

DISCLAIMER:

The following compiler options are extracted from the *Fujitsu User's Guide*, Copyright© FUJITSU LIMITED 1999-2000, All Rights Reserved. The extracted text has not been edited by Lahey. These options offer Lahey users more control than the published Lahey options. Note that some the text is specific to Fujitsu (not Lahey/Fujitsu) compilers (for example, "frt" is the name of a Fujitsu compiler driver). Not all options will work with the Lahey/Fujitsu Win32 or Linux compilers. We will add relevant options to *the Lahey User's Guide* at the next revision.

You can pass these options to the Lahey/Fujitsu Win32 compiler driver, lf95, by specifying

```
-lfe "--<option>"
```

on the command line or in a response file. For example,

```
lf95 demo.f90 -lfe "-CcdRR8"
```

You can pass these options to the Lahey/Fujitsu Linux compiler driver, lf95, by specifying the options as documented.

Option List

```
[ -c ] [ -f msg_lvl ] [ -g ] [ -lname ] [ -ml=target ] [ -mldefault=target ] [ -o exe_file ]  
[ -shared ] [ -static-flib ] [ -v msg_lvl ] [ -w ] [ -x incl_arg ]  
[ -A obj_arg ] [ -C typ_arg ] [ -D name[=tokens] ] [ -E msg_arg ] [ -Fixed ] [ -Free ]  
[ -H dbg_arg ] [ -I directory ] [ -K opt_arg ] [ -L directory ] [ -M directory ]  
[ -N src_arg ] [ -O opt_lvl ] [ -P ] [ -Q lst_arg ] [ -S ] [ -U name ] [ -V ]  
[ -Wtool,arg1[,arg2, ...] ] [ -X lan_lvl ]
```

-c

The -c (compile only) option suppresses calling the link-edit phase. If the -c option is specified, object programs are created, but an executable program is not created.

-f msg lvl msg_lvl: { i | w | s | msg_num }

The -f option specifies the lowest error level of diagnostic messages output for Fortran compiler errors. Each diagnostic message has a prefix of the form jwdxxxxi-y message, where xxxx is the message serial number, and y is the message error level. The y can be either i (information message), w (warning message), s (serious error), or u (unrecoverable error). If the -f option is omitted, it defaults to -fi. If -f msg_num is specified, the particular error message is suppressed. The msg_num argument can coexist with other argument. To specify msg_num and other arguments for the -f option, separate the arguments with a comma (,).

Example

```
frt -fw,1040,2005 a.f95
```

i

The -fi option causes the Fortran compiler to output all levels of diagnostic messages.

w

The `-fw` option causes the Fortran compiler to output diagnostic messages only for error levels w, s, and u.

s

The `-fs` option causes the Fortran compiler to output diagnostic messages only for error levels s and u.

msg_num

The `-fmsg_num` suppresses particular information and warning diagnostic messages specified by the *msg_num* argument. The *msg_num* argument is the four-digit message number; more than one *msg_num* may be specified. The *msg_num* argument can coexist with all of the options, but is ignored when `-fs` is specified.

-g

The `-g` option generates debugging information in object programs. This information is used by the debugger. The user can perform source-level debugging of programs compiled this way. If both the `-g` and `-O` options are specified, the `-O` option is ignored.

The `-g` option affects only Fortran source files compiled by the `frt` command. It does not affect `.o`, `.s`, or other files.

-lname

The `-l` option searches the library *libname.so* or *libname.a*. The position of this option within the command line is important. This is because libraries are searched for in the order in which the other libraries and object files appear within the command line. This option and its argument are passed to the linker.

`-ml=target target: { cdecl | frt }`

Changes the external procedure name according to *target*. 'cdecl' or 'frt' can be specified for *target* on the Linux system.

`-mldefault=target target: { cdecl | frt }`

Changes the default external procedure name according to *target*. 'cdecl' or 'frt' can be specified for *target* on the Linux system.

`-o exe_file`

If the `-c` option is not specified, the `-o` option specifies the file name of the executable program. This information is specified as the argument *exe_file*. If both the `-o` and `-c` options are specified, an object program is created with the name *exe_file*. If both the `-o` and `-S` options are specified, an assembly program is created with the name *exe_file*.

`-shared`

If the `-shared` option is specified, create a shared library. The `-shared` option is used at linking time.

`-static-flib`

The `-static-flib` option specify to link Fortran Library statically. Dynamic linking for Fortran Library is default. The `-static-flib` option is used at linking time.

-v *msg_lvl* *msg_lvl* : { 90o | 95d | 95e | 95o | 95s }

The -v option diagnoses whether a source code conforms to a specific Fortran standard. More than one of these options may be specified, separated by commas.

90o

The -v90o option issues diagnostic messages if the source code contains any obsolescent features in Fortran 90.

95d

The -v95d option issues diagnostic messages if the source code contains any deleted features in Fortran 95.

95e

The -v95e option issues diagnostic messages if the source code contains any Fortran 95 features not also in Fortran 90.

95o

The -v95o option issues diagnostic messages if the source code contains any obsolescent features in Fortran 95.

95s

The -v95s option issues diagnostic messages if the source code contains any nonstandard features, other than decremental features in Fortran 95.

-w

The -w option compiles source programs with the length of each line extended to 255 characters in fixed source form.

-x *inl_arg* *inl_arg* : { - | *pgm_nm* [, *pgm_nm* [, ...]] | *stm_no* | *dat_szK* }

The -x option creates an object program by applying inline expansion of user-defined external and internal procedures at the point where they are referenced. The -x option improves the execution performance of an object program above the level obtained by using the -K option. If the -x option is specified with the -O0 option, the -x option is ignored. To specify more than one argument for the -x option, separate the arguments with a comma (.).

These optimization functions increase the time and memory needed for compilation. Also, the more validating the optimization function has to do, the greater will be the size of the object program. To determine whether these optimization functions were applied, specify the option -Ei.

An external procedure in the same input source file as the program unit calling the procedure and internal procedure are subject to these optimization functions. The optimization is not applied to references to procedures in other source files.

In this version of the compiler, there is no attempt to inline module procedures.

-

The -x- option creates an object program by applying inline expansion of user-defined external and internal procedures having 30 or fewer executable statements. These statements are expanded at the reference point.

The -x- option must not be used with any other -x option.

pgm_nm[, ***pgm_nm***[, ...]]

The *-xpgm_nm*[, *pgm_nm*[, ...]] applies inline expansion for only those user-defined external and internal procedures specified in *pgm_nm* arguments. Specifying only the names of frequently called procedures can significantly reduce compilation time and memory requirements. Use a comma to separate *pgm_nm* from other arguments of the -x option specified at the same time.

Specify host name + '.' + internal procedure name for internal procedure, and specify module name + '.' + module procedure name + '.' + internal procedure name.

Example: Specifying internal procedure name

```
frt -xfoo.insub a.f95
```

stm_no

The *-xstm_no* option creates an object program by applying inline expansion of user-defined external and internal procedures having *stm_no* or fewer executable statements. The default is 30.

dat_szK

The *-xdat_szK* option specifies an upper bound on the size of local arrays in inlined procedures. The value specified for *dat_sz* is a number from 1 to 2147483. The letter K, which represents a kilobyte, must be added after the value of *dat_sz*. The default *dat_sz* is unlimited.

The *-xdat_szK* option applies inline expansion of user-defined external and internal procedures in which the size of local arrays is less than the specified *dat_szK* value.

If optimization for inline expansion of user-defined procedures is applied, the resulting object program includes the data referenced in the user-defined procedure. If this data includes a huge local array, the resulting object program can be quite large. To control this, specify the *-x dat_szK* option.

-A *obj_arg* *obj_arg* : { A | E | R | r | T | U | d | i | m | p | q | w | y | z | { 0 | 1 | 2 | n } }

A

The -AA option controls the alignment of variables in common blocks. When the A argument is specified, such variables are not forced to be aligned on the correct boundaries.

E

The -AE option specifies that the backslash (\) character is not used as an escape for control characters in character constants. For example, the string "ABC\\DE" has seven characters, including two backslashes; if the -AE option is not set, the string has six characters (only one backslash).

R

The -AR option specifies that constants are allocated in an unwritable object area. It prevents incorrect execution by overwriting a constant, and execution is stopped when an attempt is made to overwrite a constant.

r

The -Ar option doesn't specify that constants are allocated in an unwritable object area. The -Ar is the default.

T

The -AT option disables Fortran implicit typing and makes the default type for all variables undefined. This option does not affect any IMPLICIT statements.

U

The -AU option indicates that names are to be interpreted in a case-sensitive manner. Be careful of the following:

- This causes the program to be interpreted in a nonstandard manner. In ISO standard Fortran, letters are not case sensitive.

- References to service subroutine or service function must be in lowercase letters.

- The spelling of all declarations and references to an intrinsic procedure must be the same.

- In the IMPLICIT statement, the lowercase letters specified in a letter specifier list are equivalent to the corresponding uppercase letters.

- When you debug using a symbolic debugger, user-defined procedure names are case sensitive and variable names are not case sensitive.

- File names created in the directory specified by the -M option are spelled using the same case letters as the module name in the source program.

d

The -Ad option raises constants, variables, and functions (intrinsic, statement, external, module and internal functions) of default real and default complex types to double precision.

The -CcR8R16, -CcR4R8, -CcdRR8, -CcdDR16, -Ccd4d8, and -Cca4a8 options cannot be used with the -Ad option.

i

The -Ai option changes the default for integer type data from four-byte to two-byte. Two-byte integer type data will be used for:

- Integer constants with absolute values from 0 to 32767

- Data typed using the default implicit typing rules and having symbolic names beginning with I, J, K, L, M, or N

- Data without kind selector declared in an INTEGER statement

- Data resulting from intrinsic functions INT (without a four-byte integer argument), NINT, IDNINT, ICHAR, MAX1, MIN1, LEN, and INDEX

The -Cci4I8, -CcdII8, -Ccd4d8, and -Cca4a8 options cannot be used with the -Ai option.

m

The -Am option indicates the manner of saving compiled module information. The -Am option is required for all program units that contain modules.

p

The -Ap option specifies a default of private accessibility. It is same as explicitly specifies PRIVATE statement.

q

The -Aq option raises constants, variables, and functions (intrinsic, statement, external, module and internal functions) of double-precision real and double-precision complex types to quadruple precision.

The `-CcR8R16`, `-CcR4R8`, `-CcdRR8`, `-CcdDR16`, `-Ccd4d8`, and `-Cca4a8` options cannot be specified with this option.

w

The `-Aw` option causes the `IACHAR`, `ACHAR`, `IBITS`, and `ISHFTC` to be intrinsic procedures even if the `-X6`, `-Xf6`, `-X7`, or `-Xf7` option is in effect.

y

The `-Ay` option raises the kind (precision) of a signed real literal constant on the right side of an assignment statement to the kind of the variable on the left side. For example,

```
real(8) a
a=1.23456789012345
```

The above program is equivalent to the following program if the `-Ay` option is specified.

```
real(8) a
a=1.23456789012345_8
```

This option does not apply to initialization or to parameter definitions. It does not apply to components when assigning a value of derived type. It does not apply to a constant in an expression on the right hand side if the expressions consists of anything more than just the signed constant.

z

The `-Az` option appends the null character (`\0`) to the end of each character argument passed to an external procedure. This allows a character string to be passed correctly to a C function, such as `strlen`. The length of the character string argument does not include the null character. For example, the result of following program is `1234 4`, even if the `-Az` option is specified.

```
call sub('1234')
end
subroutine sub(arg)
character(len=*) arg
print *,arg , len(arg)
end
```

{0|1|2|n}

The `-A0`, `-A1`, `-A2`, and `-An` options change the information level of the trace back map. The trace back map is generated for an object program and is output to the standard error output file if there is an error during Fortran program execution. The information level of the trace back map controls what information is kept in the trace back map. If the `-A` option is omitted, the default is `-A2`.

0

The `-A0` option does not display internal statement numbers in the trace back map.

1

The `-A1` option displays the internal statement numbers associated with a call to an external procedure in which an error was detected. These numbers are displayed in the trace back map.

2

The -A2 option displays the internal statement numbers of statements in which an error was detected, in addition to the internal statement numbers displayed by the -A1 option.

n

The -An option produces the same result as the -A0 option.

-C *typ_arg typ_arg* : { *l* | autodblpad | autodblpad4 | autodblpad8 | pp | cdll8 | cl4i8 | cdll8 | cl4l8 | cdrr8 | cr4r8 | cd4d8 | ca4a8 | cdDR16 | cR8R16 }

The -C option controls numerical accuracy and data alignment, the C preprocessor, and the evaluation of data types. To specify more than one argument for the -C option, separate the arguments with a comma (,).

***l* *l* : 0 <= *l* <= 15**

The *l* arguments of the -C option determine the effect of rounding errors on the results of floating-point data operations. The *l* must be a number between 0 to 15. When the -C*l* option is specified, the rightmost *l* bits of the mantissa are set to zero each time a value is assigned to a default real, double-precision real, quadruple-precision real, default complex, double-precision complex, or quadruple-precision complex type variable by an assignment statement. (For complex types, these bits are set to zero in both the real and imaginary parts.)

autodblpad

The autodblpad option corresponds to Fujitsu's host system AUTODBL(DBLPAD). In general use of this option is not recommended, but it may be helpful in porting codes that run on mainframe systems. The -CcdII8, -Ccl4I8, -CcdLL8, -Ccl4L8, -CcdRR8, -Ccr4R8, -Ccd4d8, -Cca4a8, -CcdDR16, and -Ccr8R16 options cannot be specified with this option. See -Ad, -Aq, -CcdRR8, -Ccr4R8, -Ccd4d8, -Cca4a8, -CcdDR16, and -Ccr8R16 options for automatic promotion.

autodblpad4

The autodblpad4 option corresponds to Fujitsu's host system AUTODBL(DBLPAD4). In general use of this option is not recommended, but it may be helpful in porting codes that run on mainframe systems. The -CcdII8, -Ccl4I8, -CcdLL8, -Ccl4L8, -CcdRR8, -Ccr4R8, -Ccd4d8, -Cca4a8, -CcdDR16, and -Ccr8R16 options cannot be specified with this option. See -Ad, -Aq, -CcdRR8, -Ccr4R8, -Ccd4d8, -Cca4a8, -CcdDR16, and -Ccr8R16 options for automatic promotion.

autodblpad8

The autodblpad8 option corresponds to Fujitsu's host system AUTODBL(DBLPAD8). In general use of this option is not recommended, but it may be helpful in porting codes that run on mainframe systems. The -CcdII8, -Ccl4I8, -CcdLL8, -Ccl4L8, -CcdRR8, -Ccr4R8, -Ccd4d8, -Cca4a8, -CcdDR16, and -Ccr8R16 options cannot be specified with this option. See -Ad, -Aq, -CcdRR8, -Ccr4R8, -Ccd4d8, -Cca4a8, -CcdDR16, and -Ccr8R16 options for automatic promotion.

pp

The -Cpp option calls the C language preprocessor. (It uses the "#line" form, the same as used in of C language programs.)

cdll8

The -CcdII8 option interprets the default integer variables, constants, and functions as eight-byte integers.

If the `-CcdI18` option is specified, `INT`, `IFIX`, `IDINT`, `IQINT`, `NINT`, `IDNINT`, and `IQNINT` specific intrinsic functions must not be used as an actual argument.

The `-Cautodblpad`, `-Cautodblpad4`, `-Cautodblpad8`, and `-Ai` options cannot be used with this option.

For example,

```
integer ::a
integer(4) ::b
a=2
b=2_4
```

is evaluated as follows if the `-CcdI18` option is specified.

```
integer(8) ::a
integer(4) ::b
a=2_8
b=2_4
```

cl418

The `-Ccl418` option interprets any four-byte integer type (whether default four-byte or explicitly declared four-byte) as eight-byte integer type. It applies to variables, constants, and functions.

If the `-Ccl418` option is specified, `INT`, `IFIX`, `IDINT`, `IQINT`, `NINT`, `IDNINT`, and `IQNINT` specific intrinsic functions must not be used as an actual argument.

The `-Cautodblpad`, `-Cautodblpad4`, `-Cautodblpad8`, and `-Ai` options cannot be used with this option.

For example,

```
integer ::a
integer(4) ::b
a=2
b=2_4
```

is evaluated as follows if the `-Ccl418` option is specified.

```
integer(8) ::a
integer(8) ::b
a=2_8
b=2_8
```

cdLL8

The `-CcdLL8` option interprets default logical variables, constants, and functions as eight-byte logicals.

If the `-CcdLL8` option is specified, the specific intrinsic function `BTEST` must not be used as an actual argument.

The `-Cautodblpad`, `-Cautodblpad4`, and `-Cautodblpad8` options cannot be used with this option.

For example,

```
logical ::a
logical(4) ::b
a=.true.
b=.true._4
```

is evaluated as follows if the -CcdLL8 option is specified.

```
logical(8) ::a
logical(4) ::b
a=.true._8
b=.true._4
```

cL4L8

The -CcL4L8 option interprets any four-byte logical type (whether default four-byte or explicitly declared four-byte) as an eight-byte logical type. It applies to variables, constants, and functions. If the -CcL4L8 option is specified, the specific intrinsic function BTEST must not be used as an actual argument.

The -Cautodblpad, -Cautodblpad4, and -Cautodblpad8 option cannot be used with this option.

For example,

```
logical ::a
logical(4) ::b
a=.true.
b=.true._4
```

is evaluated as follows if the -CcL4L8 option is specified.

```
logical(8) ::a
logical(8) ::b
a=.true._8
b=.true._8
```

cdRR8

The -CcdRR8 option interprets the default real type as double-precision real type and interprets the default complex type as double-precision complex type. It applies to variables, constants, and functions.

If the -CcdRR8 option is specified, the specific intrinsic functions REAL, FLOAT, SNGL, and SNGLQ must not be used as an actual argument.

The -Cautodblpad, -Cautodblpad4, -Cautodblpad8, -Ad, and -Aq option cannot be used with this option.

For example,

```
real ::a
real(4) ::b
complex ::c
complex(4) ::d
a=2.0
```

```
b=2.0_4
c=(2.0,2.0)
d=(2.0_4,2.0_4)
```

is evaluated as follows if the `-CcdRR8` option is specified.

```
real(8) ::a
real(4) ::b
complex(8) ::c
complex(4) ::d
a=2.0_8
b=2.0_4
c=(2.0_8,2.0_8)
d=(2.0_4,2.0_4)
```

cR4R8

The `-Ccr4R8` option is the same as the `-Ad` option. It interprets any default real or complex type (whether default four-byte or explicitly declared four byte) as double-precision real or complex type. It applies to variables, constants, and functions.

If the `-Ccr4R8` option is specified, the specific intrinsic functions `REAL`, `FLOAT`, `SNGL`, and `SNGLQ` must not be used as an actual argument.

The `-Cautodblpad`, `-Cautodblpad4`, `-Cautodblpad8`, `-Ad`, and `-Aq` option cannot be used with this option.

For example,

```
real ::a
real(4) ::b
complex ::c
complex(4) ::d
a=2.0
b=2.0_4
c=(2.0,2.0)
d=(2.0_4,2.0_4)
```

is evaluated as follows if the `-Ccr4R8` option is specified.

```
real(8) ::a
real(8) ::b
complex(8) ::c
complex(8) ::d
a=2.0_8
b=2.0_8
C=(2.0_8,2.0_8)
d=(2.0_8,2.0_8)
```

cd4d8

The `-Ccd4d8` option is equivalent to specifying all of the `-CcdII8`, `-CcdLL8`, and `-CcdRR8` options.

The `-Cautodblpad`, `-Cautodblpad4`, `-Cautodblpad8`, `-Ai`, `-Ad`, and `-Aq` option cannot be used with this option.

For example,

```
integer ::a
integer(4) ::b
logical ::c
logical(4) ::d
real ::e
real(4) ::f
complex ::g
complex(4) ::h
a=2
b=2_4
c=.true.
d=.true._4
e=2.0
f=2.0_4
g=(2.0,2.0)
h=(2.0_4,2.0_4)
```

is evaluated as follows if the `-Ccd4d8` option is specified.

```
integer(8) ::a
integer(4) ::b
logical(8) ::c
logical(4) ::d
real(8) ::e
real(4) ::f
complex(8) ::g
complex(4) ::h
a=2_8
b=2_4
c=.true._8
d=.true._4
e=2.0_8
f=2.0_4
g=(2.0_8,2.0_8)
h=(2.0_4,2.0_4)
```

ca4a8

The `-Cca4a8` option is equivalent to specifying all of the `-CcI4I8`, `-CcL4L8`, and `-CcR4R8` options.

The `-Cautodblpad`, `-Cautodblpad4`, `-Cautodblpad8`, `-Ai`, `-Ad`, and `-Aq` option cannot be specified with this option.

For example,

```
integer ::a
integer(4) ::b
```

```

logical ::c
logical(4) ::d
real ::e
real(4) ::f
complex ::g
complex(4) ::h
a=2
b=2_4
c=.true.
d=.true._4
e=2.0
f=2.0_4
g=(2.0,2.0)
h=(2.0_4,2.0_4)

```

is evaluated as follows if the `-Cca4a8` option is specified.

```

integer(8) ::a
integer(8) ::b
logical(8) ::c
logical(8) ::d
real(8) ::e
real(8) ::f
complex(8) ::g
complex(8) ::h
a=2_8
b=2_8
c=.true._8
d=.true._8
e=2.0_8
f=2.0_8
g=(2.0_8,2.0_8)
h=(2.0_8,2.0_8)

```

cdDR16

The `-CcdDR16` option interprets the double-precision real type as quadruple-precision real type. It applies to variables, constants, and functions.

If the `-CcdDR16` option is specified, the specific intrinsic functions `DFLOAT`, `DBLE`, `DBLEQ`, `DREAL`, and `DPROD` must not be used as an actual argument.

The `-Cautodblpad`, `-Cautodblpad4`, `-Cautodblpad8`, `-Ad`, and `-Aq` option cannot be used with this option.

For example,

```

double precision ::a
real(8) ::b
a=2.0d0
b=2.0_8

```

is evaluated as follows if the `-CcdDR16` option is specified.

```
real(16) ::a
real(8)  ::b
a=2.0_16
b=2.0_8
```

cR8R16

The `-CcR8R16` option is the same as the `-Aq` option. It interprets any double-precision real type as quadruple-precision type and interprets any double-precision complex type as quadruple-precision complex type. It applies to variables, constants, and functions.

If the `-CcR8R16` option is specified, the specific intrinsic functions `DFLOAT`, `DBLE`, `DBLEQ`, `DREAL`, and `DPPROD` must not be used as an actual argument.

The `-Cautodblpad`, `-Cautodblpad4`, `-Cautodblpad8`, `-Ad`, and `-Aq` option cannot be used with this option.

For example,

```
double precision ::a
real(8)         ::b
complex(8)      ::c
a=2.0d0
b=2.0_8
c=(2.0_8, 2.0_8)
```

is evaluated as follows if the `-CcR8R16` option is specified.

```
real(16) ::a
real(16) ::b
complex(16) ::c
a=2.0_16
b=2.0_16
c=(2.0_16, 2.0_16)
```

-D name[=tokens]

The `-D` option defines a *name* for use by the C preprocessor (`cpp`) as if by a `#define` directive. If the *tokens* argument is unspecified, this is the same as if a `-D name=1` option appeared.

Even if `-D` is not specified, the following `#define` directives are provided by default; they set the name to have the default value 1:

```
unix
i386
__unix
__i386
__linux
__ELF__
__unix__
__i386__
__linux__
```

-E *msg_arg msg_arg* : { c | g | i | s | t | u }

The -E option tells the Fortran compiler to output diagnostic messages about operations for which round off may affect the results. This option also tells the Fortran compiler to output short diagnostic messages and optimization messages. The arguments of the -E option are c, g, i, s, t, and u. These arguments may be combined, e.g., -Ecis.

c

The -Ec option outputs level *i* diagnostic messages for relational operations that check whether two operands of any real or complex precision kind are exactly equal (`==`, `/=`, `.EQ.`, and `.NE.`). During compilation, this option also outputs level *i* diagnostic messages for arithmetic IF statements with real expressions.

g

The -Eg option checks characteristics of a procedure between procedure definition and reference, between procedure definition and interface body, and size of common block during compilation. This function increase the time needed for compilation.

If the -Eg option is specified, the -Haesux and -O0 options are in effect.

i

The -Ei option outputs level *i* diagnostic messages related to optimization for inline expansion of user-defined external procedures at the location referenced. These messages are output during compilation.

For information about optimization by inline expansion of user-defined external procedures, see the compiler option -x.

s

The -Es option changes the long diagnostic messages output during compilation to short diagnostic messages.

t

The -Et option outputs messages related to parallelization if the -Kparallel option is in effect.

u

The -Eu option outputs level *i* diagnostic messages related to optimization by loop unrolling. These messages are output during compilation. For information about optimization of loop unrolling, see the compiler options -O and -K.

-Fixed

The -Fixed option causes the Fortran compiler to interpret a Fortran source program as written in fixed source form.

-Free

The -Free option causes the Fortran compiler to interpret a Fortran source program as written in free source form.

-H *dbg_arg dbg_arg* : { a | e | s | u | x }

The -H option specifies the level of run-time error checking. During compilation some errors can be detected in Fortran object programs and error messages are generated. Program execution automatically

detects some errors, and the arguments a, e, s, u, and x can be specified to broaden the range of errors checked. Arguments a, e, s, u, and x can be combined. If the -H option is specified, the default optimization is -O0. The -O option may be specified after the -H option to make the -O option effective.

The -Eg option is supported to check the program error, too. It checks characteristics of a procedure between procedure definition and reference, between procedure definition and interface body, and size of common block during compilation.

a

The -Ha option compares the numbers, types, attributes, and sizes of actual and dummy arguments.

The types of function results are also checked during execution. If a mismatch occurs during execution, the corresponding diagnostic message is output.

e

The -He option checks shape conformance when an array assignment statement is executed. If a mismatch of shape conformance occurs during execution, the corresponding diagnostic message is output.

Example:

```
integer, allocatable, dimension(:, :) :: a, b
allocate (a(2,3), b(3,2))
a=b+1 ! Not shape conformance
end
```

s

When an array section, array element, or substring is referenced during execution, the -Hs option checks the value of each subscript or substring expression to see if it is within the declared range. The same check is done during compilation when possible. If the value is not within the declared range, a diagnostic message is output.

u

When a variable outside a common block is referenced during execution, the -Hu option checks to see if the variable is defined. If the referenced variable is undefined, a diagnostic message is output.

x

The -Hx option checks to see if the variable declared in a module or in a common block is defined when the variable is referenced during execution, to see if the pointer is associated when a pointer is referenced during execution, and to see if the incrementation parameter of the DO construct, stride of the FORALL, incrementation parameter of array constructor implied do control and stride of subscript-triplet is not 0.

If the -Hx option is specified, the -Hu option is in effect.

Be careful to the following when the -Hx option using:

All program units that has a definition or initialization for common block object shall be compiled with -Hx option.

All program units that uses the module shall be compiled with -Hx option, if the module is compiled with -Hx option.

-I *directory*

The -I option specifies directory names to be searched for files specified in an INCLUDE line or for module information (.mod) files. It is also necessary to specify the -Am option to search for .mod files.

More than one -I option may be specified. If multiple -I options are specified, the path names are searched in the order specified.

INCLUDE file retrieval is performed in the following order:

- 1) The directory of the files containing the INCLUDE lines
- 2) The directory in which frt(1) is being executed
- 3) Directories specified as the argument of an -I option

The module information (.mod) file is searched in the following order:

- 1) Directories specified as the argument of a -M option
- 2) The directory in which frt(1) is being executed
- 3) Directories specified as the argument of an -I option

-K *opt_arg opt_arg* : { auto | eval | fap | fast | fastlib | instance=*N* | loop | maxcpunum=*N* | nofap | noomitfp | NOOMP | nospinwait | nothreadheap | nothreadprivate | nothreadsafe | nounroll | ocl | omitfp | OMP | parallel | PIC | preex | reduction | spinwait | threadheap[=*size*] | threadprivate | threadsafe | threadstacksize=*N* | *cpu* }

The -K option specifies optimizations, such as advanced evaluation of invariant expressions or changing the method of operation evaluation. To specify more than one argument for the -K option, separate the arguments with commas.

auto

The -Kauto option causes local variables, except for ones that have the SAVE attribute or have been initialized, to become undefined when execution of the procedure terminates.

eval

The -Keval option creates an object program by applying optimizations that change the method of operation evaluation. To avoid unwanted side effects in execution results, use this optimization carefully. To determine whether this type of optimization was applied, check any diagnostic messages output during compilation.

fap

When the -Kfap option is specified, the compiler does not keep the values of four-byte real, double-precision real, quadruple-precision real, eight-byte complex, double-precision complex and quadruple-precision complex type user variables in the hardware registers. All four-byte real, double-precision real, quadruple-precision real, eight-byte complex, double-precision complex and quadruple-precision complex type user variables are loaded and stored to memory exactly as in the source program. Therefore, this option results in fewer arithmetic errors than -Knofap (default) operation.

Note that:

- The execution speed is slower than with -Knofap.
- There may be differences between the result of -Kfap and the result of -O0.
- You cannot specify -O0 and -Kfap at the same time.

fast

The fast argument of the option -K creates object code that runs fast on the compiling machine. It is equivalent to -O -Kloop -Komitfp , and -Kcpu selected for the compiling environment.

fastlib

The -Kfastlib option causes input/output to be fully buffered in standard files. This option is effective when it is specified for a main program.

Restriction on -Kfastlib:

The output sequence for the standard output or standard error output might not be the same, due to buffering.

instance=N

N specifies the number of instances (threads) to be created. *N* must be greater than 1 and no larger than the number of CPUs active at run time. If this option is specified, the compiler does not have to produce code identifying how many CPUs are available at run time, increasing run time performance.

The -Kparallel option also must be specified.

loop

The -Kloop option modifies the structure of nested loops.

maxcpunum=N 1 < N < 2147483647

This option is effective when it is specified with -KOMP option and for a main program. The *N* indicates the number of available CPUs. The *N* must be less than system defined CPUs' number. If the *N* is specified more value than the system-defined CPUs' number, this option is to be ineffective.

nofap

When the -Knofap option is specified, the compiler tries to keep the values of four-byte real, double-precision real, quadruple-precision real, eight-byte complex, double-precision complex and quadruple-precision complex type user variables in the hardware registers and optimizes based on that. This is the default.

noomitfp

The -Knoomitfp is default option, if -Kfast is not specified. The frame pointer in a register is kept.

NOOMP

Compiles the source programs that follow the OpenMP directives as a comment. When the -KOMP and -KNOOMP options are not specified, the -KNOOMP is selected.

nospinwait

This option is effective when it is specified with -KOMP option and for a main program. When the outside of the parallel region is executed, the thread is suspended. The consuming time of CPU will be reduced, however, its overhead is larger. This option is suitable for attaching the CPU time.

nothreadheap

This option is effective when it is specified with -Kthreadsafe option. If the -Knothreadheap is effect, local array is allocated in stack. The -Knothreadheap is default. When the -Kthreadheap and -Knothreadheap options are not specified, the -Knothreadheap is selected.

nothreadprivate

This option is effective when it is specified with -KOMP option. Except the named common blocks that are specified as threadprivate, they become the normal common blocks. When the -Kthreadprivate and -Knothreadprivate options are not specified, the -Knothreadprivate is selected.

nothreadsafe

Generates the object program that is not multi thread safe. When the -Kthreadsafe and -Knothreadsafe options are not specified, the -Knothreadsafe is selected.

nounroll

The -Knounroll suppresses loop unrolling.

ocl

The -Kocl option causes OCLs (Optimizing Control Lines) to be recognized.

omitfp

The frame pointer in a register does not keep. If the -Komitfp is specified, the information of traceback map functions are chipped.

OMP

Makes effective the directive that follows OpenMP Fortran Specifications and compiles the source programs. The -KOMP options sets the -Kthreadsafe, -Kauto, -AR, and -D_OPENMP options. The -Knothreadsafe, -Knoauto, -Ar options may be specified after the -KOMP option to make these options effective. It is ignored if the -Cautodblpad, -Cautodblpad4, is -Cautodblpad8 option is specified.

Even when only an object file name appears to the file name list, if the object program compiled by this is included, this option must be specified.

parallel

The -Kparallel option causes parallel optimization to be performed and sets the -O option. It is ignored if the -g or -H[aesux] option is specified.

PIC

The -KPIC option creates position-independent code (PIC).

preex

The -Kpreex applies optimizations that advance evaluation of invariant expressions. To avoid unwanted side effects in the execution result, use this type of optimization carefully. To determine whether this type of optimization was applied, check any diagnostic messages output during compilation.

reduction

The -Kreduction option causes reduction parallelization optimization to be performed. The -Kparallel option also must be specified.

spinwait

This option is effective when it is specified with **-KOMP** option and for a main program. During executing the part outside of parallel regions, this option keeps the thread to be active in the runtime library. Though this consumes CPU time, the overhead will be reduced. This options is suitable for taking care of the elapsed time of CPU. When the **-Kspinwait** and **-Knospinwait** options are not specified, the **-Kspinwait** is selected.

threadheap[=size] 1 <= size <= 2147483647

This option is effective when it is specified with **-Kthreadsafe** option. If the **-Kthreadheap** is effect, local array that is larger than *size* byte is allocated in heap. If the **=size** is omitted, 4096 is selected for *size*.

threadprivate

This option is effective when it is specified with **-KOMP** option. All the named common blocks are to be threadprivate.

threadsafe

Generates the object program that is multi thread safe.

threadstacksize=N 16 <= N <= 2048

This option is effective when it is specified with **-KOMP** or **-Kparallel** option, and for a main program. The amount of the stack area of each thread *N* is specified in Kbytes.

The maximum stack size of the Linux thread is 2048 Kbytes. It isn't effective when the value greater than 2048 is specified for *N*.

cpu

For *cpu* specify one of the PENTIUM or PENTIUM_PRO. If the **-Kcpu** option is not specified, **-KPENTIUM** is chosen.

PENTIUM

The **-KPENTIUM** option creates an object program that uses PENTIUM instructions.

PENTIUM_PRO

The **-KPENTIUM_PRO** option creates an object program that uses PENTIUM PRO instructions.

-L directory

The **-L** option add *directory* to the list of directories in which the linker searches for libraries. This option and its argument are passed to the linker.

-M directory

The **-M** option specifies an alternate *directory* for module information files (.mod files). If **-M** is specified, then **-Am** must also be specified. .mod files are created in *directory* during compilation.

-N src_arg *src_arg* : { { allextput | noallextput } | { autoobjstack | noautoobjstack } | { compdisp | nocompdisp } | { copyarg | nocopyarg } | { f90move | nof90move } | { f95 | nof95 } | { freealloc | nofreealloc } | { mallocfree | nomallocfree } | maxserious=*maxnum* | { obsfun | noobsfun } | { recursive | norecursive } | { save | nosave } | { Rtrap | Rnotrap } }

allextput

The -Nallextput option creates an external name which appears only on EXTERNAL statement. The -Nallextput is default.

noallextput

The -Nnoallextput option doesn't create an external name which appears only on EXTERNAL statement.

autoobjstack

The -Nautoobjstack option allocates automatic data object on the stack.

noautoobjstack

The -Nnoautoobjstack option allocates automatic data object on the heap. The -Nnoautoobjstack is default.

compdisp

Displays filename and program name in compiling.

nocompdisp

Don't display filename and program name in compiling. -Nnocompdisp is default.

copyarg

The -Ncopyarg option specify to copy scalar constant argument to generated variable argument. The program is not Fortran standard compliant.

Example: Output 1 when -Ncopyarg is specified

```
call sub(1)
print *,1
end
subroutine sub(i)
i=2
end
```

nocopyarg

The -Nnocopyarg option specify to pass the scalar constant argument. It is default.

f90move

In FORTRAN77 or FORTRAN66 (-X6, -Xf6, -X7, or -Xf7 is specified), overlapping data (in a storage area) can be specified on both sides of a character assignment statement if -Nf90move is in effect. If the -X9 option is specified, the -Nf90move option is in effect by default.

For example,

```
CHARACTER(LEN=5) :: C
```

```

C='12345'
C(1:4) = C(2:5)
PRINT *,C ! output is 23455

```

Note. In FORTRAN77 or FORTRAN66, if the `-Nf90move` is not specified, the action is illegal and the result is undefined.

nof90move

In FORTRAN77 or FORTRAN66 (`-X6`, `-Xf6`, `-X7`, or `-Xf7` is specified), `-Nnof90move` is the default. `-X9` option cancels the `-Nnof90move` option.

f95

The `CPU_TIME` and `NULL` procedure names are interpreted as intrinsic procedures. The `CPU_TIME` and `NULL` procedure names are new in Fortran 95. If the `-X9` option is specified, `-Nf95` is default. An `-X7`, `-Xf7`, `-X6`, or `-Xf6` option cancels the `-Nf95` option.

nof95

The `CPU_TIME` and `NULL` procedure names are interpreted as user-defined procedures. If `-X7`, `-Xf7`, `-X6`, or `-Xf6` is specified, the `-Nnof95` option is not canceled.

freealloc

If an unsaved allocatable array has a status of currently allocated when a procedure is exited, the array is deallocated. This is the behavior specified by the Fortran 95 standard and `-Nfreealloc` is the default.

nofreealloc

If an unsaved allocatable array has a status of currently allocated when a procedure is exited, the array is not deallocated. This is not the behavior specified by the Fortran 95 standard.

mallocfree

Evaluates `MALLOC` and `FREE` as intrinsic procedures.

nomallocfree

Evaluates `MALLOC` and `FREE` as service routines. `-Nnomallocfree` is default.

maxserious=*maxnum*

Stop the compilation if s-level(serious) error detected more than *maxnum*. *maxnum* must be greater than or equal to 1. The default is unlimited.

obsfun

The following names are interpreted as intrinsic functions.

```

AIMAX0, AJMAX0, I2MAX0, IMAX0, JMAX0, IMAX1, JMAX1,
AIMIN0, AJMIN0, I2MIN0, IMIN0, JMIN0, IMIN1, JMIN1,
FLOATI, FLOATJ, DFLOTI, DFLOTJ,
IIABS, JIABS, I2ABS, IIDIM, JIDIM, I2DIM,
IIFIX, JIFIX, IFIX, INT1, INT2, INT4, IINT, JINT,
ININT, JNINT, IIDNNT, I2NINT, JIDNNT, IIDINT, JIDINT,
IMOD, JMOD, I2MOD, IISIGN, JISIGN, I2SIGN,
BITEST, BJTEST, IIBCLR, JIBCLR, IIBITS, JIBITS, IIBSET, JIBSET,
IBCHNG, ISHA, ISHC, ISHL,

```

IIAND, JIAND, IIEOR, JIEOR, IIOR, JIOR, INOT, JNOT,
IISHFT, JISHFT, IISHFTC, JISHFTC,
IZEXT, JZEXT, IZEXT2, JZEXT2, JZEXT4,
VAL

noobsfun

-Nnoobsfun is the default. See the -Nobsfun option.

recursive

The RECURSIVE keyword is added in each SUBROUTINE and FUNCTION statement. For example,

```
SUBROUTINE SUB
```

is evaluated as follows if the -Nrecursive option is specified.

```
RECURSIVE SUBROUTINE SUB
```

norecursive

The RECURSIVE keyword is not added in SUBROUTINE and FUNCTION statements. -norecursive is the default.

save

The SAVE statement without saved entity list is added in each program unit except main program.

nosave

The SAVE statement without saved entity list is not added in program unit. -Nnosave is the default.

Rtrap

If the main program is compiled with -Rtrap option, at the execution, the intrinsic operation error messages are output and the floating point exceptions are taken. See the -NRnotrap option.

Rnotrap

If the main program is compiled with -Rnotrap option, at the execution, the intrinsic operation error messages are not output and the floating point exceptions are not taken. -NRnotrap is the default.

-O [*opt_lvl*] *opt_lvl* : 0

The -O option specifies the optimization level used by the compiler.

The system provides optimization(-O) and no optimization(-O0). If the -O option is not specified, it defaults to -O.

If the -O option is specified and the -g option is also specified, the combined specification is treated as -O0 and -g.

In addition to the -O option, the Fortran compiler supports the -K and -x options for optimization.

0

The -O0 option creates an object program without applying optimization. A program compiled with the zero optimization level requires the least compile time and memory. Specify this argument to debug compilation errors in Fortran source programs.

-P

The -P option causes temporary files produced by the C preprocessor (cpp) to be saved. The -Cpp option also must be specified unless a suffix of the source file is .F, .FOR, .F90, or .F95.

The temporary files generated are as follows:

file.F	file.cpp.f
file.f	file.cpp.f
file.FOR	file.cpp.for
file.for	file.cpp.for
file.f90	file.cpp.f90
file.F90	file.cpp.f90
file.f95	file.cpp.f95
file.F95	file.cpp.f95

-Q [*lst_arg*] *lst_arg* : { a | d | i | *ofile* | x }

The -Q option outputs compilation information to a file with suffix .lst. If more than one Fortran source program files is specified, it is output to the first-file-name.lst.

The -Q option can be specified with arguments a, d, i, *ofile*, and x. The arguments may be specified at the same time, separated by comma as in -Qa,*ofile*,x. These arguments are optional. If no argument is specified, the source program list and diagnostic error messages are produced.

a

If the -Qa option is specified, the attributes of names are output in addition to the compilation information produced by the -Q option.

d

If the -Qd option is specified, the layout of derived types is output in addition to the compilation information produced by the -Q option.

i

If the -Qi option is specified, include files are listed in addition to the compilation information produced by the -Q option.

ofile

The *file* name must be specified immediately after o. The compilation information is output to the *file*.

x

If the -Qx option is specified, the cross reference list of name and label is output in addition to the compilation produced by the -Q option.

-S

The -S option suppresses creation of the corresponding object programs and does not call the link editor. Instead, assembly programs (files with the suffix .s) are created.

-U name

The -U option undefines name, which has the same effect as an #undef preprocessing directive. If the same name is specified for both -D and -U, name is not defined, regardless of the order of the options.

-V

The -V option displays the version information of each tool.

-Wtool,arg1[,arg2, ...]

Passes each argument *arg1* [, *arg2*, ...] as a separate argument to a *tool*. The arguments must be separated by commas (a comma can be part of an argument by using a backslash as an escape character before it. The backslash is removed from the resulting argument).

tool can be one of the following:

- p Preprocessor
- 0 Compiler
- a Assembler
- l Linker

For example, -Wa,-oobjfile passes -o and objfile to the assembler, in that order. Also, -Wl,-lname causes the linking phase to override the default name of the dynamic linker. The order in which arguments are passed to a *tool* may with respect to the other specified command line options.

-X lan_lvl lan_lvl : { 6 | 7 | 9 | f6 | f7 }

The -X option indicates the level of language specification. Either 6, 7, 9, f6, or f7 can be specified as the argument of the -X option.

The interpretation of Fortran source programs is different for each level of language specification. To compile Fortran source programs, indicate the level of language specification used by the Fortran compiler.

6

The 6 argument of the -X option compiles Fortran source programs as FORTRAN66 source.

7

The 7 argument of the -X option compiles Fortran source programs as FORTRAN77 source.

If the suffix of the Fortran source file is .f or .F, -X7 is the default.

9

The 9 argument of the -X option compiles Fortran source programs as Fortran 95 or Fortran 90 source.

If the suffix of Fortran source file is .f90, .F90, .f95, or .F95, -X9 is the default.

f6

The f6 argument of the -X option compiles Fortran source programs in Fujitsu FORTRAN66 host system compatibility mode.

f7

The f7 argument of the -X option compiles Fortran source programs in Fujitsu FORTRAN77 host system compatibility mode.