Subject: Need integer consts eq long not short
Posted by sgs on Wed, 02 Nov 1994 20:32:03 GMT
View Forum Message <> Reply to Message

I need pvwave to have long (4 byte) constant integers by default rather than short (2 byte). Is there a way to force this behavior with a system variable, or something?

Short integers cause problems like this:

```
for i=0,n do begin ...    ; buggy, i is a short integer
for i=0L,n do begin ...   ; correct, i is a long integer
```

Pvwave creates i as a short integer, because the constant 0 is, by default, short. However, when (long) n is bigger than 32767, the loop only loops to (n mod 32767).

(1) Unfortunately, these loops often produce reasonable, but incorrect answers.

Many of the std/* and user/* routines provided with pvwave have this bug. (grep for lines with "for i=0", and any lines with integer constants). (Watch out! They work with your small test case, but fail with your real, large dataset, see (1).)

Steve Spray sgs@lmsc.lockheed.com

Subject: Re: Need integer consts eq long not short
Posted by black on Wed, 09 Nov 1994 16:05:37 GMT
View Forum Message <> Reply to Message

While we're on the subject of bugs due to short integers rather than long integers, I might point out another variation of bugs due to this.

This is using the FIX function as opposed to LONG. A example of this is in the PV-WAVE routine SUM. The routine works so long as the output array is not over 32767 elements. I think this particular routine only exists in PV-WAVE - the IDL routine that performs the equivalent operation is TOTAL (note the IDL version of total not the PV-WAVE one). This particular bug is still in SUM in the version of PV-WAVE that we have which is 4.20

Note also that AVG calls SUM, so there's effectivly a bug in AVG too.

John Black

Subject: Re: Need integer consts eq long not short
Posted by landers on Thu, 10 Nov 1994 14:33:04 GMT
View Forum Message <> Reply to Message

In article <39qs0h$g7@ovid.dra.hmg.gb>, black@signal.dra.hmg.gb (John Black) writes:
|>
|> While we're on the subject of bugs due to short integers rather than long
|> integers, I might point out another variation of bugs due to this.
|>
|> This is using the FIX function as opposed to LONG. A example of this is in the
|> PV-WAVE routine SUM. The routine works so long as the output array is not over
|> 32767 elements. I think this particular routine only exists in PV-WAVE - the
|> IDL routine that performs the equivalent operation is TOTAL (note the IDL
|> version of total not the PV-WAVE one). This particular bug is still in SUM in
|> the version of PV-WAVE that we have which is 4.20
|>
|> Note also that AVG calls SUM, so there's effectivly a bug in AVG too.
|>
|> John Black

PV-WAVE has bot TOTAL and SUM procedures. TOTAL just totals the contents of an array, and is an 'intrinsic' function (it's compiled in - there's no total.pro source). SUM is a User's Lib routine that will sum values from a particular array index.

The bug you mention in SUM has been fixed for version 5.00, as well as for the PC version, PV-WAVE Personal Edition.

--
Dave