
Subject: Re: Programming annoyances

Posted by [Kenneth P. Bowman](#) on Fri, 09 Sep 2005 21:44:41 GMT

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

In article <1126300702.285794.90680@g43g2000cwa.googlegroups.com>, swingnut@gmail.com wrote:

> I tried putting N_REFR into a separate pro file, but I got the same
> behavior. I eventually changed the function name to be N_refr, and now
> it seems to work but unreliably. I've fiddled with changing the
> function call to match, but the subtleties of case sensitivity seem to
> be showing up. This happens whether both at the IDL prompt and in
> idlde. In any case, what can cause this happen?

The function N_REFR should be in a file called n_refr.pro. Obviously that file must be in IDL's search path. Internally IDL is case insensitive, but filenames should be all lowercase for portability.

> Second, on the occasions when the program does run, when it reaches the
> end of the main procedure, IDL prints two or three error messages to
> the log:
>
> Program caused arithmetic error: Floating point divide by 0
> Program caused arithmetic error: Floating underflow
>
> and sometimes
>
> Program caused arithmetic error: Floating overflow
>
> I added a tracing statement to determine for sure that this happens at
> the end, and sure enough, it is not generated by anything in the code,
> though I could imagine it being generated by something being left out at
> the end. Any ideas?

Floating underflow errors can usually be ignored safely.

Overflow and divide by zero errors generally should not be ignored (unless you don't mind getting Inf's and NaN's for answers). You can use the FINITE function to find Inf's and NaN's in your variables. Personally, I make a point of ensuring that my programs do not generate exceptions other than, possibly, underflows.

Check out the !EXCEPT system variable. Setting !EXCEPT = 2 forces IDL to report exceptions after each IDL statement, which should help you identify the exact location where the exception occurs.

Ken Bowman

Subject: Re: Programming annoyances

Posted by [Antonio Santiago](#) on Mon, 12 Sep 2005 05:52:00 GMT

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

swingnut@gmail.com wrote:

> Ok, I've got some code. I've tweaked it so that it actually runs again
> (sortof; it was written by another grad student back in 2001-2002 but
> hasn't been used much since then). Two issues that have come up, which
> I don't understand.

>

> *****

> First, the pro file contains a function, N_REFR. IDL compiles this when
> I select the compile all option, and sometimes IDL sees it but most of
> the time IDL gives a variable undefined error.

>

> ... (other modules compiled)

> % Compiled module: N_REFR

> % Compiled module: MASSPROFILE4

> IDL> massprofile 4

> ...(log messages)

> % Variable is undefined: N_REFR

> % Execution halted at: RPIPROFILER 286

> /data/aramisgm/research/tracing/testing/massprofile4.pro

> % MASSPROFILE4 667

> /data/aramisgm/research/tracing/testing/massprofile4.pro

> % \$MAIN\$

> % Program caused arithmetic error: Floating underflow

>

Perhaps a bad assign instruction is the problem. Although you have been
compiled the N_REFR function and MASSPROFILE4 procedure/function you
could have a:

a= NREFR ---> BAD (without parenthesis) Variable N_REFR undefined.

a= NREFR() --> OK

> I tried putting N_REFR into a separate pro file, but I got the same
> behavior. I eventually changed the function name to be N_refr, and now
> it seems to work but unreliably. I've fiddled with changing the
> function call to match, but the subtleties of case sensitivity seem to
> be showing up. This happens whether both at the IDL prompt and in
> idlde. In any case, what can cause this happen?

>

> BTW, this is IDL 6.2 on a Red Hat 8 cluster.

>

> *****

> Second, on the occasions when the program does run, when it reaches the
> end of the main procedure, IDL prints two or three error messages to

> the log:
>
> Program caused arithmetic error: Floating point divide by 0
> Program caused arithmetic error: Floating underflow
>
> and sometimes
>
> Program caused arithmetic error: Floating overflow
>
> I added a tracing statement to determine for sure that this happens at
> the end, and sure enough, it is not generated by anything in the code,
> though I could imagine it being generated by something being left out at
> the end. Any ideas?
>
> Thanks.
>

--

Antonio Santiago Piñeres
(email: santiago@grahi.upc.edu)
(www: <http://www.grahi.upc.edu/santiago>)
(www: <http://asantiago.blogspot.org>)

GRAHI - Grup de Recerca Aplicada en Hidrometeorologia
Universitat Politècnica de Catalunya
