newsgroup,
> I don't know what is. :-(
>
> Cheers,
>
> David
>
> P.S. Really do yourself a favor and ditch the GUI_BUILDER. You
> (and your boss) will be glad you did. :-)
>
```

--

nrb@  
Robbie Barnett  
imag  
Research Assistant  
wsahs  
Nuclear Medicine & Ultrasound  
nsw  
Westmead Hospital  
gov  
Sydney Australia  
au  
+61 2 9845 7223

```
; Robbies Tools (rt)
; Plot brown noise
; Present a "redraw" button to redraw the plot in the given window
; This code has absolutely now warranty at all and is not guaranteed
; to actually work.
; Written by Robbie Barnett, Westmead Hospital Sydney

; Draw the graph of the brown noise in the specified window
```

```

pro rt_brownian_draw, window
; Keep the seed as a global variable between calls
common rt_brownian, seed
; Set the window to draw in
wset, window
; Create the array which contains the values of the brown noise
values = lonarr(20)
for i=1,19 do values[i] = values[i] + (randomu(seed)-0.5)*10.0
; Plot the brown noise
plot, values
end

; Process an event generated from the widget_button
pro rt_brownian_event, ev
; Get the draw_id which has been packed in the user value of the
; widget button. If this problem were more complicated I would have
; packed a pointer to a structure or object.
widget_control, ev.id, GET_UVALUE=draw_id
; Get the window id which maps to the desired the draw_widget
widget_control, draw_id, GET_VALUE=window
; Draw onto the desired window
rt_brownian_draw, window
end

; Create a widget which allows the user to redraw a plot of brown
; noise.
function rt_brownian, PARENT=parent
; Create the "top level base" widget. This lays out the widget_button
; and widget_draw in a column. Allow for the possibility that the
; parent was not specified.
if (keyword_set(parent)) then tlb = widget_base(parent,COLUMN=1) $
else tlb = widget_base(COLUMN=1)
; Create the canvas (window) for drawing on to
draw_id = widget_draw(tlb,XSIZE=300,YSIZE=300)
; Create the button
button_id = widget_button(tlb,VALUE="redraw",UVALUE=draw_id)
; Realize the widget elements
widget_control, tlb, /realize
; Register for events
xmanager, "rt_brownian", button_id, /JUST_REG
; Return an id for referencing to this widget
return, tlb
end

; Create 3 widgets arranged in a row. Each widget is independant and
; allows the user to redraw a plot of brown noise in each
pro rt_brownians
; Create the parent widget for holding all of them

```

```
parent = widget_base(ROW=1)
; Create the independant plots arranged in a single row
a = rt_brownian(PARENT=parent)
b = rt_brownian(PARENT=parent)
c = rt_brownian(PARENT=parent)
end
```

## File Attachments

---

1) [rt\\_brownians.pro](#), downloaded 88 times

---