
Subject: Application programming--missing features

Posted by [jdlb](#) on Fri, 16 Apr 1993 03:35:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

Though I love IDL and commend David Stern et al. for their fine work, I feel deprived of two important features which are important for application programming. I'd like to see them implemented, or failing that to be shown an elegant (i.e., non-tedious) way to simulate them.

1) Adding a "wrapper" to an intrinsic IDL routine is difficult.

For example, consider William Thompson's <thompson@serts.gsfc.nasa.gov> just-posted routine PLOT_DROP for dropping bad data values when plotting data. That procedure, in effect, just adds a single keyword DROP_VALUE to the generic PLOT routine.

In such an application, for each of the usual optional parameters accepted by PLOT one must make an entry in the procedure declaration and properly pass the parameter to PLOT. This involves either (a) defining defaults for each option or (b) tediously building up a command line and passing it to the EXECUTE function.

A solution might be to have IDL parse the argument list and separate out those parameters which are not defined in the procedure declaration, placing them in another list. Then PLOT could be called with this list as a parameter and do its own parsing of the new command line. This requires that IDL not balk at unrecognized arguments.

2) User-defined global variables for customizing program behavior do not exist.

Consider, for example, a routine which displays an image, first magnifying the image by a factor ZOOM using REBIN. ZOOM is set with an optional input parameter and has a default value specified in the program. It would be nice to first check if the user had defined his or her favorite value of ZOOM (say in .idl_startup). Other routines might also use this same ZOOM.

The only two options are to (a) define a new system variable using DEFSYSV or (b) use common blocks. Option (a) fails because N_ELEMENTS(!FOO) returns an error ("Not a legal system variable") instead of zero if !FOO is undefined. Option (b) is very tedious for both programmer and user because common blocks are finicky beasts.

The simplest solution would be to modify IDL such that n_elements(!foo) returns 0 if !foo is undefined.

Have I unfairly maligned the otherwise-delightful IDL in claiming these features are missing? Am I the only one who would find them useful? Any comments would be welcomed.

Regards,
Jeff

% Jeff de la Beaujardiere % jdlb@mamane.ifa.hawaii.edu %
% Institute for Astronomy % 808-956-9843 %
% University of Hawai'i % fax 956-9402 %
