
Subject: Re: universal structure editor progress
Posted by [Martin Schultz](#) on Thu, 29 Apr 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dyer Lytle wrote:

>
> I'll attach
> what I have so far to this post and see what happens! Currently,
> all this is is an attempt at a Universal Structure Display (USD) and
> it even fails to that properly in many situations. However, it DOES
> do enough for me to look at stuff in my big nested structures, which
> is what I wanted to do.

if that's the new policy ;-) then let me publicize what I have done as well. Guess it doesn't even go that far. Just creates a widget that displays every item of a structure and adds a button which is intended to show a "pop-up" widget for sub-structures or arrays (they would otherwise take too much space). Maybe most important is the pointer feature: pointers are de-referenced (up to 20 times) and listed as value (the name being preceded by '*'s). Still needs A LOT of work of course, so maybe next decade ;-)

Regards,
Martin.

PS: the most meaningful name bla.pro denotes that this is a preprepre-alpha release ;-)

|||||\\\-----//|
Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/
pro bla1,arg,group_leader=group_leader

```
; would be nice to print name of passed variable  
; instead of local variable name.  
; for now let's be happy with structure name
```

```
; if arg was not a structure, print a warning and return  
if (not chkstru(arg)) then begin  
    message,'Argument is not a structure!','/Continue  
    return  
endif
```

```
; make temporary working copy of argument  
tmp = arg
```

```

SName = Tag_Names(tmp,/structure_name)
if (SName eq "") then SName = '<anonymous>'

print,SName

TNames = Tag_Names(tmp)

ntags = N_Tags(tmp)
TType = intarr(ntags)
TNDim = intarr(ntags)
for i = 0,ntags-1 do begin
    value = tmp.(i)
    ptr_recursion_level = 0
ptr_reentry:
    if (ptr_recursion_level gt 5) then $
        stop,'Weird pointer detected: Recursion depth > 5!'

    TType[i] = size( value, /type )
    TNDim[i] = size( value, /N_Dimensions )
    ; treat arrays with 1 value as scalars
    SumDim = iproduct( size( value, /Dimensions ) )
    if (SumDim eq 1) then $
        TNDim[i] = 0
    ; treat sub-structures as arrays
    if (TType[i] eq 8) then $
        TNDim[i] = 1

    ; some extra precaution for pointers:
    ; - set dimensions according to de-referenced value
    ; - add * to tag name
    ; only treat scalar pointers here
    if (TNDim[i] eq 0 AND TType[i] eq 10) then begin
        if (ptr_valid(value)) then begin
            value = *value
            TNames[i] = '*' + TNames[i]
            goto,ptr_reentry
        endif else begin
            ; value = '<nil pointer>'
            TType[i] = -10
        endelse
    endif

endfor

; array with widget IDs (first 3 are reserved for OK, Apply, Cancel)
ids = lonarr(ntags+3)

```

```

; create base widget
if (!d.name eq 'X' OR !d.name eq 'WIN' OR !d.name eq 'MAC') then $
  device,get_screen_size=ScreenSize $
else $
  Screen_Size = [640, 480 ] ; be conservative
BaseWidgetId = widget_base(/COLUMN, $
  xoffset=0.35*ScreenSize[0],yoffset=0.25*ScreenSize[1], $
  title='Widget Editor (+SName+)', $
  Group_Leader=Group_Leader, /SCROLL )

ww = widget_base(BaseWidgetID, /Column, Frame=2, /BASE_ALIGN_RIGHT, $
  XSize=280 )

; add fields
; - an edit field is added for scalar structure elements
; - a button is added for multidimensional fields or substructures
for i=0,ntags-1 do begin
  if (TNDim[i] eq 0) then begin
    noedit = 0 ; default is editable
    value = ( tmp.(i) )[0]
    ; de-reference pointers
    while (size( tmp.(i), /TYPE ) eq 10) do $
      if (ptr_valid(value)) then $
        value = *value $
      else begin
        value = '<nil pointer>'
        noedit=1
      endelse
    ; make a note on objects
    if (TType[i] eq 11) then begin
      value = '<object>'
      noedit = 1
    endif

    ids[i+3] = CW_Field( ww, title=TNames[i], value=value, $
      float=(TType[i] eq 4 OR TType[i] eq 5 OR TType[i] eq 6 $
        OR TType[i] eq 9), $
      integer=(TType[i] eq 1 OR TType[i] eq 2), $
      long = (TType[i] eq 3 OR TType[i] eq 12 OR TType[i] eq 13 $
        OR TType[i] eq 14 OR TType[i] eq 15), $
      noedit=noedit )

    endif else $ ; handle arrays etc.
    ids[i+3] = Widget_Button(ww, value=TNames[i]+' >> ')
  endif
endfor

```

```
buttonlist = widget_base(BaseWidgetId, /ROW)
ids[0] = Widget_Button(buttonlist, value=' Apply ')
ids[1] = Widget_Button(buttonlist, value=' OK ')
ids[2] = Widget_Button(buttonlist, value=' Cancel ')

print,'about to realize widget ..'
Widget_Control,BaseWidgetId,/Realize,Set_Uvalue=IDs
print,'widget should be visible now!'

wait,10

print,'destroy widget!'
Widget_Control,BaseWidgetId,/Destroy

end
```

```
function iproduct,x

res = 1L
for i=0,n_elements(x)-1 do $
res = res * x[i]

return,res
end
```

```
pro bla

test = { p:1, o:findgen(2), q:[1], r:fltarr(1,1,1) }
; bla1,test
bla1,!d

return
end
```

File Attachments

1) [bla.pro](#), downloaded 102 times
