Subject: Re: Forcing READ_ASCII output to be a set of strings...
Posted by Michael Wallace on Mon, 14 Mar 2005 22:15:43 GMT
View Forum Message <> Reply to Message

> 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. Text files have the nice advantage in that you can look at the contents of the file without needing to load the files into any program. Just because we happen to store data within a text file doesn't mean that it should be interpreted as text.

I'm not just talking about one particular project here, but we create ASCII representations of a lot of our data on many of our projects. Sure, it's not the primary form we store the data in for the long term, but it's great for our scientists ad hoc work, especially since our scientists seem to be allergic to netCDF, CDF, HDF and the like.

Another issue is that it'd be terrible performance-wise to read in a bunch of values only to convert them to numeric types. And most of the time, your CSV or other files will be numeric in nature. Having strings in there is a special case. Maybe the change should be to make templates easier to modify rather than change the fundamental nature of read_ascii(). I just don't buy the explanation that the majority of folks will have ASCII files of text rather than ASCII files of numbers. You can try to convince me otherwise, but I don't know that you'll get very far. ;-)

I feel pretty dirty now that I'm actually defending something in the language rather than railing against something in the language, which is my typical. ;-)

-Mike