
Subject: Re: GUI interface update issues

Posted by [Haje Korth](#) on Wed, 22 Jun 2005 12:16:14 GMT

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

Karl,
thanks for introducing me to the IDL_BailOut function, which may be very helpful in my dlm loops. While it may be documented, the IDL documentation consists of several thousand pages now, and it is impossible (for me at least) to remember every entry in it. Rick raised some interesting question that I would be curious to hear about, too. Can you ellaborate on the interrupt issue?

Thanks,
Haje

"Karl Schultz" <k____schultz@rsinc.com> wrote in message
news:pan.2005.06.21.20.04.05.703000@rsinc.com...

> On Tue, 21 Jun 2005 09:23:39 -0700, Rick Towler wrote:

>

>> Hello group,

>>

>> I have an interesting problem with gui redraw on WinXP which maybe

>> someone can shed some light on.

>

> snip

>

>> Why does IDL stop updating the gui? Any ideas? While the functions in

>> the dlm are compute intensive, they aren't particularly complicated and

>> only rely on some simple macros in an .h file. Just a *lot* of looping

>> over a moderate amount of data.

>

> You might try calling the (documented) IDL_BailOut function. One purpose

> for this function is to let users have the opportunity to interrupt IDL

> during a long-running calculation. Many of the IDL internal system

> routines call IDL_BailOut as they make progress through a calculation. It

> is probably good form to code your C DLM's to call this function every so

> often so that users can break out of it.

>

> One side effect of IDL_BailOut is to flush window events, which depending

> on how your app is designed, might help keep the GUI's and progress bars

> fresh during the long-running operation.

>

> I'm not completely certain this will help you, but it is worth a try. It

> is very easy to give it a shot. If it helps, then spend a little more

> time to figure out a good policy for calling BailOut so that you don't

> call it too often or too infrequently. You don't want to slow down your

> function TOO much.

>
> Karl
>
