

NAME

dtos – convert Fortran program from double-precision to single-precision

SYNOPSIS

dtos [--?] [--author] [--copyright] [--help] [--version] <infile>outfile

dtos copies its standard input to standard output, converting Fortran double-precision constants, built-in functions, and type declarations to single precision.

Floating-point FORMAT specifications are left intact; on some ancient systems, they may require modifications. They do *not* under the rules of Fortran 77.

Leading tabs are correctly interpreted according to common extended Fortran rules.

dtos recognizes all of the standard Fortran 77 single- and double-precision functions, as well as the pair **rand/drand** (UNIX pseudo-random number generators), and the pair **r1mach/d1mach** from the PORT library framework.

dtos's other purpose is to demonstrate a modest **lex(1)** program.

OPTIONS

Options can be prefixed with either one or two hyphens, and can be abbreviated to any unique prefix. Thus, **-v**, **-ver**, and **--version** are equivalent.

All options in this program are diagnostic, and suppress processing of the input stream. Execution terminates with a success return code after processing one or more options, but unrecognized options cause immediate termination with a failure return code.

- ?** Same as **--help**.
- author** Display a brief author credit on *stdout*.
- copyright** Display copyright and license information on *stdout*.
- help** Display a brief help message on *stdout*, giving a usage description.
- version** Display the program version number and release date on *stdout*.

BUGS

Undeclared variables are not type-converted. To find such instances, use the Extended PFORT Verifier, **pfort(1)**, or the Fortran checker, **ftnchek(1)**. Some UNIX Fortran compilers have a compile-time option, usually called **-u**, to flag undeclared variables.

Text beyond column 72 is discarded when lines are collected into Fortran statements.

dtos does not handle embedded ASCII tab characters correctly when long lines are to be broken. A Fortran-sensitive detabbing utility should be applied first if the input file possibly contains embedded tabs. Note that **expand(1)** *cannot* be used to do this job correctly!

Mixed-precision code may not be converted correctly. For example, **DBLE(FLOAT(N))** will become **SNGL(FLOAT(N))**, which is syntactically incorrect.

Functions and variables of type **COMPLEX** are not converted, because Fortran 77 does not define a double precision complex type. Complex constants will be converted, however, since their real and imaginary parts look like normal floating-point values.

SEE ALSO

ftnchek(1), **lex(1)**, **pfort(1)**, **stod(1)**.

AUTHOR

Nelson H. F. Beebe
 Department of Mathematics
 University of Utah
 Salt Lake City, UT 84112-0090
 USA
 Tel: +1 801 581 5254

FAX: +1 801 581 4148

Email: beebe@math.utah.edu, beebe@acm.org, beebe@computer.org

Web: <http://www.math.utah.edu/~beebe/>