
Subject: Re: Passing info and destroying widgets...
Posted by [R.Bauer](#) on Tue, 22 Jun 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Liam Gumley wrote:

```
> "Robert S. Mallozzi" wrote:
>> I believe you must use XMANAGER in blocking mode for
>> this technique to work.
>
> Here's an example which works in non-blocking mode:
>
> ;---cut here---
> PRO TEST_EVENT, EVENT
>
> ;- Get pointer from top level base, then the info structure
>
> widget_control, event.top, get_uvalue=ptr
> info = *ptr
>
> ;- Handle the widget which caused this event
>
> widget_control, event.id, get_uvalue=name
> case 1 of
>   name eq 'Button 1' or name eq 'Button 2' : info.name = name
>   else : widget_control, event.top, /destroy
> endcase
>
> ;- Update the info structure
>
> *ptr = info
>
> END
>
>
```

I agree to this method because sometimes I have to handle a lot of MBytes of data.

But in difference to your solution I am using a string in the tagname to identify pointers. This gives me the possibility to destroy the pointers before I destroy the widget. I have posted the routine `get_tagname` as attachment.

```
free_ptr=get_tagname(map,'PTR*')
FOR i=0,N_TAGS(free_ptr)-1 DO PTR_FREE,free_ptr.(i)

WIDGET_CONTROL,map.base_0,/destroy
```

The number of pointers are not unlimited, so they should carefully be destroyed after usage. If you don't free pointers you are caught sometimes later not knowing which to destroy.

Another important difference is that I don't copy the pointer into a normal variable. All changes are done directly on the pointer.

e.g.:

```
(*map.ptr_drawids)[i]=drawid
```

R.Bauer

```
;  
; Copyright (c) 1998, Forschungszentrum Juelich GmbH ICG-1  
; All rights reserved.  
; Unauthorized reproduction prohibited.  
; This software may be used, copied, or redistributed as long as it is not  
; sold and this copyright notice is reproduced on each copy made. This  
; routine is provided as is without any express or implied warranties  
; whatsoever.  
;  
;+  
; NAME:  
; get_tagname  
;  
; PURPOSE:  
; This function generates a new structure from a given structure by a search string  
;  
;  
; CATEGORY:  
; PROG_TOOLS/STRUCTURES  
;  
; CALLING SEQUENCE:  
; Result=get_tagname(structure,search_string,[without=without] )  
;  
; INPUTS:  
; structure: The structure which should be scanned
```

```

; search_string:   The string which should be used for searching in the structure
;
; OPTIONAL INPUTS:
; without:   A string which should not included in the result
;
; OUTPUTS:
; The return value is a structure
; defined by the search parameter
;
; EXAMPLE:
; Result=get_tagname(inhalt,'pi*')
; gives back all tags starting with pi
;
; ** Structure <11867f8>, 1 tags, length=8, refs=1:
;   PI_NAME      STRING  'Peter Mustermann'
;
; Result=get_tagname(inhalt,'*name')
; gives back all tags ending with name
;
; ** Structure <1186108>, 4 tags, length=32, refs=1:
;   PI_NAME      STRING  'Peter Mustermann'
;   NAME         STRING  'sin0'
;   PARAM_LONG_NAME STRING  'SIN(X)'
;   TIME_LONG_NAME STRING  'time'
;
; Result=get_tagname(inhalt,'*name*')
; gives back all tags including name
;
; ** Structure <1171ce8>, 5 tags, length=112, refs=1:
;   PI_NAME      STRING  'Peter Mustermann'
;   GPARAM_NAMELIST STRING  Array[10]
;   NAME         STRING  'sin0'
;   PARAM_LONG_NAME STRING  'SIN(X)'
;   TIME_LONG_NAME STRING  'time'
;
; Result=get_tagname(inhalt,'*name*',without='gparam_namelist' )
; gives back all tags including name without gparam_namelist
;
; ** Structure <1180108>, 4 tags, length=32, refs=1:
;   PI_NAME      STRING  'Peter Mustermann'
;   NAME         STRING  'sin0'
;   PARAM_LONG_NAME STRING  'SIN(X)'
;   TIME_LONG_NAME STRING  'time'
;
; Result=get_tagname(inhalt,'name')
; gives back the tag name named 'name'
;
; ** Structure <118ab68>, 1 tags, length=8, refs=1:

```

```

; NAME      STRING  'sin0'
;
;
;
; MODIFICATION HISTORY:
; Written by: R.Bauer (ICG-1), 1998-07-30
;-

```

```

FUNCTION get_tagname, struct,search_string,$
  without = without

```

```

IF N_PARAMS() LT 2 THEN BEGIN
  MESSAGE,'result=get_tagname(struct,search_string)'/info
  RETURN,' '
ENDIF

```

```

n_search_string = N_ELEMENTS(search_string)-1
FOR o=0,n_search_string DO BEGIN
  search = (search_string[o])(0)

```

```

IF STRPOS(search,'*') EQ -1 THEN BEGIN
  only=1
  begining=1 ; removed ;nn
  tag_such=search
ENDIF

```

```

IF STRPOS(search,'*') GT 0 THEN BEGIN
  begining=1
  only=0
  tag_such=STRMID(search,0,STRPOS(search,'*'))
ENDIF

```

```

IF STRPOS(search,'*') EQ 0 THEN BEGIN
  begining=0
  only=0
  IF STRPOS(reverse_text(search),'*') NE 0 THEN BEGIN
    tag_such=STRMID(search,1,STRLEN(search)-1)
    ende=1
  ENDIF ELSE BEGIN
    ende=0
    tag_such=STRMID(search,1,STRLEN(search)-2)
  ENDELSE
ENDIF
CATCH,errvar

```

```

;if errvar ne 0 then goto , help

tag_such=STRUPCASE(tag_such)

tags=TAG_NAMES(struct)

tcount= STRPOS(tags, tag_such)
count=WHERE(tcount NE -1,m)

IF m EQ 0 THEN RETURN,-1

IF KEYWORD_SET(begining) THEN BEGIN
  count = WHERE(tcount EQ 0,m)
  IF m EQ 0 THEN RETURN,-1
ENDIF

IF KEYWORD_SET(only) THEN BEGIN
  a=WHERE(tags(count) NE tag_such,zaehler)
  IF zaehler GT 0 THEN BEGIN
    count(a)=-1
    mcount=WHERE(count NE -1,m)
    IF m EQ 0 THEN RETURN,-1
    count=count(mcount)
  ENDIF
ENDIF

IF KEYWORD_SET(ende) THEN BEGIN
  en=WHERE(STRPOS(reverse_string(tags[count]),reverse_string(tag_such)) EQ
0,count_en)
  IF count_en GT 0 THEN count=count[en] ELSE RETURN,-1

ENDIF

IF N_ELEMENTS(without) GT 0 THEN BEGIN
  without=STRUPCASE(without)
  a=WHERE(tags(count) EQ without,zaehler)
  IF zaehler GT 0 THEN BEGIN
    count(a)=-1
    mcount=WHERE(count NE -1,m)
    count=count(mcount)
  ENDIF
ENDIF

IF m GT 0 THEN BEGIN

  keyws=tags(count)

```

```
FOR i=0 , N_ELEMENTS(count)-1 DO $
  IF N_ELEMENTS(antwort) EQ 0 THEN
antwort=CREATE_STRUCT(keyws[i],struct.(count[i])) ELSE $
  antwort=CREATE_STRUCT(antwort,keyws(i),struct.(count[i]))
ENDIF
```

```
ENDFOR
IF N_ELEMENTS(antwort) EQ 0 THEN BEGIN
  MESSAGE,'nothing found like '+tag_such,/cont
  RETURN,antwort
ENDIF
RETURN, antwort
```

END

File Attachments

1) [get_tagname.pro](#), downloaded 120 times
