
Subject: Re: stop on arithmetic error

Posted by [simona bellavista](#) on Wed, 20 Mar 2013 16:13:05 GMT

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

of course, I can put !except=0 and put a conditional breakpoint with 'check_math() eq 128' in the offending line, but I wanted to avoid this convolved solution.

On top of that, I would like to be able to examine the stack (as I would say in fortran), I mean I would like to be able to see where the function was called and with which input values, is this possible?

On Wednesday, March 20, 2013 2:12:14 PM UTC+1, simona bellavista wrote:

> I have a code with several arithmetic errors (Floating illegal operand, overflow/underflow) (sigh!), and I use !EXCEPT=2 for locating them, but idl only tells me the line of the function where there is the error and not where the function is called in the main.

>

>

>

> Is there a way to stop the execution when the error happens? I am using idlwave with IDL 8.2

Subject: Re: stop on arithmetic error

Posted by [wlandsman](#) on Wed, 20 Mar 2013 16:55:11 GMT

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

On Wednesday, March 20, 2013 12:13:05 PM UTC-4, simona bellavista wrote:

>

> On top of that, I would like to be able to examine the stack (as I would say in fortran), I mean I would like to be able to see where the function was called and with which input values, is this possible?

>

I am a little confused by this part of the question. If you are using the Workbench then you should automatically see the traceback and variables at a breakpoint. If you are at the command line, then HELP should display all the variables; individual variables can be displayed with PRINT, and you can get a traceback with HELP,/TRACE. You can examine variables in the calling program using HELP,LEVEL=-1 --Wayne
