Subject: Re: Lose control of IDL window
Posted by Mark Hadfield on Wed, 24 Jan 2001 20:52:18 GMT
View Forum Message <> Reply to Message

<philippe_fischer2000@yahoo.ca> wrote in message
news:94mts8$8lv$1@nnrp1.deja.com...
> I have recently switched from using IDL in a UNIX environment to
> Windows2000 (not my choice!). When running a job in the idlde I lose
> control of the window while the job is running. I regain control once
> the job finishes. By lose control I mean I cannot maximize the window
> (if I have minimized from the toolbar), I cannot view output in the
> window and I cannot "break" out of the job. If I try to put the cursor
> over the idlde window it disappears. An example of job that causes this
> problem is:
>
> for i=1L,1000000 do print,'hi'
>
> Has anybody encountered this problem?

Er ... yes.

> I have been corresponding with
> somebody at RSI but he hasn't come up with any solutions yet.

That doesn't surprise me.

What does surprise me (slightly) is that the problem doesn't occur on Unix. (I have run IDL on various versions of Windows for 7 years now, but hardly ever on Unix.) I think it has to do with the differences between Unix (or X-Windows) & Windows in how they allocate responsibility for managing windows: in Unix the window manager can minimise a window but on Windows it leaves this to the application. If IDLDE is busy then it neglects to manage its window and all events received inside the IDLDE window boundaries are queued up. The only exception seems to be that IDLDE clears its events when a program calls the WIDGET_EVENT function. (But I am guessing to some extent here.)

<old_fart_mode>Actually you young people of today don't know how lucky you are. Back in the days of 16-bit Windows, running any time-consuming command in IDL would make the entire machine unresponsive until the command had completed.</old_fart_mode>

Anyway to get to a solution (of sorts): Back in the 16-bit Windows days I wrote a widget application that allowed other applications a look in by calling the Windows API Yield function. This has metamorphosed over the years into my MGHwaiter class. The idea is that you create a MGHwaiter object then during a lengthy calculation you can make regular calls to the object's Yield method. See:

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/
http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mghwai ter__define.pro

I don't know if this "solution" is of any use to you. You may find, as I have, that you can live with an unresponsive IDL window after all. Why not play Pinball while the command is running? Or scan the comp.lang.idl-pvwave newsgroup?

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield/
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand