Subject: Re: really dumb widget question
Posted by Paddy Leahy on Tue, 03 Sep 2013 14:37:32 GMT
View Forum Message <> Reply to Message

On Friday, April 14, 2000 8:00:00 AM UTC+1, Mark Fardal wrote:
> (David Fanning) writes:
>
>>> When IDL is stopped within a procedure, a non-blocking widget will
>>> still listen to events. But it won't act on them. Why not? Is there
>>> any way to make it act on them?
>>
>> I presume because STOP means STOP. If it meant MOSTLY STOP,
>> I expect RSI would hear about it. :-)
>>
>> I would try .CONTINUE to get them going again.
>
> Hi,
>
> Well, I did know that much. But as I noted, you have to wait until
> returning to the main level. (Or is it just leaving the procedure with
> the STOP? Didn't test.) That isn't always what I want.
>
> Say you write two versions of a procedure. One takes user input from
> the command line, the other takes it from a non-blocking widget. If
> you stop at a certain point and need to use the procedure before going
> on, you can use the command-line version. But the GUI version is
> useless, unless you rewrite it as a blocking widget.
>
> Non-blocking widgets introduce parallel threads of execution to an
> environment that doesn't otherwise use threads. I would naively
> expect a STOP to stop the thread it's in, but apparently it stops all
> of them--except for the one that _records_ widget events for later
> processing. I'm basically wondering if one thread can tell another
> thread to go ahead and execute.
>
> Mark

Here we are in 2013, IDL 8.2 and this is still happening. I think it is a bug, not a feature. The symptom is this: if a script has been run and hit a stop statement, then control returns to the command line. If you don't bother to .continue or retall or equivalent, you can happily run any normal procedure and get the expected results. But if you have a non-blocking widget program running under XMANAGER, the event loop is halted. It re-starts again (with the accumulated requested events being acted on) as soon as you .continue etc. If you start off a widget utility after some script has hit a stop, the widget procedure works (creates the widget etc) up until the point that xmanager is invoked, then halts, leaving you with a frozen widget until you .continue the script. Typing "STOP" on the command line does not have this effect, it has to be in a procedure: any procedure (or function), typically with no connection at all with the widget you are trying to run in the background.