Subject: Re: Forcing READ_ASCII output to be a set of strings...
Posted by JD Smith on Fri, 18 Mar 2005 23:11:46 GMT
View Forum Message <> Reply to Message

On Mon, 14 Mar 2005 16:15:43 -0600, Michael Wallace wrote:

>> The reason I thought this would be better as a text default is that you can
>> easily go from string -> number, but you can't go backwards. Considering
>> the input is text, why would they just assume it is filled with floating
>> point numbers? It appears to be completely arbitrary... A string format
>> appears to be the most "generalized" form you could use.
>
> The reason you assume that it's filled with numbers is because it most
> likely is. (Don't you just love circular logic? ;-) ) IDL is built
> around *data* manipulation. We use text files all the time in my
> industry and aside from the header, if present, the contents are all
> numbers. Some of the numbers are integers and some are floating point.
> Obviously, a floating point type covers both cases.

Actually, it doesn't. This is an aside, but you may have wondered how
a little old floating point number which fits in just four bytes could
manage to squeeze all those numbers from 1.17549e-38 to 3.40282e+38 in
there, when a 4 byte integer (aka LONG) only holds from -2147483648 to
+2147483647, and no decimals, to boot. The secret is, it doesn't.
Not by a long shot. Consider the range of 1 trillion integers:

10000000000000000000ULL - 10000001000000000000ULL

How many unique floating point value do you have inside of this range?
None actually. Any such value will be rounded up or down outside of
this range. So if you are using integers to count the quantity of
sand grains in a sandpile, this may not matter much, but if you are
using them to represent an exact serial code or combination to your
bank vault, then this distinction is very important.

JD