
Subject: Re: Runtime error woes

Posted by [James Kuyper](#) on Mon, 30 Jul 2007 11:17:40 GMT

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

mort canty wrote:

> I mentioned this one before, but I'm really hung up on it, so I'll try
> again:
>
> Running a rather long program on my desktop PC under Windows XP, IDL 6.3
> with !EXCEPT=2 I get the following messages:
>
> Program caused arithmetic error: Floating illegal operand
> Detected at FFN::COST 160 D:\idl\projects\supervised\ffn__define.pro
> Program caused arithmetic error: Floating illegal operand
> Detected at FFN::COST 162 D:\idl\projects\supervised\ffn__define.pro
> Program caused arithmetic error: Floating illegal operand
> Detected at FFN::COST 164 D:\idl\projects\supervised\ffn__define.pro
> Program caused arithmetic error: Floating illegal operand
> Detected at FFN::COST 164 D:\idl\projects\supervised\ffn__define.pro
> Program caused arithmetic error: Floating illegal operand
> Detected at FFNKAL::TRAIN 112
> D:\idl\projects\supervised\ffnk__define.pro
> Program caused arithmetic error: Floating illegal operand
> ...
> ...
>
> This goes on for a while and then the program crashes. I cannot for the
> life of me find any errors in the code at the lines indicated that
> could cause this. In fact, I don't believe there are any, because what's

The key question is what precisely happens on the lines from which those messages are generated? I'd recommend inserting debugging print statements just before line 160, printing out the value of every variable whose value could affect the behaviour of that line of code. Depending upon the nature of your program, that might involve an awful lot of output, so you might need to dump it to a file. It's a brute force method, but I've found it useful when more efficient methods don't work.

> driving me nuts is this: On my laptop, exact same program, exact same
> OS, exact same IDL, still with !EXCEPT=2, I get no runtime error
> messages whatsoever and the program completes successfully.

Is the underlying hardware the same? If a different kind of chip is handling the floating point math on your desktop than is handling it on your laptop, that could be the source of the differences. It could as minor as a tiny difference in the accuracy with which certain operations are performed, with that tiny difference being magnified into an error condition by poorly written code - or it could be an outright error in the FP implementation on one of the chips.

It's much easier to diagnose such problems, when you know which particular values the variables have at the time the arithmetic error message is generated.
