Subject: Re: Going outside the interger limits
Posted by David Fanning on Tue, 21 Sep 2004 13:25:36 GMT
View Forum Message <> Reply to Message

Lloyd Watkin writes:

> Have been muddling about with this problem for the past weeek, and I'm
> hoping someone can help.
>
> I'm creating an atmospheric transmission model for the sub-mm/far
> infrared region. If I try to calculate the spectrum between 1 and 500
> wavenumbers (numbers are not important) and I set a high resolution,
> the end of the spectrum does not seem to have had spectral line data
> applied to it.
>
> Having looked at where this is happening (at about point number
> 32700), it lies at the point where an integer runs outside it's
> limits. Inside the code, if a spectral line value is negative then it
> gets ignored, hence why I'm assuming (well guessing really) that this
> is the problem.
>
> I have been through my code and converted as much as I can see into
> either a long or a float depending on what it needs to be.
>
> Still getting problems!
>
> I was wondering whether there is anyway to tell in IDL if an integer
> is trying to go outside it's limits? Such as !except = 2, or is there
> a compiler switch which would break the execution if this problem was
> encountered (the latter would be very handy).
>
> Thanks for any help,

I think I would put this inside the program module:

```
Compile_Opt defint32
```

Then, *all* your integers will be long enough to avoid overflow.

More commonly, we use:

```
Compile_Opt idl2
```

which sets long integers and enforces strict array subscripting,
another great idea. :-)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http:/www.dfanning.com/
Phone: 970-221-0438, IDL Book Orders: 1-888-461-0155

Subject: Re: Going outside the interger limits
Posted by K. Bowman on Tue, 21 Sep 2004 13:26:19 GMT
View Forum Message <> Reply to Message

In article <c3f97ff.0409210321.47f8725f@posting.google.com>,
lloyd@evilprofessor.co.uk (Lloyd Watkin) wrote:

> Hi all,
>
> Have been muddling about with this problem for the past weeek, and I'm
> hoping someone can help.
>
> I'm creating an atmospheric transmission model for the sub-mm/far
> infrared region. If I try to calculate the spectrum between 1 and 500
> wavenumbers (numbers are not important) and I set a high resolution,
> the end of the spectrum does not seem to have had spectral line data
> applied to it.
>
> Having looked at where this is happening (at about point number
> 32700), it lies at the point where an integer runs outside it's
> limits. Inside the code, if a spectral line value is negative then it
> gets ignored, hence why I'm assuming (well guessing really) that this
> is the problem.
>
> I have been through my code and converted as much as I can see into
> either a long or a float depending on what it needs to be.
>
> Still getting problems!
>
> I was wondering whether there is anyway to tell in IDL if an integer
> is trying to go outside it's limits? Such as !except = 2, or is there
> a compiler switch which would break the execution if this problem was
> encountered (the latter would be very handy).
>
> Thanks for any help,
>
> Lloyd Watkin

As far as I know, IDL does not report underflow or overflow for integer
data types, and there is no way to force it to do so. You will have to

do it "manually" (e.g., IF (wavenumber LT 0) THEN STOP) and use HELP to find the culprit.

I put

COMPILE_OPT IDL2

in my startup.pro and in all program units (procedures and functions).

This is equivalent to

COMPILE_OPT DEFINT32, STRICTARR

It changes the default integer type to LONG (32-bit) and requires the use of square brackets [ ] for array subscripts (to avoid confusion with function calls).

In my opinion, with modern PCs and workstations, you should have a really good reason to use INTs instead of LONGs.

Ken Bowman

Subject: Re: Going outside the interger limits
Posted by lloyd on Wed, 22 Sep 2004 14:37:07 GMT
View Forum Message <> Reply to Message

In the end, it turned out that I was taking fix(a float) where as what I should of done is taken long(a float) as I knew I'd be going outside the limit.

Typical!

Thanks for all the help as usual,

Lloyd Watkin
