NAME

ftnchek – Fortran program checker

SYNOPSIS

ftnchek [–**arguments=***num*] [–**array=***num*] [–**[no]calltree**] [–**columns=***num*]
[–**common=***num*] [–**[no]crossref**] [–**[no]declare**] [–**[no]division**]
[–**[no]extern**] [–**[no]f77**] [–**[no]help**] [–**[no]hollerith**] [–**include=***str*]
[–**[no]library**] [–**[no]list**] [–**[no]novice**] [–**output=***str*] [–**[no]portability**]
[–**[no]pretty**] [–**[no]project**] [–**[no]pure**] [–**[no]reference**] [–**[no]sixchar**]
[–**[no]sort**] [–**[no]symtab**] [–**[no]truncation**] [–**usage=***num*]
[–**[no]verbose**] [–**[no]volatile**] [–**wordsize=***num*] [–**wrap=***num*] [*files ...*]

INTRODUCTION

ftnchek (short for Fortran checker) is designed to detect certain errors in a Fortran program that a compiler usually does not. **ftnchek** is not primarily intended to detect syntax errors. Its purpose is to assist the user in finding semantic errors. Semantic errors are legal in the Fortran language but are wasteful or may cause incorrect operation. For example, variables which are never used may indicate some omission in the program; uninitialized variables contain garbage which may cause incorrect results to be calculated; and variables which are not declared may not have the intended type. **ftnchek** is intended to assist users in the debugging of their Fortran program. It is not intended to catch all syntax errors. This is the function of the compiler. Prior to using **ftnchek**, the user should verify that the program compiles correctly.

INVOKING FTNCHEK

ftnchek is invoked through a command of the form:

```
$ ftnchek [-option -option ...] filename [filename ...]
```

The brackets indicate something which is optional. The brackets themselves are not actually typed. Here options are command-line switches or settings, which control the operation of the program and the amount of information that will be printed out. If no option is specified, the default action is to print error messages, warnings, and informational messages, but not the program listing or symbol tables.

Each option begins with the '–' character. (On VAX/VMS or MS-DOS systems you may use either '/' or '–'.) The options are described at greater length in the next section.

ftnchek options fall into two categories: switches, which are either true or false, and settings, which have a numeric or string value. The name of a switch is prefixed by 'no' to turn it off: e.g. **–nopure** would turn off the warnings about impure functions. The 'no' prefix can also be used with numeric settings, having the effect of turning off the corresponding warnings. Only the first 3 characters of an option name (not counting the '–') need be provided.

The switches and settings which **ftnchek** currently recognizes are:

–**arguments=***n*
: Set level of strictness in checking subprogram arguments. Min is 0 (no checking). Max is 3 (most checking). Default = 3.

–**array=***n*
: Set level of strictness in checking array arguments of subprograms. Min is 0 (least strict). Max is 3 (most strict). Default = 3.

–**calltree**
: Print tree of subprogram call hierarchy. Default = no.

–**columns=***n*
: Set maximum line length to *n* columns. (Beyond this is ignored.) Max is 132. Default = 72.

–**common=***n*
: Set level of strictness in checking COMMON blocks. Min is 0 (no checking). Max is 3 (must be identical). Default = 3.

−crossref
Print cross-reference list of subprogram calls and COMMON block use. Default = no.

−declare
Print a list of all identifiers whose datatype is not explicitly declared. Default = no.

−division
Warn wherever division is done (except division by a constant). Default = no.

−extern Warn if external subprograms which are invoked are never defined. Default = yes.

−f77 Warn about extensions to the Fortran 77 standard. Default = no.

−help Print command summary. Default = no.

−hollerith
Warn about Hollerith constants if **−portability** option is in effect. Default = yes.

−include=*path*
Define a directory to search for INCLUDE files. Cumulative.

−library
Begin library mode: do not warn about subprograms in file that are defined but never used. Default = no.

−list Print source listing of program. Default = no.

−novice Give output suitable for novice users. Default = yes.

−output=*filename*
Send output to the given file. Default is to send output to the screen. (Default filename extension is . *lis*).

−portability
Warn about non-portable usages. Default = no.

−pretty Give warnings for possibly misleading appearance of source code. Default = yes.

−project
Create project file (see explanation below). Default = no.

−pure Assume functions are pure, i.e. have no side effects. Default = yes.

−reference
Print table of subprograms referenced by each subprogram. Default = no.

−sixchar
List any variable names which clash at 6 characters length. Default = no.

−sort Print list of subprograms sorted in prerequisite order. Default = no.

−symtab
Print symbol table for each subprogram. Default = no.

−truncation
Check for possible loss of accuracy by truncation. Default = yes.

−usage=*n*
Control warnings about unused variables, etc. Min is 0 (no checking). Max is 3 (most checking). Default = 3.

−verbose
Produce full amount of output. Default = yes.

−volatile
Assume COMMON blocks lose definition between activations. Default = no.

–wordsize=*n*

Set the default word size for numeric quantities to *n* bytes. Default = 4 bytes.

–wrap=*n*

Set output column at which to wrap long error messages and warnings to the next line. If set to 0, turn off wrapping. Default = 79.

When more than one option is used, they should be separated by a blank space, except on systems such as VMS where options begin with slash (/). No blank spaces may be placed around the equals sign (=) in a setting. `ftnchek "?"` will produce a command summary listing all options and settings.

When giving a name of an input file, the extension is optional. If no extension is given, **ftnchek** will first look for a project file with extension *.prj*, and will use that if it exists. If not, then **ftnchek** will look for a Fortran source file with the extension *.for* for VMS systems, *.f* for UNIX systems. More than one file name can be given to **ftnchek**, and it will process the modules in all files as if they were in a single file.

Wildcards are allowed in the specification of filenames on the command line for the VMS and MS-DOS versions, as also of course under UNIX and any other system that performs wildcard expansion in the command processor.

If no filename is given, **ftnchek** will read input from the standard input.

INSTALLATION AND SUPPORT

The **ftnchek** program can be obtained by anonymous ftp from many software servers, including host `netlib.att.com` (192.20.225.252) where it is located in directory `/netlib/fortran`. It can also be obtained via electronic mail by sending the message: ‘‘`send ftnchek from fortran`’’ to the Internet address: `netlib@netlib.ornl.gov`.

Installation requires a C compiler for your computer. Executable binary in ZIP format for IBM PC computers under MS-DOS is available by anonymous ftp from `wuarchive.wustl.edu` (128.252.135.4) where it is located in directory `/mirrors/msdos/fortran`, filename `ftnchk27.zip`. Executable binary in binhexed stuffit format for Macintosh computers is available from sumex-aim.stanford.edu (36.44.0.6), in directory `info-mac/Development`, filename `ftnchek-27.hqx`.

ftnchek has a number of fixed internal parameters that affect its operation, such as the symbol table sizes and support for various extensions and particular operating systems. They are implemented as C language macro names. If you wish to change a parameter defined by one of these macros, it is necessary to recompile **ftnchek** with the macro definition changed to the desired value. See the `README` file for a list of these macro names and instructions for defining them according to your preferences.

Information about the latest version and the status of the project can be obtained by the Internet command ‘‘`finger ftnchek@mary.fordham.edu`’’. For further information and to report bugs, you may contact Dr. Robert Moniot at either of the following network addresses:

`MONIOT@FORDMULC.BITNET`
`moniot@mary.fordham.edu`