Subject: Re: loop limit ???
Posted by Foldy Lajos on Wed, 18 Apr 2007 21:02:48 GMT
View Forum Message <> Reply to Message

On Wed, 18 Apr 2007, kostis wrote:

> i wouldnt expect that IDL has a limit for loops !!
> trying a loop of 50.000 steps i got the message:
>
> % Compiled module: TRAJECTORY.
> % Loop limit expression too large for loop variable type.
> <LONG (49999)>.
> % Execution halted at: TRAJECTORY 16 /home/kostis/PROJECT
> LARMOR/dipole/trajectory.pro
> % $MAIN$
>
> Why is this?
> This means i cant have bigger loops... or is there a trick??
>

IDL> for i=0,49999 do begin & endfor
% Loop limit expression too large for loop variable type.
<LONG (49999)>.
% Execution halted at: $MAIN$

0 is of type INT (default integer), whose max. value is 32767.
Use 0l (LONG) instead.

regards,
lajos

Subject: Re: loop limit ???
Posted by Michael Galloy on Wed, 18 Apr 2007 21:12:10 GMT
View Forum Message <> Reply to Message

On Apr 18, 2:54 pm, kostis <kostis...@gmail.com> wrote:
> i wouldnt expect that IDL has a limit for loops !!
> trying a loop of 50.000 steps i got the message:
>
> % Compiled module: TRAJECTORY.
> % Loop limit expression too large for loop variable type.
> <LONG (49999)>.
> % Execution halted at: TRAJECTORY 16 /home/kostis/PROJECT
> LARMOR/dipole/trajectory.pro
> % $MAIN$
>

> Why is this?
> This means i cant have bigger loops... or is there a trick??

There's a "trick". A FOR loop variable gets it type from the "start"
value of the loop and it doesn't change in the course of the loop.

So, (note the 0L -- it's the "trick"):

```
for i = 0L, 50000L do begin
  ; stuff here
endfor
```

should do it for you.

Mike
--
www.michaelgalloy.com

Subject: Re: loop limit ???
Posted by cgguido on Wed, 18 Apr 2007 21:45:06 GMT
View Forum Message <> Reply to Message

On Apr 18, 4:54 pm, kostis <kostis...@gmail.com> wrote:
> i wouldnt expect that IDL has a limit for loops !!
> trying a loop of 50.000 steps i got the message:
>
> % Compiled module: TRAJECTORY.
> % Loop limit expression too large for loop variable type.
> <LONG (49999)>.
> % Execution halted at: TRAJECTORY 16 /home/kostis/PROJECT
> LARMOR/dipole/trajectory.pro
> % $MAIN$
>
> Why is this?
> This means i cant have bigger loops... or is there a trick??
>
> Thanx...

Or you could get into the habit of typing

```
Compile_Opt IDL2
```

at the beginning of our programs and never worry about it again.
('downside' is you are forced to use [] for arrays and () only for
functions)

Gianguido

Subject: Re: loop limit ???
Posted by Michael Galloy on Wed, 18 Apr 2007 22:06:52 GMT
View Forum Message <> Reply to Message

On Apr 18, 3:45 pm, Gianguido Cianci <gianguido.cia...@gmail.com>
wrote:
> Or you could get into the habit of typing
>
> Compile_Opt IDL2
>
> at the beginning of our programs and never worry about it again.
> ('downside' is you are forced to use [] for arrays and () only for
> functions)
>
> Gianguido

I don't consider it a downside, but if you only want the default
integer to be 32 bit then use

compile_opt defint32

instead of idl2 (which is just a convenience for both defint32 and
strictarr).

Mike
--
www.michaelgalloy.com

Subject: Re: loop limit ???
Posted by mmeron on Wed, 18 Apr 2007 22:28:08 GMT
View Forum Message <> Reply to Message

In article <1176929681.182929.291930@q75g2000hsh.googlegroups.com>, kostis
<kostiskaz@gmail.com> writes:
> i wouldnt expect that IDL has a limit for loops !!
> trying a loop of 50.000 steps i got the message:
>
> % Compiled module: TRAJECTORY.
> % Loop limit expression too large for loop variable type.
> <LONG (49999)>.
> % Execution halted at: TRAJECTORY 16 /home/kostis/PROJECT
> LARMOR/dipole/trajectory.pro
> % $MAIN$
>
> Why is this?
> This means i cant have bigger loops... or is there a trick??
>

You can have much larger loops, just be careful about the indexing.
In the expression

```
 for i = a, b do ....
```

the type of the loop variable is determined by the type of a
(regardless of what b is). So, if you write

```
 for i = 0, 50000
```

or even

```
 for i = 0, 50000l
```

the loop variable will be a standard (aka "short") integer and these
are limited to 2^15 - 1, i.e. 32767. But if you'll write

```
 for i = 0l, 50000
```

the loop variable is of type LONG and the limit is at 2^31 - 1, more
then 2 billion. If that's not enough you can use LONG64, at which
point it'll probably be the longevity of the computer that'll
determine the limit

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"
