
Subject: cancel button to stop w/in an iterative loop?
Posted by [T Bowers](#) on Mon, 08 Mar 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi.

I'm trying to create a simple cancel button for my software. I slapped a GUI on an iterative function that processes alot of data files. I want a cancel button to appear just as the run enters the processing loop, and exit out of processing if the user pressed the button.

Below is a file tGUI.pro to show one of the variations I've tried. I even tried to register the cancelBase base widget with xmanager, then destroy it in the cancelProcessing() callback, then check the value of xregistered in the outer loop to see if it's registered. No such luck. I'm obviously missing something here. Some of this code is strait outta my app so please forgive my coding style (e.g. the loop to create the widget buttons in the procedure tGUI). Oh, and global variables via a COMMON block is my LEAST favorite option. If that's the only way to go, I guess I'll have to do it, but that'll be my last option. Even so, I did try it with a common block, and I still couldn't get it to work.

Thanks alot in advance

IDL> print, !version

{ x86 Win32 Windows 5.2 Oct 30 1998}

;---CUT HERE-----

;//

```
function cancelProcessing_event, event
  widget_control, event.top, GET_VALUE=eventValue
  if eventValue EQ "CANCEL PROCESSING" then $
    widget_control, event.id, SET_UVALUE = 2
  return, 1
end
```

;//

;//

```
function processData
  ;//Create my cancel button
  cancelBase = Widget_Base(UNAME='cancelBase',/ALIGN_CENTER,$
    /GRID_LAYOUT ,TLB_FRAME_ATTR=9)
  cancelButton = Widget_Button(cancelBase, UNAME='cancelButton',$
    /ALIGN_CENTER ,VALUE='CANCEL PROCESSING', $
```

```

EVENT_FUNC="cancelProcessing_event")
Widget_Control, /REALIZE, cancelBase

; //Set something I can use to know it's been pressed
widget_control, cancelBase, SET_UVALUE=0

; //Create an outer and inner loop. I wanna check to see if the cancel
; // button was pressed at the beginning of every iteration of outer loop.
for j = 1,3 do begin
widget_control, cancelBase, GET_UVALUE=widValue
if widValue EQ 2 then begin
print, "User Cancelled"
goto, BREAK
endif
for i = 0, 100 do begin
print, i*j
endfor
endfor

; //User pressed cancel button so jumped to here
BREAK:
; // ...destroy the cancel button 'n base
widget_control, cancelBase, /destroy

return, 0
end
;//////////////////////////
////////////////////////

;//////////////////////////
////////////////////////
pro MainForm_event, ev
widget_control, ev.id, GET_VALUE = eventval
case eventval of
"Process Data": retCode = processData()
"Quit": widget_control, ev.top, /destroy
endcase
end
;//////////////////////////
////////////////////////

;//////////////////////////
////////////////////////
pro tGUI
; //There'll be 2 buttons
mainFormButtonList = ["Process Data","Quit"]
; //Create base
baseWidget = widget_base(TLB_FRAME_ATTR=1)

```

```

; //Create button base widget off of base
buttonBaseWidget = widget_base(baseWidget, /column, space=10)

; //create buttons...
numButtons = n_elements(mainFormButtonList)
mainFormButtons = lonarr(numButtons)
for i=0,numButtons-1 do begin
    mainFormButtons[i] = widget_button(buttonBaseWidget,
value=mainFormButtonList[i])
endfor

widget_control, baseWidget, /realize

; //Pass off to xmanager
xmanager, "mainForm", baseWidget, /no_block
end
; //////////////////////////////////////
////////////////////////////////////
; ---CUT HERE-----

```

Subject: Re: cancel button to stop w/in an iterative loop?
Posted by [R.Bauer](#) on Wed, 10 Mar 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

an example is included.
try x_slider_example.

After you pressed break a breakpoint is set into the running script.
It breaks then itself.

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:
; x_slider

```

```

;
; PURPOSE:
; This widget function is used to place a slider in front to show steps of
; actions like an increment of a do loop.
; The break key is used to set if pressed a breakpoint into the source code.
; This widget don't use the xmanager
;
; CATEGORY:
; PROG_TOOLS/WIDGETS
;
; CALLING SEQUENCE:
; result=x_slider(minimum,maximum,[title=title],[/BREAK])
;
; INPUTS:
; minimum: the start value
; maximum: the end value
;
; KEYWORD PARAMETERS:
; title: the title for the base
; break: if set the break key is defined - if pressed a breakpoint
;       will be generated where x_slider_widget_event was started
;
; OUTPUTS:
; The result is a structure:
;   base: the base id
;   slider: the slider id
;
; EXAMPLE:
; result=x_slider(0,10000)
; widget_control,result.slider,set_val=100
; widget_control,result.base,/destroy
;
;
; PRO x_slider_example
;   result=x_slider(0,1000,title='CC',/break)
;   FOR I=0,10 DO BEGIN
;     WAIT,1
;     WIDGET_CONTROL,result.slider,set_val=100*I
;     X_SLIDER_WIDGET_EVENT,result
;   ENDFOR
;
;   WIDGET_CONTROL,result.base,/destroy
;
; END
;
; MODIFICATION HISTORY:
; Written by: R.Bauer (ICG-1), 1998-Nov-18
; 1998-12-02: keyword break added

```

;-

```
PRO x_slider_example
  result=x_slider(0,1000,title='CC',/break)
  FOR I=0,10 DO BEGIN
    WAIT,1
    WIDGET_CONTROL,result.slider,set_val=100*I
    X_SLIDER_WIDGET_EVENT,result
  ENDFOR
  WIDGET_CONTROL,result.base,/destroy
END
```

```
PRO x_slider_widget_event,x_slider_struct
  tagname=STRUPCASE('break')
  tags=TAG_NAMES(x_slider_struct)
  a=WHERE(tags EQ tagname,count)
  IF count GT 0 THEN BEGIN
    HELP,call=call
    line=LONG((STR_SEP((STR_SEP(call[1], ' ')[1], '))[0])
    file=(STR_SEP((STR_SEP(call[1], '<')[1], '('))[0]
    x_slider_struct.file=file
    x_slider_struct.line=line
    WIDGET_CONTROL,x_slider_struct.base,set_uval= x_slider_struct
    res=WIDGET_EVENT(x_slider_struct.break,/nowait)
  ENDIF
END
```

```
PRO x_slider_manager_break,event
  WIDGET_CONTROL,event.top,get_uval=uval
  BREAKPOINT,uval.file,uval.line
END
```

```
FUNCTION x_slider,minimum,maximum,title=title,BREAK=BREAK
  base=WIDGET_BASE(row=2,title=title)
  IF KEYWORD_SET(break) THEN id_break=WIDGET_BUTTON(base,Value='BREAK',$
    UVALUE='BREAK',/ALIGN_CENTER,EVENT_PRO='x_slider_manager_bre ak')
  base2=WIDGET_BASE(base,col=3)
  IF N_ELEMENTS(minimum) EQ 0 THEN minimum=0
  IF N_ELEMENTS(maximum) EQ 0 THEN maximum=100
  id_label=WIDGET_LABEL(base2,value=STRTRIM(STRING(minimum),2) )
  slider=WIDGET_SLIDER(base2,minimum=minimum,maximum=maximum)
  id_label=WIDGET_LABEL(base2,value=STRTRIM(STRING(maximum),2) )
  WIDGET_CONTROL,base,/realize,/timer
  result={base:base,slider:slider,line:0l,file:''}
  WIDGET_CONTROL,base,set_uval= result
  IF KEYWORD_SET(break) THEN result=CREATE_STRUCT(result,'break',id_break)
  RETURN,result
```

END
