
Subject: Re: Precision Problem

Posted by [Paul Van Delst\[1\]](#) on Mon, 24 Jul 2006 19:00:03 GMT

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

DMac wrote:

```
> Hello All,  
> Using IDL version 6.2 on a Windows XP machine I am reading in a set of  
> coordinates from a binary file (it is LiDAR data stored in LAS Version  
> 1.0 format) stored in UTM NAD83 Zone 11 and these points are stored as  
> 4 byte long integers in the binary file . The coordinates are stored  
> in the binary file with an offset such that they need to be divided by  
> 100 to obtain the actual coordinates. The binary file is read into a  
> heap variable (Data below). The X and Y coordinates are then written  
> from the heap variable into a double precision array as follows:  
>  
>     x = dblarr(Num_pts)  
>     x = TEMPORARY(data.x*1.000e-002)  
>  
>     y = dblarr(Num_pts)  
>     y = TEMPORARY(data.y*1.000e-002)
```

That is going to give you single precision arrays.

```
IDL> x=dblarr(num_pts)  
IDL> x = TEMPORARY(data.x*1.000e-002)  
IDL> help, x  
X          FLOAT    = Array[100]
```

You need to do

```
IDL> x = TEMPORARY(data.x*1.000d-002)  
IDL> help, x  
X          DOUBLE   = Array[100]
```

i.e. using 1.000d-002 rather than the single precision literal 1.000e-002.

I don't think the TEMPORARY() is doing anything either since the argument is an expression which is pretty much temporary already.

All you need to is:

```
scale_factor=1.0d-02  
x = data.x*scale_factor  
y = data.y*scale_factor
```

to get the double precision arrays at the required size.

As to why your final "y" array is bogus, I don't know. Maybe data.y is somehow bugged up prior to your scaling?. My little tests gave the expected result.

Maybe on Windows if you do a TEMPORARY(data.x*1.000e-002) is whacks the entire structure?
(I hope not)

paulv

--

Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx
Ph: (301)763-8000 x7748
Fax:(301)763-8545
