Intel(R) C++ Compiler Help
==========

```
usage: icpc [options] file1 [file2 ...]

  where options represents zero or more compiler options

        fileN is a C/C++ source (.C .c .cc .cpp .cxx .c++ .i .ii),
              assembly (.s .S), object (.o), static library (.a), or other
              linkable file


  Commonly used options may be placed in the icpc.cfg file.
```

Compiler Option List

```
Optimization
------------
-O1    optimize for maximum speed, but disable some optimizations which
       increase code size for a small speed benefit.
-O2    enable optimizations (DEFAULT)
-O3    enable -O2 plus more aggressive optimizations that may not improve
       performance for all programs
-O0    disable optimizations
-O     same as -O2
-Os    enable speed optimizations, but disable some optimizations which
       increase code size for small speed benefit
-fast  enable -xT -O3 -ipo -no-prec-div -static
       Options set by -fast cannot be overridden, list options separately to
       change behavior
-falias       assume aliasing in program (DEFAULT)
-fno-alias    assume no aliasing in program
-ffnalias     assume aliasing within functions (DEFAULT)
-fno-fnalias  assume no aliasing within functions, but assume aliasing across
              calls
-inline-level=<n> control inline expansion:
    n=0  disables inlining
    n=1  inline functions declared with __inline, and perform C++ inlining
    n=2  inline any function, at the compiler's discretion (same as -ip)
-f[no-]inline inline functions declared with __inline, and perform C++ inlining
-f[no-]inline-functions
              inline any function, at the compiler's discretion (same as -ip)
-finline-limit=<n>
              set maximum number of statements a function can have and still
              be considered for inlining
-f[no-]builtin
              disable inline expansion of intrinsic functions
-fno-builtin-<func>
              disable the <func> intrinsic
-ffunction-sections
              separate functions for the linker (COMDAT)
-f[no-]exceptions
              enable(DEFAULT)/disable exception handling
-fdata-sections
              same as -ffunction-sections
-f[no-]omit-frame-pointer
              enable(DEFAULT)/disable use of EBP as general purpose register.
```

```
              -fno-omit-frame-pointer replaces -fp
-f[no-]keep-static-consts
              enable/disable(DEFAULT) emission of static const variables even
              when not referenced

Advanced Optimization
---------------------
-unroll[n]      set maximum number of times to unroll loops.  Omit n to use
                default heuristics.  Use n=0 to disable loop unroller.
-funroll-loops  unroll loops based on default heuristics
-nolib-inline   disable inline expansion of intrinsic functions
-[no-]alias-args       enable(DEFAULT)/disable C/C++ rule that function
                       arguments may be aliased; when disabling the rule, the
                       user asserts that this is safe
-[no-]alias-const      enable/disable(DEFAULT) a heuristic stating that if two
                       arguments to a function have pointer type, a pointer to
                       const does not alias a pointer to non-const. Also known
                       as the input/output buffer rule, it assumes that input
                       and output buffer arguments do not overlap.
-fargument-alias       arguments may alias each other and may alias global
                       storage. Replaces -alias-args
-fargument-noalias     arguments do not alias each other but may alias global
                       storage. Replaces -no-alias-args
-fargument-noalias-global
                       arguments do not alias each other and do not alias
                       global storage
-[no-]scalar-rep       enable(DEFAULT)/disable scalar replacement (requires
                       -O3)
-[no-]ansi-alias       enable/disable(DEFAULT) use of ANSI aliasing rules in
                       optimizations; user asserts that the program adheres to
                       these rules
-[no-]complex-limited-range
                       enable/disable(DEFAULT) the use of the basic
                       algebraic expansions of some complex arithmetic
                       operations.  This can allow for some performance
                       improvement in programs which use a lot of complex
                       arithmetic at the loss of some exponent range.
-ftls-model=<model>    change thread-local storage model, where <model> can
                       be the following:  global-dynamic, local-dynamic,
                       initial-exec or local-exec

Code Generation
---------------
-mcpu=<cpu>   same as -mtune=<cpu>
-mtune=<cpu>  optimize for a specific cpu
       pentium3   - optimize for Pentium(R) III processors
       pentium4   - optimize for Pentium(R) 4 processor (DEFAULT)
-march=<cpu>  generate code excusively for a given <cpu>
       pentium3   - streaming SIMD extensions
       pentium4   - Pentium(R) 4 New Instructions
-msse  generate code for Intel Pentium III and compatible Intel processors
-msse2 generate code for Intel Pentium 4 and compatible Intel processors
-msse3 generate code for Intel(R) Core(TM) Duo processors, Intel(R) Core(TM)
       Solo processors, Intel Pentium 4 and compatible Intel processors with
       Streaming SIMD Extensions 3 (SSE3) instruction support
-ax<codes> generate code specialized for processors specified by <codes>
           while also generating generic IA-32 code.  <codes> includes
           one or more of the following characters:
```

```
    K  Intel Pentium III and compatible Intel processors
    W  Intel Pentium 4 and compatible Intel processors
    N  Intel Pentium 4 and compatible Intel processors.  Enables new
       optimizations in addition to Intel processor-specific optimizations
    P  Intel(R) Core(TM) Duo processors, Intel(R) Core(TM) Solo processors,
       Intel Pentium 4 and compatible Intel(R) processors with Streaming
       SIMD Extensions 3 (SSE3) instruction support
    T  Intel(R) Core(TM)2 Duo processors, Intel(R) Core(TM)2 Quad
       processors, and Intel(R) Xeon(R) processors with SSSE3
-x<codes>  generate specialized code to run exclusively on processors
           indicated by <codes> as described below
    K  Intel Pentium III and compatible Intel processors
    W  Intel Pentium 4 and compatible Intel processors
    N  Intel Pentium 4 and compatible Intel processors.  Enables new
       optimizations in addition to Intel processor-specific optimizations
    P  Intel(R) Core(TM) Duo processors, Intel(R) Core(TM) Solo processors,
       Intel Pentium 4 and compatible Intel(R) processors with Streaming
       SIMD Extensions 3 (SSE3) instruction support
    T  Intel(R) Core(TM)2 Duo processors, Intel(R) Core(TM)2 Quad
       processors, and Intel(R) Xeon(R) processors with SSSE3
    O  Intel(R) Core(TM)2 Duo processors.  Code is expected to run properly
       on any processor that supports SSE3, SSE2 and SSE instruction sets

Language
--------
-[no-]restrict
           enable/disable the 'restrict' keyword for disambiguating pointers
-export    enable the export template feature
-export-dir <dir>
           add directory to export template search path
-ansi      equivalent to GNU -ansi
-strict-ansi
           strict ANSI conformance dialect
-std=c99   enable C99 support for C programs
-std=c++0x enable preliminary support for some C++0x features
-trigraphs support ISO C trigraphs; also enabled in ANSI and C99 modes
-Kc++      compile all source or unrecognized file types as C++ source files
-fno-rtti  disable RTTI support
-x <type>  all source files found subsequent to -x <type> will be recognized
           as one of the following types:
    c                  - C source file
    c++                - C++ source file
    c-header           - C header file
    cpp-output         - C pre-processed file
    c++-cpp-output     - C++ pre-processed file
    assembler          - assembly file
    assembler-with-cpp - assembly file that needs to be preprocessed
    none               - disable recognition, and revert to file extension
-[no]align analyze and reorder memory layout for variables and arrays
-malign-double
           same as -align
-Zp[n]     specify alignment constraint for structures (n=1,2,4,8,16)
-fshort-enums
           allocate as many bytes as needed for enumerated types
-fsyntax-only
           perform syntax and semantic checking only (no object file produced)
-funsigned-char
           change default char type to unsigned
```

```
-f[no-]unsigned-bitfields
           change default bitfield type to unsigned
-fno-implicit-templates
           never emit code for non-inline templates which are instantiated
           implicitly; only emit code for explicit instantiations
-fno-implicit-inline-templates
           do not emit code for implicit instantiations of inline templates
-ftemplate-depth-<n>
           control the depth in which recursive templates are expanded
-fno-operator-names
           disable support for operator name keywords
-fno-gnu-keywords
           do not recognize 'typeof' as a keyword
-fpermissive
           allow for non-conformant code
-f[no-]non-lvalue-assign
           allow (DEFAULT) or disallow casts and conditional expressions to
           be used as lvalues
-[no-]early-template-check
           enable/disable (DEFAULT) semantic checking of function template
           prototypes (before instantiation). Requires -gcc-version=340 or
           later
-[no-]check-uninit
           check for uninitialized variables

Compatibility
-------------
-gcc-name=<name>
              name and location of gcc if not where expected
-gcc-version=<version>
              specify the <version> of gcc compatibility.  Default value
              matches gcc version installed.  Major/Minor versions listed
              but patch levels (i.e. 345) are permissible
              320 - gcc 3.2.x compatibility
              330 - gcc 3.3.x compatibility
              340 - gcc 3.4.x compatibility
              400 - gcc 4.0.x compatibility
              410 - gcc 4.1.x compatibility
              420 - gcc 4.2.x compatibility
-B<prefix>    find libraries, headers and executables in <prefix>
-[no-]multibyte-chars
              provide support for multi-byte characters
-fabi-version=<val>
              directs the compiler to select a specific ABI implementation
              0 - most recent ABI implementation
              1 - g++ 3.2 compatible ABI implementation
              2 - most conformant ABI implementation

Compiler Diagnostics
--------------------
-w                 disable all warnings
-w<n>              control diagnostics:
   n=0               display errors (same as -w)
   n=1               display warnings and errors (DEFAULT)
   n=2               display remarks, warnings, and errors
-wn<n>             print a maximum of n errors
-wd<L1>[,<L2>,...] disable diagnostics L1 through LN
-we<L1>[,<L2>,...] change severity of diagnostics L1 through LN to error
```

```
-ww<L1>[,<L2>,...] change severity of diagnostics L1 through LN to warning
-wr<L1>[,<L2>,...] change severity of diagnostics L1 through LN to remark
-wo<L1>[,<L2>,...] issue diagnostics L1 through LN only once
-Werror            force warnings to be reported as errors
-Wall              enable all warnings
-Wbrief            print brief one-line diagnostics
-Wcheck            enable more strict diagnostics
-W[no-]format      enable argument checking for calls to printf, scanf, etc
-W[no-]missing-declarations
                   warn for global functions and variables without prior
                   declaration
-W[no-]missing-prototypes
                   warn for missing prototypes
-W[no-]strict-prototypes
                   warn for functions declared or defined without specified
                   argument types
-W[no-]pointer-arith
                   warn for questionable pointer arithmetic
-W[no-]uninitialized
                   warn if a variable is used before being initialized
-Winline           enable inline diagnostics
-W[no-]deprecated  print warnings related to deprecated features
-W[no-]abi         warn if generated code is not C++ ABI compliant
-Wcontext-limit=<n>
                   set maximum number of template instantiation contexts shown
                   in diagnostic
-W[no-]unused-function
                   warn if declared function is not used
-W[no-]unknown-pragmas
                   warn if an unknown #pragma directive is used (DEFAULT)
-W[no-]main        warn if return type of main is not expected
-W[no-]comment[s]  warn when /* appears in the middle of a /* */ comment
-W[no-]return-type warn when a function uses the default int return type and
                   warn when a return statement is used in a void function
-W[no-]extra-tokens
                   warn about extra tokens after prepreprocessor directives
-W[no-]pragma-once warn about the use of #pragma once
-W[no-]shadow      warn when a variable declaration hides a previous
                   declaration
-W[no-]trigraphs   warn about the recognition and conversion of trigraphs
-W[no-]multichar   warn if a multicharacter constant ('ABC') is used
-Wp64              print diagnostics for 64-bit porting
-W[no-]shorten-64-to-32
                   warn for values implicitly converted from a 64-bit to
                   a 32-bit type. Similar to -Wp64
-Weffc++           enable effective C++ diagnostic warnings
-[no]traceback     specify whether the compiler generates data to allow for
                   source file traceback information at runtime (only to be
                   used when linking with Fortran programs)

-diag-enable <v1>[,<v2>,...]
           enable a specific diagnostic or a specified group of diagnostics
-diag-disable <v1>[,<v2>,...]
           disable a specific diagnostic or a specified group of diag-nostics
-diag-error <v1>[,<v2>,...]
           make the specified diagnostic or group of diagnostics warnings
           when emitted
-diag-warning <v1>[,<v2>,...]
```

```
           make the specified diagnostic or group of diagnostics remarks
           when emitted
-diag-remark <v1>[,<v2>,...]
           make the specified diagnostic or group of diagnostics errors
           when emitted
-diag-dump
           display the currently enabled diagnostic messages to stdout or to
           a specified diagnostic output file.  No compilation is performed
-diag-file[=<file>]
           <file> where diagnostics are emitted to.  Not specifying this
           causes messages to be output to stderr
-[no-]diag-id-numbers
           enable(DEFAULT)/disable the diagnostic specifiers to be output
           in numeric form

Inlining
--------
-inline-min-size=<n>
           set size limit for inlining small routines
-no-inline-min-size
           no size limit for inlining small routines
-inline-max-size=<n>
           set size limit for inlining large routines
-no-inline-max-size
           no size limit for inlining large routines
-inline-max-total-size=<n>
           maximum increase in size for inline function expansion
-no-inline-max-total-size
           no size limit for inline function expansion
-inline-max-per-routine=<n>
           maximum number of inline instances in any function
-no-inline-max-per-routine
           no maximum number of inline instances in any function
-inline-max-per-compile=<n>
           maximum number of inline instances in the current compilation
-no-inline-max-per-compile
           no maximum number of inline instances in the current compilation
-inline-factor=<n>
           set inlining upper limits by n percentage
-no-inline-factor
           do not set inlining upper limits
-inline-forceinline
           treat inline routines as forceinline

Interprocedural Optimizations (IPO)
-----------------------------------
-ip     enable single-file IP optimizations (within files)
-ipo[n] enable multi-file IP optimizations (between files)
-ipo-c  generate a multi-file object file (ipo_out.o)
-ipo-S  generate a multi-file assembly file (ipo_out.s)
-ip-no-inlining    disable full and partial inlining (requires -ip or -ipo)
-ip-no-pinlining   disable partial inlining (requires -ip or -ipo)
-ipo-separate      create one object file for every source file
                   (overrides -ipo[n])
-ipo-jobs<n>       specify the number of jobs to be executed simultaneously
                   during the IPO link phase

Profile Guided Optimization (PGO)
```

```
---------------------------------
-prof-dir <d>   specify directory for profiling output files (*.dyn and *.dpi)
-prof-file <f>  specify file name for profiling summary file
-prof-gen[x]    instrument program for profiling; with the x qualifier, extra
                information is gathered
-prof-use       enable use of profiling information during optimization
-prof-gen-sampling
                prepare code for use with profrun sample gathering tool
-[no-]func-groups
                enable(DEFAULT with PGO)/disable function grouping
-p              compile and link for function profiling with UNIX gprof tool
-f[no-]instrument-functions
                determine whether function entry and exit points are
                instrumented

Optimization Reports
--------------------
-vec-report[n]  control amount of vectorizer diagnostic information:
            n=0 no diagnostic information
            n=1 indicate vectorized loops (DEFAULT)
            n=2 indicate vectorized/non-vectorized loops
            n=3 indicate vectorized/non-vectorized loops and prohibiting
                data dependence information
            n=4 indicate non-vectorized loops
            n=5 indicate non-vectorized loops and prohibiting data
                dependence information

-opt-report [n]           generate an optimization report to stderr
             0            disable optimization report output
             1            minimum report output
             2            medium output (DEFAULT when enabled)
             3            maximum report output
-opt-report-file<file>    specify the filename for the generated report
-opt-report-level[level]  specify the level of report verbosity (min|med|max)
-opt-report-phase<name>   specify the phase that reports are generated against
-opt-report-routine<name> reports on routines containing the given name
-opt-report-help          display the optimization phases available for
                          reporting

-tcheck [mode]        enable analysis of threaded applications (requires
                      Intel(R) Thread Checker; cannot be used with compiler
                      alone)
           tci        instruments a program to perform thread-count-
                      independent analysis
           tcd        instruments a program to perform thread-count-
                      dependent analysis (DEFAULT when mode is not used)
           api        instruments a program at the api-imports level
-tprofile             generate instrumentation to analyze multi-threading
                      performance (requires Intel(R) Thread Profiler; cannot
                      be used with compiler alone)
-tcollect[=<lib>]     insert instrumentation probes calling the Intel(R)
                      Trace Collector API.  The library -l<lib> is linked in
                      the default being -lVT (requires Intel(R) Trace
                      Collector)

OpenMP and Parallel Processing
------------------------------
-openmp               enable the compiler to generate multi-threaded code
```

```
                        based on the OpenMP directives
-openmp-profile         enable analysis of OpenMP application when the Intel(R)
                        Thread Profiler is installed
-openmp-stubs           enables the user to compile OpenMP programs in
                        sequential mode.  The openmp directives are ignored and
                        a stub OpenMP library is linked (sequential)
-openmp-report{0|1|2} control the OpenMP parallelizer diagnostic level

-parallel               enable the auto-parallelizer to generate multi-threaded
                        code for loops that can be safely executed in parallel
-par-report{0|1|2|3}  control the auto-parallelizer diagnostic level
-par-threshold[n]       set threshold for the auto-parallelization of loops
                        where n is an integer from 0 to 100
-opt-streaming-stores <arg>
                         specifies whether streaming stores are generated
                         always - enables generation of streaming stores under
                                  the assumption that the application is memory
                                  bound
                         auto   - compiler decides when streaming stores are
                                  used (DEFAULT)
                         never  - disables generation of streaming stores

Floating Point
--------------
-fp-model <name>    enable <name> floating point model variation
           [no-]except - enable/disable floating point semantics
           double      - rounds intermediates in 53-bit (double) precision
           extended    - rounds intermediates in 64-bit (extended) precision
           fast[=1|2]  - enables more aggressive floating point optimizations
           precise     - allows value-safe optimizations
           source      - enables intermediates in source precision
           strict      - enables -fp-model precise -fp-model except, disables
                         contractions and enables pragma stdc fenv_access
-fp-speculation<mode>
           enable floation point speculations with the following <mode>
           conditions:
           fast   - speculate floating point operations (DEFAULT)
           safe   - speculate only when safe
           strict - same as off
           off    - disables speculation of floating-point operations
-mp           maintain floating point precision (disables some optimizations)
-mp1          improve floating-point precision (speed impact is less than -mp)
-m[no-]ieee-fp
              same as -mp
-[no-]prec-sqrt
              determine if certain square root optimizations are enabled
-[no-]fp-port round fp results at assignments & casts (some speed impact)
-fp-stack-check enable fp stack checking after every function/procedure call
-pc32         set internal FPU precision to 24 bit significand
-pc64         set internal FPU precision to 53 bit significand
-pc80         set internal FPU precision to 64 bit significand (DEFAULT)
-rcd          rounding mode to enable fast float-to-int conversions
-[no-]prec-div
              improve precision of floating-point divides (some speed impact)
-ssp                    enable software-based speculative pre-computation

Preprocessor
------------
```

```
-A<name>[(val)] create an assertion 'name' having value 'val'
-A-        remove all predefined macros
-C         don't strip comments
-D<name>[=<text>]  define macro
-E         preprocess to stdout
-EP        preprocess to stdout omitting #line directives
-P         preprocess to file omitting #line directives
-I<dir>    add directory to include file search path
-idirafter<dir>
           add directory to the second include file search path (after -I)
-isystem<dir>
           add directory to the start of the system include path
-iprefix <prefix>
           use <prefix> with -iwithprefix as a prefix
-iwithprefix <dir>
           append <dir> to the prefix passed in by -iprefix and put it on
           the include search path at the end of the include directories
-iwithprefixbefore <dir>
           similar to -iwithprefix except the include directory is placed
           in the same place as -I command line include directories
-iquote <dir>
           add directory to the front of the include file search path for
           files included with quotes, but not brackets
-U<name>   remove predefined macro
-imacros <file>
           treat <file> as an #include file, but throw away all preprocessing
           while macros defined remain defined
-X         remove standard directories from include file search path
-nostdinc  same as -X
-nostdinc++
           remove standard C++ directories from include file search path
-H         print include file order
-M         generate makefile dependency information
-MM        similar to -M, but do not include system header files
-MG        similar to -M, but treat missing header files as generated files
-MD        preprocess and compile, generating output file containing
           dependency information ending with extension .d
-MMD       similar to -MD, but do not include system header files
-MF<file>  generate makefile dependency information in file (must specify -M
           or -MM)
-MP        add a phony target for each dependency
-MT<target>
           change the default target rule for dependency generation
-MQ<target>
           same as -MT, but quotes special Make characters
-dM        output macro definitions in effect after preprocessing (use with -E)
-dD        same as -dM, but output #define directives in preprocessed source
-dN        same as -dD, but #define directives contain only macro names
-gcc       Predefine the __GNUC__, __GNUC_MINOR__, and __GNUC_PATCHLEVEL__
           macros (DEFAULT).
-no-gcc    Do not predefine GNUC macros listed in -gcc mode. Warning: can
           prevent correct system header compilation, see -gcc-sys
-gcc-sys   same as -no-gcc, except that the GNU macros are defined only
           while preprocessing the system include headers
-no-icc    do not predefine the __ICC and __INTEL_COMPILER macros.
           Warning: can prevent correct Intel header compilation
-pragma-optimization-level=[Intel|GCC]
           process #pragma optimize using Intel (DEFAULT) or GCC syntax
```

```
Output, Debug, PCH
------------------
-c          compile to object (.o) only, do not link
-S          compile to assembly (.s) only, do not link (*I)
-[no-]use-asm
            produce objects through assembler
-use-msasm Support Microsoft style assembly language insertion using MASM
            style syntax
-fcode-asm produce assembly file with optional code annotations (requires -S)
-fsource-asm
            produce assembly file with optional source annotations (requires -S)
-f[no-]verbose-asm
            produce assembly file with compiler comments (DEFAULT) (requires -S)
-o<file>    name output file
-g          produce symbolic debug information in object file (implies -O0 when
            another optimization option is not explicitly set)
-g0         disable generation of symbolic debug information
-gdwarf-2   enable generation of debug information using the DWARF2 format
-debug [keyword]
            enable debug information and control output of enhanced
            debug information.
            keywords:  all, full, minimal, none, [no]inline-debug-info,
            [no]variable-locations, [no]semantic-stepping, extended,
            [no]expr-source-pos
-inline-debug-info preserve the source position of inlined code instead
            of assigning the call-site source position to inlined code
-ftrapuv    trap uninitialized variables
-pch        enable automatic precompiled header file creation/usage
-pch-create <file> create precompiled header file
-pch-use <file>    use precompiled header file
-pch-dir <dir>     name precompiled header directory
-map-opts   enable option mapping tool
-print-multi-lib
            print information about libraries being used

Data
----
-f[no-]pic, -f[no-]PIC
               required to build fully preemptable and position independent code
               for shared objects (OFF by default)
-fvisibility=[extern|default|protected|hidden|internal]
               Global symbols (data and functions) will get the visibility
               attribute given by default.  Symbol visibility attributes
               explicitly set in the source code or using the symbol visibility
               attribute file options will override the -fvisibility setting
-fvisibility-extern=<file>
               Space separated symbols listed in the <file> argument will get
               visibility set to extern
-fvisibility-default=<file>
               Space separated symbols listed in the <file> argument will get
               visibility set to default
-fvisibility-protected=<file>
               Space separated symbols listed in the <file> argument will get
               visibility set to protected
-fvisibility-hidden=<file>
               Space separated symbols listed in the <file> argument will get
               visibility set to hidden
```

```
-fvisibility-internal=<file>
              Space separated symbols listed in the <file> argument will get
              visibility set to internal
-fvisibility-inlines-hidden
              mark inline member functions as hidden
-fminshared   Compilation is for the main executable. Absolute addressing can
              be used and non-position independent code generated for symbols
              that are at least protected
-f[no-]common Enables the compiler to treat common variables as if they were
              defined.  That in turn allows the use of gprel addressing of
              common data variables
-fpack-struct pack structure members together
-freg-struct-return
              return struct and union values in registers when possible
-f[no-]math-errno
              set ERRNO after calling standard math library functions
-fstack-security-check
              enable overflow security checks
-long-double  enable 80-bit 'long double'
-no-bss-init  disable placement of zero-initialized variables in BSS (use DATA)
-[no-]global-hoist
              enable(DEFAULT)/disable external globals are load safe

Miscellaneous
-------------
-V            display compiler version information
--version     display GCC style version information
-dumpversion  display the compiler version number only
-dryrun       show driver tool commands but do not execute tools
-v [file]     show driver tool commands and execute tools
-[no-]sox     enable/disable(DEFAULT) saving of compiler options and version
              in the executable
-save-temps   Store the intermediate files in current directory and name
              them based on the source file.
-f[no-]keep-static-consts
              enable/disable(DEFAULT) emission of static const variables even
              when not referenced

Component Control
-----------------
-Qoption,<str>,<opts>   pass options <opts> to tool specified by <str>
-Qlocation,<str>,<dir>  set <dir> as the location of tool specified by <str>
-Qinstall <dir>         set <dir> as root of compiler installation

Linking/Linker
--------------
-L<dir>        instruct linker to search <dir> for libraries
-l<string>     instruct the linker to link in the -l<string> library
-shared-intel  link Intel provided libraries dynamically
-static-intel  link Intel provided libraries statically
-shared-libgcc link libgcc dynamically
-static-libgcc link libgcc statically
-dynamic-linker<file>
               select dynamic linker other than the default
-[no-]cxxlib   do not link in C++ runtime libraries
-cxxlib[=dir]  link using C++ run-time libraries provided with gcc
               dir is an optional top-level location for the gcc
               binaries and libraries
```

```
-nodefaultlibs do not use standard libraries when linking
-nostartfiles  do not use standard startup files when linking
-nostdlib      do not use standard libraries and startup files when linking
-static        prevents linking with shared libraries
-shared        produce a shared object
-Bstatic       specify following libraries are linked statically
-Bdynamic      specify following libraries are linked dynamically
-static-libcxa link Intel libcxa C++ library statically
-shared-libcxa link Intel libcxa C++ library dynamically, overrides the default
               behavior when -static is used
-pthread       use pthreads library for multithreading support
-cxxlib-<mode> tell the compiler which C++ run-time libraries to use
               nostd     - do not link in standard C++ library
-u <symbol>    pretend the <symbol> is undefined
-T <file>      direct linker to read link commands from <file>
-Xlinker <val> pass <val> directly to the linker for processing
-Wa,<o1>[,<o2>,...] pass options o1, o2, etc. to the assembler
-Wl,<o1>[,<o2>,...] pass options o1, o2, etc. to the linker for processing
-Wp,<o1>[,<o2>,...] pass options o1, o2, etc. to the preprocessor

-help [category]       print full or category help message
 Valid categories include
       advanced        - Advanced Optimization
       codegen         - Code Generation
       compatibility   - Compatibility
       component       - Component Control
       data            - Data
       deprecated      - Deprecated Options
       diagnostics     - Compiler Diagnostics
       float           - Floating Point
       help            - Help
       inline          - Inlining
       ipo             - Interprocedural Optimizations (IPO)
       language        - Language
       link            - Linking/Linker
       misc            - Miscellaneous
       openmp          - OpenMP and Parallel Processing
       opt             - Optimization
       output          - Output
       pgo             - Profile Guided Optimization (PGO)
       preproc         - Preprocessor
       reports         - Optimization Reports
```