
Subject: IDL -> Fortran: Summary

Posted by [dlmatt](#) on Thu, 24 Aug 1995 07:00:00 GMT

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

This summary is condensed; email dlmatt@uss-enterprise.bu.edu for possible further details. I have added my own words to some of the quotations, so that the persons cited should not be held responsible for them.

John Davis (davis@space.mit.edu) points out that IDL hides many details of memory management that you must worry about when working with a compiled language and using its memory management capabilities (IDL is an interpreted language, even though it says it compiles). In general the data structures are different in IDL and Fortran -- one is row-major and the other column-major, just to start. One must really find out what the actual algorithm was in IDL and then program it in Fortran, essentially from scratch. Finding out what the algorithm is or what the program does may be harder in IDL because experienced IDL programmers use many tricks to vectorize the code as much as possible. (Also, in my experience at least, IDL programs are even less well documented than those written in Fortran.) Davis has nevertheless taken a fair-sized IDL procedure, found out what it was supposed to do, and written a C program to do the same thing -- not exactly a translation/conversion/port.

Mitchell Grunes (grunes@news.nrl.navy.mil) adds: The problem is almost insoluble since compiled languages require compile-time information about symbols, whereas in IDL many symbols are read in from the user's workspace. Each such symbol reference would have to be dealt with individually by means of DATA or PARAMETER statements, file reads, etc., in subroutines or the main program. Not only that, symbols change their meanings dynamically in IDL: A can start off as an integer scalar, change to a double precision 2D array, and then become the name of a procedure!

On the other hand D. Huenemoerder (dph@space.mit.edu) says that PVwave /Advantage has translators to Fortran, C, and Maple. Hugues Sicotte (sicotte@pupgg.princeton.edu) says it's feasible to port IDL by hand to matlab, which has some similarities to IDL, and also has the advantage of being available free in the form of GNU Octave.

I suppose that applying a translation program could help -- IDL is full of vectorizations or implied do loops which a good translator ought not to miss -- but there seems little hope of the translator doing almost all the work, as, e.g., in translating Fortran to C using f2c.

David Matthews, Center for Space Physics, Boston University
dlmatt@uss-enterprise.bu.edu
