F2C for VAX/VMS

Date of last revision: June 4, 1990

This document describes the use of **f2c**, a filter for translating input files which are Fortran-77 code to either C or C++ code. This document may be printed on the line printer in CPE-3.102 by entering the command;

DOCPRINT SYS$DOCUMENT:F2C.DOC

Copies of this document may be printed on the laser printer in CPE-3.159 by entering the command;

PSPRINT SYS$DOCUMENT:F2C.PS

You may view copies of this document on workstations or personal computers which recognize the *X* window system protocol by entering the command;

XDVI SYS$DOCUMENT:F2C.DVI

Description

F2c converts Fortran 77 source code in *file* to C (or C++) a source file in the current directory, with *.c* substituted for the file type extension. If no Fortran files are named, **f2c** reads Fortran from standard input and writes C on standard output. *File* names that end with **.P** are taken to be prototype files, as produced by option *-P*, and are read first.

Format

f2c [*option* ...] *file*

f2c is compiled with VAXC which converts uppercase command line arguments to lowercase. To pass an uppercase argument or a negation argument to **f2c**, the argument must be enclosed in quotation marks (i.e. "-C" or "-!bs"). Arguments enclosed in brackets ([ ]) are optional.

Options

The following options have the same meaning as in *f77*(1) on *Unix* systems.

-C Compile code to check that subscripts are within declared array bounds.

-I2 Render INTEGER and LOGICAL as short, INTEGER*4 as long int. Assume the default F2CLIB.OLB library: allow only INTEGER*4 (and no LOGICAL) variables in INQUIREs. Option -I4 confirms the default rendering of INTEGER as long int.

-onetrip Compile DO loops that are performed at least once if reached. (Fortran 77 DO loops are not performed at all if the upper limit is smaller than the lower limit.)

-U Honor the case of variable and external names. Fortran keywords must be in *lower* case.

-u Make the default type of a variable 'undefined' rather than using the default Fortran rules.

-w Suppress all warning messages. If the option is `-w66`, only Fortran 66 compatability warnings are suppressed.

The following options are peculiar to **f2c**.

-A Produce ANSI C. Default is old-style C.

-a Make local variables automatic rather than static unless they appear in a "DATA, EQUIVALENCE, NAMELIST," or SAVE statement.

-C++ Output C++ code.

-c Include original Fortran source as comments.

-E Declare uninitialized COMMON to be `Extern` (overridably defined in `f2c.h` as `extern`).

-ec Place uninitialized COMMON blocks in separate files: COMMON /ABC/ appears in file `abc_com.c`. Option `-e1c` bundles the separate files into the output file, with comments that give an unbundling *sed*(1) script.

-ext Complain about *f77*(1) extensions.

-g Include original Fortran line numbers as comments.

-h Try to align character strings on word (or, if the option is *-hd* , on double-word) boundaries.

-i2 Similar to `-I2`, but assume a modified F2CLIB.OLB library (compiled with `-Df2c_i2`), so INTEGER and LOGICAL variables may be assigned by INQUIRE.

-kr Use temporary values to enforce Fortran expression evaluation where K&R (first edition) parenthesization rules allow rearrangement. If the option is *-krd*, use double precision temporaries even for single-precision operands.

-P Write a *file*`.P` of ANSI (or C++) prototypes for procedures defined in each input *file*`.f` or *file*`.F`. When reading Fortran from standard input, write prototypes at the beginning of standard output. Implies `-A` unless option `-C++` is present. Option `-Ps` implies `-P`, and gives exit status 4 if rerunning **f2c** may change prototypes or declarations.

-p Supply preprocessor definitions to make common-block members look like local variables.

-R Do not promote REAL functions and operations to DOUBLE PRECISION. Option `-!R` confirms the default, which imitates *f77*.

-r8 Promote REAL to DOUBLE PRECISION, COMPLEX to DOUBLE COMPLEX.

-T*dir* Put temporary files in directory *dir*.

-w8 Suppress warnings when COMMON or EQUIVALENCE forces odd-word alignment of doubles.

-W*n* Assume *n* characters/word (default 4) when initializing numeric variables with character data.

-z Do not implicitly recognize *DOUBLE COMPLEX*.

-!bs Do not recognize *b*ack*s*lash escapes (\", \', \0, \\, \b, \f, \n, \r, \t, \v) in character strings.

-!c Inhibit C output, but produce **-P** output.

-!I Reject `include` statements.

-!it Don't infer types of untyped EXTERNAL procedures from use as parameters to previously defined or prototyped procedures.

-!P Do not attempt to infer ANSI or C++ prototypes from usage.

The resulting C invokes the support routines of *f77*; object code should be linked with the **f2c** support library as well as the VAXC run-time library. The appropriate *LINK* command under VMS would be;

$ LINK *file*,**SYS$PUBLIC:F2CLIB/LIB,SYS$LIBRARY:VAXCRTL/LIB**

Calling conventions are those of *f77*: see the reference below.

Files

file.[**f**][**F**][**for**][**FOR**]	input file
***.c**	output file
SYS$PUBLIC:F2C.H	header file
SYS$PUBLIC:F2CLIB.OLB	intrinsic function library and Fortran I/O library
SYS$LIBRARY:VAXCRTL.OLB	C run-time support library

See Also

S. I. Feldman and P. J. Weinberger, 'A Portable Fortran 77 Compiler', *UNIX Time Sharing System Programmer's Manual*, Tenth Edition, Volume 2, AT&T Bell Laboratories, 1990.

You should also read the technical report on **f2c** from AT&T which provides a more detailed explanation of **f2c**. This file is normally found in SYS$DOC:F2C_TECHNICAL_REPORT.PS.

Diagnostics

The diagnostics produced by **f2c** are intended to be self-explanatory.

Bugs

Floating-point constant expressions are simplified in the floating-point arithmetic of the machine running **f2c**, so they are typically accurate to at most 16 or 17 decimal places.

Untypable *EXTERNAL* functions are declared **int**.