Subject: Re: Scrolling text in a widget_label
Posted by Russell[1] on Fri, 09 Dec 2011 21:06:51 GMT
View Forum Message <> Reply to Message

I didn't like the idea of testing if a tag exists, and felt like there had to be a better way. Here's a better way, it still uses / tracking_events, but it's free of that call to tag_names.pro. Also, if the structure of a tracking_event changes, this should still work since it's using a widget_timer event.

```
pro scroll_event,event
widget_control,event.id,get_uval=uval
case uval of
   'TEXT': begin
      widget_control,event.top,get_uval=state
      (*state).stop=1b-event.enter
      if event.enter then begin
         widget_control,event.id,get_val=text
         (*state).iter=0L
         (*state).text=text(0)
         (*state).textwid=event.id
         widget_control,(*state).wtime,timer=(*state).time
      endif
   end
   'TIME': begin
      widget_control,event.top,get_uval=state
      if (*state).stop then begin
         widget_control,(*state).textwid,set_val=(*state).text

      endif else begin
         text=(*state).text+'   '
         (*state).iter=((*state).iter+1) mod strlen(text)
         widget_control,(*state).textwid,set_val=$
                    string(shift(byte(text),-(*state).iter))
         widget_control,event.id,timer=(*state).time
      endelse
   end
   'CLOSE': widget_control,event.top,/destroy
   else: return
endcase
end

pro scroll

base=widget_base(/col)
wtime=widget_base(base,uval='TIME')
```

```
wtext=widget_text(base,uval='TEXT',xsize=20,/track,$
               value='This is a long string in a small box that
will scroll')
wstart=widget_button(base,value='Close',uval='CLOSE')

state={wtext:wtext,$           ;widget id for the text box
      wtime:wtime,$            ;widget id for the timer (null base)
      text:'',$                ;the text to scroll
      textwid:-1L,$            ;the widget id of the text to scroll
      time:0.1,$               ;the wait time (ie. scroll speed)
      stop:0b,$                ;flag to stop the scrolling
      iter:0l}                 ;some counter
state=ptr_new(state,/no_copy)
widget_control,base,/realize,set_uval=state
xmanager,'scroll',base,/no_block


end
```

I wonder if I can make this a compound widget? with it's own event handler.... hmmmmm...

R

On Dec 2, 11:20 am, Russell <rryan....@gmail.com> wrote:

> Yeah, I've read that tracking events are a delicate thing. My hope
> would be that the Users will be able to deal with this, since the only
> purpose of this was to show this long string. In my real problem, the
> widget_label displays a file name, and for various reasons, I'd like
> that file name to be the fullpath to the file. That said, it is a
> real possibility that the filename will be longer than the space
> available for it. All I wanted was a way of keeping the fullpath and
> widget_label size, but allowing the user to see the filename. My
> first hope was that I could use CONTEXT_EVENTS with a widget_label,
> but alas. If all else failed, I was going to change the widget_labels
> to widget_texts and then they could use the arrows to move around.
> But the widget_texts take up (proportionally) more space on the GUI

```
>  than does the widget_label.  I have space to spare, but I didn't want
>  to spare it to that....
>
>  Thanks again!!!
>
>  Russell
>
>  On Dec 1, 6:32 pm, David Fanning <n...@dfanning.com> wrote:
>
>
>
>
>
>
>
>>  Russell writes:
>>>  So, I did get it to do what I want, though I don't entirely understand
>>>  why it works.  It seems that widget_event(/nowait) returns the "bare
>>>  bones" structure of {ID:0, TOP:0, HANDLER:0} for all times that the
>>>  mouse is *inside* the widget.  When you move out, it will add the tag
>>>  "ENTER".  So it was simple enough to test on whether or not the
>>>  "ENTER" tag exists or not.  I've attached a quick script to illustrate
>>>  it.  The first function is from the IDL astro-library maintained by W.
>>>  Landsman --- just in case someone else is interested in this.  I'm not
>>>  sure it was worth all the time I spent on this, but it was a fun
>>>  exercise!  Now to figure out why widget_event changes its return
>>>  structure like that, and whether or not I can trust this in the
>>>  future....
>
>>  Pretty neat! :-)
>
>>  Just a word of warning, though. Tracking enter and exit
>>  events is *extremely* fragile and operating system dependent.
>>  It helps a great deal if you can train your users to move
>>  their cursors v-e-r-y s-l-o-w-l-y.
>
>>  I guess what I am saying is, this kind of thing works better
>>  on your computer than it does on everyone else's. :-)
>
>>  Cheers,
>
>>  David
>
>>  --
>>  David Fanning, Ph.D.
>>  Fanning Software Consulting, Inc.
>>  Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
>>  Sepore ma de ni thui. ("Perhaps thou speakest truth.")
```