
Subject: Re: pop up widget

Posted by [davidf](#) on Wed, 28 Jul 1999 07:00:00 GMT

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

Arno R. Schleich (warschew@zedat.fu-berlin.de) writes:

- > I am looking for a technique to produce a widget with the following
- > characteristics:
- >
- > 1.) It pops up upon fulfillment of a certain program condition
- > 2.) It always stays in front
- > 3.) No user interaction can take place with the widget or with the
- > remaining widgets of the program
- > 4.) The widget disappears automatically after a limited period of time
- > and normal interaction with the remaining widgets is going to take place
- > again

After thinking about this for quite some time, I've decided this is probably NOT what you want. (I have a friend who says that prayer is so powerful you shouldn't try it more than once or twice a year. You might actually get what you want and then you would be in REAL trouble.)

But, because I'm a nice guy, I've given you both what you asked for and what you probably need in the same program. :-)

Ever since someone asked about a Show Progress indicator the other day, I've been thinking that this should really be done as an object. This question gave me the opportunity to write something up. While I was at it, I decided to add the sort of blocking action you refer to above. I call this the AutoUpdate mode. But I think you will probably find the Normal mode more useful to you.

Those of you who have fooled around with my ShowProgress program (which I have always found way too hard to understand how to use, even after I wrote the darn thing), will be pleased with the ease of use that this object provides.

Here is the essence of how it works, in a bit of the documentation header:

```
; PROCEDURE:
; There are two modes. In AutoUpdate mode, a delay and number of steps is
; required. The modal widget stays on the display until the total time
; exceeds the DELAY or the requested number of STEPS is taken. A TIMER
; widget is used to generate update events. Nothing can be going on
; concurrently in AutoUpdate mode. To enter AutoUpdate mode, type this:
```

```

;
; progressBar = Obj_New("SHOWPROGRESS", /AutoUpdate, Delay=2, Steps=10)
; progressBar->Start
; Obj_Destroy, progressBar
;
; In normal mode, the user is responsible for starting, updating, and
; destroying the progress indicator. The sequence of commands might look
; like this:
;
; progressBar = Obj_New("SHOWPROGRESS")
; progressBar->Start
; FOR j=0,9 DO BEGIN
;   Wait, 0.5 ; Would probably be doing something ELSE here!
;   progressBar->Update, (j+1)*10
; ENDFOR
; progressBar->Destroy
; Obj_Destroy, progressBar
;
; Normal mode gives you the opportunity to update the Progress Bar
; in a loop while something else is going on. See the example program
; at the end of this file.

```

I've included a short example program at the end of the file, so you can quickly see how things are done. Just compile the file and type "EXAMPLE":

```

IDL> .Compile showprogress__define
IDL> Example

```

I haven't done any extensive testing of this program. (Indeed, I've spent WAY too much time on it as it is.) So, if you find a bug, let me know. (If you could fix it too, I would be most appreciative.)

You can find the program here:

ftp://www.dfanning.com/pub/dfanning/outgoing/misc/showprogress__define.pro

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155