Subject: Re: GUI interface update issues
Posted by Rick Towler on Wed, 22 Jun 2005 15:57:42 GMT
View Forum Message <> Reply to Message

Clearly I am learning this as I go and I probably posted a little too early... I never have had issues with GUI interface drawing and I never really cared about interrupting program execution. Let the users wait. My code is pure genius at work! :) lol

> Reading the IDL_BailOut docs leads me to believe that IDL faithfully
> captures interrupts and will break before the next IDL command is
> executed and that the reason for IDL_BailOut is to provide an exit
> from external routines.
>
> But it seems that this is overly simplified. If I try to break
> my (pure IDL) program's execution once it has entered the model
> procedure it will continue to run thru the loop and thru 50 or
> so more lines of IDL code after the loop before it stops at a
> call to SAVE.
>
> So IDL is capturing the interrupt signal, but there seems to be
> limits on when it will stop execution based on where IDL is in
> the call stack. I would guess that SAVE has a call to
> IDL_BailOut to which is why my program breaks on SAVE. If I
> didn't have that call to SAVE execution would continue back up
> the call stack to some point where the interrupt would be handled?
>
> What's the logic? Is this an issue of how I have structured
> my program or is this how IDL always handles interrupts and I
> haven't noticed until now?

I'll add that the GUI gets updated when program execution returns to the main program event handler. For example, I batched 3 model runs where the 3 runs are executed via a loop in the main event handler. GUI drawing is suspended while within the model procedure, but when control returns to the main event handler the GUI is updated right before the program enters another model procedure (where GUI updating is again suspended).

The thing that confounds this issue is that when the application enters a model procedure the GUI is updated for a few seconds, maybe 20 seconds. *Then* the GUI fails to update. The reason why I never noticed this until now is that all of our models ran in less than 20 seconds. Now with models that take 2 minutes to 4 hours the lack of updating is apparent.

So now I am really curious how IDL handles this on other OSes. And it is starting to smell fishy (that's funny. These are acoustic

backscattering models of fish. Get it? Fishy? <nudge> <nudge> <wink>
<wink>)

<sigh> I need to make a pot of coffee.

-Rick