Subject: Re: RFC 1: Common functions for beginners
Posted by Jeff Guerber on Fri, 16 May 2003 01:38:42 GMT
View Forum Message <> Reply to Message

On Wed, 14 May 2003, Ed Wright wrote:

> I learned the basics of IDL use and programming over a long time interval
> while writing the magnum opus of dlms. Rather than have new users repeat my
> learning process, I'd like your suggestions for the twenty or so most common
> IDL functions a new user should understand.
>
> I mean most basic-most used. Open file, read numbers, plot numbers, write
> text, close file.

1) Not really a function or procedure, but fundamental to making good use of IDL: Vectors, arrays, and operations on them, including in particular vector-vector, vector-array, and array-array operations, in addition to those involving scalars. Especially for those who cut their teeth on C or who haven't upgraded their Fortran compiler in 25 years! :-) As a subtopic here, vector indexing and WHERE. (Someone mentioned SORT and UNIQ along with WHERE, but I find I use those _much_ less often. Also, I think REBIN and HISTOGRAM, which Pavel mentioned, are much more advanced topics.)

2) PLOT. After all, it's pretty much IDL's raison d'... d'... oh phooey... reason for existence!

3) _Correct_ use of AND, OR, and NOT in making logical comparisons. As bitwise operators, this isn't always obvious to the uninitiated.

4) On a somewhat more advanced level (so probably further down than #4 in the final list), pointers. Since IDL's pointer facility is so different from C's, it's something that users often find confusing when they first try them.

5) Something I've rarely seen discussed, but which might be a really good topic: When NOT to use IDL! Heretical as it may sound, it's not always the best choice... or even a good one. For example, I once had the misfortune to work on a project where we were archiving large quantities of data from Exabyte tapes onto video disks, doing various validations along the way. The Civil Servant in charge insisted on using IDL for the processing pipeline. There was no way it could keep up with the flood! A Fortran version I coded up ran in 1/2 to 2/3 the time, and he still wouldn't let me use it. Don't get me wrong, IDL's terrific for many things... but this wasn't one of them!

Jeff Guerber