
Subject: Re: Try this
Posted by [davidf](#) on Tue, 22 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ray Sterner (sterner@tesla.jhuapl.edu) writes:

```
> Try this:  
>  
> x=dindgen(1000)/999.*20.  
> for f=0.,10.,.02 do begin plot,/xstyl,x+f*1D6,sin(x) & empty & endfor
```

Totally cool! :-)

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Try this
Posted by [Karl Krieger](#) on Wed, 23 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 23 Sep 1998, Joe wrote:

```
> But wouldn't they (RSI) have to go out of their way to purposefully  
> convert the doubles to float within plot in order for this to occur?  
> I would have thought that the double precision x-value would have  
> caused an inheritance cascade throughout the plot routine so that  
> everything would be double...
```

I guess that the internal C-code of "plot" uses automatic conversion of the arguments to float as demonstrated in the "Advanced Development Guide". To enable plots of double precision data, you needed either to switch entirely to C double data inside "plot" or duplicate the internal code to handle both data types appropriately.

Karl

--

Max-Planck-Institute for Plasma Physics
Boltzmannstr.2, 85740 Garching, Germany Email: krieger@ipp.mpg.de

Subject: Re: Try this
Posted by [Joe\[2\]](#) on Wed, 23 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
>
> In article <MPG.1071fa8a1e63e2a49896c8@news.frii.com>
> davidf@dfanning.com (David Fanning) writes:
>>
>> Ray Sterner (sterner@tesla.jhuapl.edu) writes:
>>
>>> Try this:
>>>
>>> x=dindgen(1000)/999.*20.
>>> for f=0.,10.,.02 do begin plot,/xstyl,x+f*1D6,sin(x) & empty & endfor
>>
>> Totally cool! :-)
>
> Weird yes, but definitively not cool for those who *want* to plot
> those data points!
>
> It took me a while to see what was going on here, but
> after running Ray's lines, do:
>
> plot,(x+1D7)(0:200),/xstyle,/ystyle,psym=3
>
> Now, a DOUBLE should have more dynamic range than to say that
> e.g., (x+1D7)(50) is *equal* to (x+1D7)(51). Luckily, it does:
>
> IDL> print,(x+1d7)([50,51]),form='(g20)'
> 10000001.00100100
> 10000001.02102102
>
> However, it appears that the PLOT command internally just works
> with FLOAT precision:
>
> IDL> print,FLOAT((x+1d7)([50,51])),form='(g20)'
> 1.000000e+07
> 1.000000e+07
>
> This could have been OK if some "zero point" value (typically the
> value of one of the tickmarks) had been subtracted in the process of
> converting to float. As it stands, this will have to be done
> by the user/programmer.
>
> Regards,
>
> Stein Vidar
```

But wouldn't they (RSI) have to go out of their way to purposefully convert the doubles to float within plot in order for this to occur? I would have thought that the double precision x-value would have caused an inheritance cascade throughout the plot routine so that everything would be double. Hmmmm... Oh wait, I got it! The !x and !y structures are not double precision in their .range (and other related) fields so it won't matter. This has to be done for speed reasons since to make them double would force all plots to use double precision calculations along with the attendant decrease in computational rate. Still it is not obvious why they need to force the x (and presumably y) variables to be float-type.

Z

Subject: Re: Try this
Posted by [steinhh](#) on Wed, 23 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <MPG.1071fa8a1e63e2a49896c8@news.frii.com>
davidf@dfanning.com (David Fanning) writes:
>
> Ray Sterner (sterner@tesla.jhuapl.edu) writes:
>
>> Try this:
>>
>> x=dindgen(1000)/999.*20.
>> for f=0.,10.,.02 do begin plot,/xstyl,x+f*1D6,sin(x) & empty & endfor
>
> Totally cool! :-)

Weird yes, but definitively not cool for those who *want* to plot those data points!

It took me a while to see what was going on here, but after running Ray's lines, do:

```
plot,(x+1D7)(0:200),/xstyle,/ystyle,psym=3
```

Now, a DOUBLE should have more dynamic range than to say that e.g., (x+1D7)(50) is *equal* to (x+1D7)(51). Luckily, it does:

```
IDL> print,(x+1d7)([50,51]),form='(g20)'  
10000001.00100100  
10000001.02102102
```

However, it appears that the PLOT command internally just works with FLOAT precision:

```
IDL> print,FLOAT((x+1d7)([50,51])),form='(g20)'  
1.000000e+07  
1.000000e+07
```

This could have been OK if some "zero point" value (typically the value of one of the tickmarks) had been subtracted in the process of converting to float. As it stands, this will have to be done by the user/programmer.

Regards,

Stein Vidar

(OK, so the visual effect **was** kind of cool :-)
