Subject: Re: Converting Doubles to Strings
Posted by biophys on Sat, 05 Nov 2005 10:40:49 GMT
View Forum Message <> Reply to Message

I had thought of this before when I was writing a widget. My understanding is that this so-called natural width is only meaningful when you are reading ascii data. In other words, the natural width is always tied to the digital/string representation of a number. However, the problem with string() function is that it converts numerical data in its binary representation into string. For example, when we issue the following command to idl.

IDL>print,String(-22.123, Format='(D0)')
-22.122999

you provide a string representation of a number which is '-22.123', idl will read it and figure out it is a floating number and convert that into its binary representation, which rounds to -22.122999, and pass it to function string(), and then it convert it into the string ' -22.122999'. Similarly if you try the following command,

IDL>print,String(-22.123d0, Format='(D0)')
-22.123000

Now idl knows it is a double precision and converts it into 8 bytes double data and then converts it into string with the natural width of the converted double, which is '-22.123000'.

But if you really provide more than 6 digits after decimal point, you will see what david was showing. Natural width seems to be fixed to 6 digits after decimal point no matter whether it is float or double.

IDL> print,String(-22.1234567890, Format='(D0)')
-22.123457
IDL> print,String(-22.1234567890d0, Format='(D0)')
-22.123457

if you use full width, you will get

IDL> print,String(-22.1234567890, Format='(D)')
-22.1234570
IDL> print,String(-22.1234567890d0, Format='(D)')
-22.1234567889999987

So the bottomline is, numbers in digits are always rounded and stored in binary representation in computer. Which means exact digital representation of float/double numbers only exist in some "string

representation". Once the number is taken by the machine and being stored in its closest binary representation you lose all your width information of its original digital representation. And if later you want to have some digital representation, then you have to specify the width. Otherwise, you can either use F0/D0 to get 6 digits after decimal point or use F/D to get the closest full width digital/string representation. I end up using the latter for my widgets.

Please correct me if I'm wrong.

Thx

David Fanning wrote:
> Folks,
>
> I've run into an interesting problem. I have a double precision
> number that I wish to convert to a string (so I can put it into
> a text widget, for example). I don't know ahead of time how many
> significant digits will be in this number. The number could
> look like this 45.6, or this 123456789.123456789, or this
> -22.1234567890. If it looks like the latter number, I am
> having a very hard time converting it to a string. For example,
> this doesn't work:
>
> IDL> v = String(-22.1234567890, Format='(D0)')
> IDL> Print, v
> -22.123457
>
> Is this a bug in IDL? Or am I overlooking something?
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.,
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/