
Subject: There is always a first !

Posted by [Sean Heukels](#) on Mon, 22 Jan 2001 15:30:48 GMT

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

Sean Heukels (sean77=cuthere=@dds.nl) writes:

- > The problem is that I am trying to pick up where
- > I left and it's very hard. I haven't touched IDL for over
- > a year now.
- > Is there someone who would share a widget program source
- > with me. (I'll exchange if necessary) that doesn't hang or act
- > obscure. i.e. a stable version.

:-)Widget programs are so hard to write that almost
:-)everyone keeps them under strict lock and key. Sorry. :-(

That's basically a shame then.
Maybe if someone would share, others could
view source and do better. Maybe even start a new
IDL widget newsgroup.

I always enjoyed making the widgets. It's something like
creating a whole new UI that other programs fail in.
A lot of people find it time consuming to start programming
widgets, because there is nowhere to start.
I don't mind sharing at all.
(--> See bottom for attachment. No attachments allowed in this NG)

Grrrr Seansan
sean@invivonmr.uu.nl

++++
When give, always get

Mr. Miyagi, Karate kid I
++++

***** FILE STARTS HERE *****

```
; filename : pulsgen.pro
; Date : 12/11/99
; Author : Sean Heukels
; Created with IDL 5.0 on Unix Solaris system
; Remarks It seems this program can only be run by starting
; the IDL UI enironment. "idlde on Unix"
; Concept :
; Define some variables, let the user input others,
```

```

; define an array for the x and Y coordinates and show
; the graph of a pre defined function.
; Modified :      04/11/99 Sean Heukels      Created a widget
base interfase
;
Original file was cal.pro [My own]
;      12/12/99 Sean Heukels      Used wi.pro
for pulsgen.pro [I think found in /examples]
;-----
--

```

```

PRO exitthis_EVENT, event

```

```

; Get the value posted for detection of buttons
WIDGET_CONTROL, event.id, GET_UVALUE = uval

```

```

; Read variable info through U_VALUE
WIDGET_CONTROL, event.top, GET_UVALUE=info, /NO_COPY

```

```

; set up case for destroying
; Added the var 'nee' so a simple enter wouldn.t kill the program
CASE uval OF

```

```

'exit' : BEGIN
    !P.MULTI = 0
    WIDGET_CONTROL, event.top, /DESTROY
    ENDCASE
'clear' : BEGIN
    WIDGET_CONTROL, info.drawwin, /DESTROY
    info.drawwin = WIDGET_DRAW(info.winbase, FRAME=0, SCR_XSIZE=500,
SCR_YSIZE=500, /ALIGN_CENTER)
    !P.MULTI = 0
    ENDCASE
'icon' : WIDGET_CONTROL, info.basein , ICONIFY=1
'multi' : !P.MULTI = [0, 2, 2]
'nomulti' : !P.MULTI = 0
'clearall' : BEGIN
    ; RETALL This clears the STRUCTURE info.
    CLOSE, /ALL
    HEAP_GC, /VERBOSE
    ; WIDGET_CONTROL, /RESET
    PRINT, "***** System cleared. *****"
    WAIT, 3
    PRINT, "If this doesn't help. Restart the program !!"
    ENDCASE
'restart' : BEGIN
    ; RETALL
    CLOSE, /ALL
    HEAP_GC, /VERBOSE

```

```

WIDGET_CONTROL, /RESET
PRINT, "Restarting ..."
pulsgen
ENDCASE
'nee' : RETURN
ENDCASE
IF (uval NE 'exit') AND (uval NE 'restart') THEN WIDGET_CONTROL, event.top,
SET_UVALUE=info

END

;-----
--
PRO windowval, event
; Read info
WIDGET_CONTROL, event.top, GET_UVALUE=info, /NO_COPY
; Read the PLOT or Oplot button
WIDGET_CONTROL, event.id, GET_UVALUE=plotter

; Read the title
;sinccos.pro
;Author: Sean Heukels
;Aim: Implementation of hyperbolic secant pulse
;
;Date: 15/12/1999
;=====
;PRO hypsecant
; Read variables
WIDGET_CONTROL, event.id, GET_UVALUE = uval
WIDGET_CONTROL, info.var1, GET_VALUE=np
WIDGET_CONTROL, info.var2, GET_VALUE=lobe
WIDGET_CONTROL, info.var3, GET_VALUE=saveas
; Return update
WIDGET_CONTROL, info.var4, SET_VALUE="Reading info ....."
WAIT,2

; set floats
np = fix(np[0])
lobe = fix(lobe[0])
saveas = string(saveas[0])
PI=acos(-1)
;Create empty arrays with variables depth defined with the number of points
;-----

phase = FLTARR(np+1)
halfnp = np/2
timeleft = findgen(halfnp)

```

```
timeleft = timeleft + 1
timeleft = -reverse(timeleft)
```

```
timeright = findgen(halfnp)
timeright = timeright + 1
```

```
totaltime = findgen(np + 1)
```

```
; Left part
for i = 0, halfnp-1 do begin
totaltime[i] = timeleft[i]
endfor
```

```
; Center
totaltime[halfnp] = 0
```

```
; Right part
for i = 0, halfnp-1 do begin
t = halfnp + 1 + i
totaltime[t] = timeright[i]
endfor
```

```
amp = fltarr(np + 1)
last = np
top = totaltime[last]
;lobe = lobe / 2
```

```
omreken = top/(lobe*PI)
totaltime[*] = totaltime[*] / omreken
```

```
for i = 0, np do begin
amp[i] = SIN(totaltime[i])/totaltime[i]
endfor
```

```
amp[halfnp] = 1
amp = amp * 1023
mult = HANNING(np + 1, ALPHA=0.50)
mult2 = HANNING(np + 1, ALPHA=0.56)
fltsinc = amp * mult
fltsinc2 = amp * mult2
phase(*)=180
phase=phase*(amp lt 0)
```

```
WIDGET_CONTROL, info.var4, SET_VALUE="Pulse created"
plot, amp, TITLE="Pulse"
```

```

WAIT, 2
WIDGET_CONTROL, info.var4, SET_VALUE="Hanning / Hamming filter"
plot, mult, TITLE="Filter"
WAIT, 2
WIDGET_CONTROL, info.var4, SET_VALUE="Filtered Pulse"
plot, filtsinc, TITLE="Filtered Pulse"
WAIT, 1

; Return update
WIDGET_CONTROL, info.var4, SET_VALUE="Arrays created ....."
WAIT, 1
WIDGET_CONTROL, info.var4, SET_VALUE="Plotting ....."
WAIT, 1

WIDGET_CONTROL, info.var4, SET_VALUE="Plot succesfull"
WAIT,2

IF (uval eq 'save') THEN BEGIN
WIDGET_CONTROL, info.var4, SET_VALUE="Reading save info ....."
WAIT,1

IF (saveas eq "") THEN BEGIN
dee = DIALOG_MESSAGE("No filename supplied", TITLE="Filename mistake",
DIALOG_PARENT="base1")
WIDGET_CONTROL, info.var4, SET_VALUE="No Save file as value described"
return
ENDIF

;Write pulse to ASCII file
;-----
fname = "
fname = saveas
path = '~/vnmrsys/shapelib/'
rf_fname = path + fname + '.RF'
rf_text = 'Save file as ' + rf_fname + '?'
doit = DIALOG_MESSAGE(rf_text, /QUESTION, TITLE="Save new puls ?",
DIALOG_PARENT="base1")

IF (doit eq 'Yes') THEN BEGIN

WIDGET_CONTROL, info.var4, SET_VALUE="Saving current puls ....."
WAIT,2

OPENW,65, rf_fname
PRINTF,65,'# Hyperbolic secant pulse'
PRINTF,65,'# Generated with pulsgen99'
PRINTF,65,'#
printf,65,'# Number of points in pulse   : ', np

```

```
printf,65,'# Number of lobes in sinc pulse : ',lobe
PRINTF,65,'#
```

```
FOR i=0,np-1 DO BEGIN
PRINTF,65,ROUND(phase(i)), ' ', ROUND(abs(amp(i))), ' 1.0'
ENDFOR
CLOSE,65
WIDGET_CONTROL, info.var4, SET_VALUE="File saved ....."
WAIT,1
ENDIF
```

```
fname = "
fname = saveas + 'HN'
path = '~/vnmrsys/shapelib/'
rf_fname = path + fname + '.RF'
rf_text = 'Save file as ' + rf_fname + '?'
doit2 = DIALOG_MESSAGE(rf_text, /QUESTION, TITLE="Save Hanning filtered too
?", DIALOG_PARENT="base1")
IF (doit2 eq 'Yes') THEN BEGIN
;Write pulse to ASCII file
;-----
```

```
WIDGET_CONTROL, info.var4, SET_VALUE="Saving hanning filtered puls ....."
WAIT,1
```

```
OPENW,65, rf_fname
PRINTF,65,'# Hyperbolic secant pulse'
PRINTF,65,'# Generated with pulsgen99'
PRINTF,65,'#
printf,65,'# Number of points in pulse : ', np
printf,65,'# Number of lobes in sinc pulse : ',lobe
PRINTF,65,'#
```

```
FOR i=0,np-1 DO BEGIN
PRINTF,65,ROUND(phase(i)), ' ', ROUND(abs(filtsinc(i))), ' 1.0'
ENDFOR
CLOSE,65
```

```
WIDGET_CONTROL, info.var4, SET_VALUE="Files saved ..."
WAIT,2
```

```
ENDIF
```

```
fname = "
```

```

fname = saveas + 'HM'
path = '~/vnmrsys/shapelib/'
rf_fname = path + fname + '.RF'
rf_text = 'Save file as ' + rf_fname + '?'
doit2 = DIALOG_MESSAGE(rf_text, /QUESTION, TITLE="Save Hamming filtered too
?", DIALOG_PARENT="base1")
IF (doit2 eq 'Yes') THEN BEGIN
;Write pulse to ASCII file
;-----

```

```

WIDGET_CONTROL, info.var4, SET_VALUE="Saving hamming filtered puls ....."
WAIT,1

```

```

OPENW,65, rf_fname
PRINTF,65,'# Hyperbolic secant pulse'
PRINTF,65,'# Generated with pulsgen99'
PRINTF,65,'#
printf,65,'# Number of points in pulse    : ', np
printf,65,'# Number of lobes in sinc pulse : ',lobe
PRINTF,65,'#

```

```

FOR i=0,np-1 DO BEGIN
PRINTF,65,ROUND(phase(i)), ' ', ROUND(abs(filtsinc2(i))), ' 1.0'
ENDFOR
CLOSE,65

```

```

WIDGET_CONTROL, info.var4, SET_VALUE="Files saved ..."
WAIT,2

```

```

ENDIF

```

```

;WIDGET_CONTROL, info.var4, SET_VALUE="File not saved, returning ..."
;WIDGET_CONTROL, info.var3, SET_VALUE=""
;WAIT,2
;ENDELSE

```

```

ENDIF

```

```

WIDGET_CONTROL, info.var4, SET_VALUE=""
WIDGET_CONTROL, event.top, SET_UVALUE=info
return
END

```

```

;-----

```

```

--
PRO setmo, event
; Get all the values in U_VALUE
WIDGET_CONTROL, event.top, GET_UVALUE=info, /NO_COPY

; ***** First input field widget
*****
; First destroy the old widget base (for quick realize afterwards)
WIDGET_CONTROL, info.basein, /DESTROY

; Set all the new widget values
info.basein = WIDGET_BASE(info.base, ROW=6, /BASE_ALIGN_LEFT)
; Kill window updating so everything can be set first
WIDGET_CONTROL, info.basein, UPDATE=0

labellabda = WIDGET_LABEL(info.basein, /ALIGN_LEFT, VALUE="Number of
points", FRAME=1, SCR_YSIZE=30, SCR_XSIZE=180);
info.var1 = WIDGET_TEXT(info.basein, SCR_XSIZE=120, /EDITABLE,
UVALUE="nee")
labellabda1 = WIDGET_LABEL(info.basein, /ALIGN_LEFT, VALUE="Number of
zerocrossings", FRAME=1, SCR_YSIZE=30, SCR_XSIZE=180);
info.var2 = WIDGET_TEXT(info.basein, SCR_XSIZE=120, /EDITABLE,
UVALUE="nee")
labellabda2 = WIDGET_LABEL(info.basein, /ALIGN_LEFT, VALUE="Save file as",
FRAME=1, SCR_YSIZE=30, SCR_XSIZE=100);
info.var3 = WIDGET_TEXT(info.basein, SCR_XSIZE=200, /EDITABLE,
UVALUE="nee")
labellabda3 = WIDGET_LABEL(info.basein, /ALIGN_LEFT, VALUE="info", FRAME=1,
SCR_YSIZE=60, SCR_XSIZE=100);

info.var4 = WIDGET_TEXT(info.basein, YSIZE=3, SCR_XSIZE=260, UVALUE="nee",
VALUE="", /WRAP)

info.tit = "Hyperbolic secant function"

; Setup the buttons for info.basein
button = WIDGET_BUTTON(info.basein, VALUE=" New plot  ", /ALIGN_CENTER,
EVENT_PRO="windowval", FRAME=1, UVALUE="plot")
button4 = WIDGET_BUTTON(info.basein, VALUE="Plot and Save", /ALIGN_CENTER,
EVENT_PRO="windowval", FRAME=1, UVALUE="save" )

; reset the window top update
WIDGET_CONTROL, info.basein, /UPDATE

; Post all the values back
WIDGET_CONTROL, event.top, SET_UVALUE=info

```

RETURN

END

```

;-----
--
;
PRO killer, event

; Get the U_VALUE
WIDGET_CONTROL, event.top, GET_UVALUE=info, /NO_COPY

; Destroy existing widget basein
WIDGET_CONTROL, info.basein, /DESTROY

; Create new empty one
info.basein = WIDGET_BASE(info.base,/BASE_ALIGN_LEFT)

;Repost values in U_VALUE
WIDGET_CONTROL, event.top, SET_UVALUE=info

```

RETURN

END

```

;-----
--
;-----
--
; Main window setup. Decide max values for windows here
;-----
--

```

PRO pulsgen

```

; Never change the next sentence, unless ver1.0+ has been made
titlemenu = "Pulsngen99 ver. 1.0 beta"

```

```
; Create info for the posting of all the widgetvalues through the whole
program
; A problem for IDL is that you can't just refer to every widget there is
; by stating the name (SO why give names??? Don't know)
; So every time you strat a PRO or function you have to read the main widget
U_VALUE
; Do some changes and at the end post it all back in to U_VALUE.
```

```

; Bleeuuuh
info = {var1:0.0, var2:0.0, var3:0.0, var4:0.0, var5:0.0, var6:0.0,
var7:0.0, var8:0.0, var9:0.0, $
var10:0.0, var11:0.0, var12:0.0, var13:0.0, var14:0.0, var15:0.0,
var16:0.0, var17:0.0, var18:0.0, $
base:0L, winbase:0L, drawwin:0L, basein:0L, tit:"}

; Main widget creation
base1 = WIDGET_BASE(COLUMN=2, TITLE=tit, MBAR=bar, SCR_XSIZE=977,
SCR_YSIZE=502, $
TLB_FRAME_ATTR=1)

; ***** Menu bar settings
*****

; First value in taskbar
bare1 = WIDGET_BUTTON(bar, VALUE="File", /MENU);
; bare1sub1 = WIDGET_BUTTON(bare1, VALUE="Iconify", UVALUE="icon");
bare1sub2 = WIDGET_BUTTON(bare1, VALUE="Exit", UVALUE="exit",
/SEPARATOR);
; bare1sub5 = WIDGET_BUTTON(bare1, VALUE="Print", EVENT_PRO="print_it",
/SEPARATOR);
bare1sub3 = WIDGET_BUTTON(bare1, VALUE="Clear system", UVALUE="clearall",
/SEPARATOR);
bare1sub4 = WIDGET_BUTTON(bare1, VALUE="Restart Pulsgen99",
UVALUE="restart", /SEPARATOR);

; Second value in taskbar
bare2 = WIDGET_BUTTON(bar, VALUE="View", /MENU);
bare2sub1 = WIDGET_BUTTON(bare2, VALUE="Clear input field",
EVENT_PRO="killer")
bare2sub2 = WIDGET_BUTTON(bare2, VALUE="Clear image field",
UVALUE="clear")
bare2sub3 = WIDGET_BUTTON(bare2, VALUE="!P.MULTI", UVALUE="multi",
/SEPARATOR)
bare2sub4 = WIDGET_BUTTON(bare2, VALUE="One Image view",
UVALUE="nomulti")

; Third value in taskbar
bare3 = WIDGET_BUTTON(bar, VALUE="Functions", /MENU);
bare3sub1 = WIDGET_BUTTON(bare3, VALUE="Hyperbolic secant",
EVENT_PRO="setmo");

; Xmanager for exiting (All the other values go by a pro)
; You could do it all by Xmanager, but clarity would be lost
XMANAGER, "exitthis", base1

; Create the main surrounding widget for input values, because when you kill
the basein widget

```

```
; the whole UI doesn't crash because of its surrounding parent info.base
info.base = WIDGET_BASE(base1,SCR_XSIZE=472, SCR_YSIZE=500, FRAME=1)

; Insertion of input values in the parent info.base
info.basein = WIDGET_BASE(info.base, ROW=8, /BASE_ALIGN_LEFT)

; Creation of the base widget for the drawwidget
info.winbase = WIDGET_BASE(base1, TITLE="Graphics window", SCR_XSIZE=722,
SCR_YSIZE=500, /BASE_ALIGN_CENTER, FRAME=1)

; Drawwidget creation
info.drawwin = WIDGET_DRAW(info.winbase, FRAME=0, SCR_XSIZE=500,
SCR_YSIZE=500, /ALIGN_CENTER, RETAIN=0)

; The final control function that sets the U_VALUE with info
; Says it;s NO_COPY, so the info is overwritten every time you change
something
; not just copied.
; AND the base1 widget is REALIZED
WIDGET_CONTROL, base1, /REALIZE, SET_UVALUE=info, /NO_COPY,
DEFAULT_FONT="HELVETICA*14"
END
```

```
***** END OF PROGRAM *****
```
