XCC Command-Line Manual

IN THIS DOCUMENT

XCC is the front-end to the xCORE C, C++ and XC compilers. Typical usage results in preprocessing, compilation, assembly, linking, and mapping code and data onto tiles. Some options allow this process to be stopped at intermediate stages and other options are passed to one stage of processing. Most options have negative forms (for example, `-fno-`*option*). A space between an option and its argument is permitted.

Build settings for an application are defined in the application Makefile. Double click the Makefile in the *Project Explorer* to open it in the *Makefile Editor*.

1 Overall Options

The four possible stages of compilation are preprocessing, compilation proper, assembly and linking/mapping. The first three stages are applied to an individual source file, producing an object file. Linking and mapping combine the object files and an XN file into a single executable XE file, which contains the code and data segments for each tile.

Publication Date: 2013/11/12 REV E

source-file The suffix of a source file determines how it is handled by default.

Extension	Type of File	Preprocessed by XCC
`.xc`	XC source code	Y
`.c`	C source code	Y
`.cpp`	CPP source code (for compatability, the extensions `cc`, `cp`, `c++`, `C` and `cxx` are also recognized)	Y
`.S`	Assembly code	Y
`.xta`	xCORE Timing Analyzer script	N
`.xn`	xCORE Network Description	N
`.xi`	XC source code	N
`.i`	C source code	N
`.ii`	C++ source code	N
`.s`	Assembly code	N
other	Object file .o be given to the linker	N

Figure 1: File extensions recognized by XCC and their meaning

`-x`*language* Specifies the language for the following input files. This option applies to all following input files until the next `-x` option. Supported values for *language* are:

`xc`
`c`
`c++`
`assembler`
`assembler-with-cpp`
`xn`
`xta`
`none` (turn off language specification)

`-std=`*standard*
Specifies the language variant for the following input C or C++ file. Supported values for *standard* are:

`c89`
: ISO C89

`gnu89`
: ISO C89 with GNU extensions

`c99`
: ISO C99

`gnu99`
: ISO C99 with GNU extensions (default for C programs)

`c++98`
: ISO C++ (1998)

`gnu++98`
: ISO C++ (1998) with GNU extensions (default for C++ programs)

`-fsubword-select`
: In XC, allows selecting on channel inputs where the size of the desstination variable is less than 32 bits.

  This is default for targets based on XS1-L devices. It is not default for targets based on XS1-G devices. For further details, see XM-000971-PC.

`-target=`*platform*
: Specifies the target platform. The platform configuration must be specified in the file `platform.xn`, which is searched for in the paths specified by the `XCC_DEVICE_PATH` environment variable (see §8).

`-foverlay`
: Enable support for memory overlays. Functions marked as overlay roots are placed in external memory and are loaded on demand at runtime. The option should be passed when compiling and linking. An overlay runtime should be supplied in the application.

`-foverlay=flash`
: Enable support for memory overlays linking in the flash overlay runtime. Overlays are only enabled on tiles which boot from flash.

`-foverlay=syscall`
: Enable support for memory overlays linking in the syscall overlay runtime. Overlay are enabled on all tiles. Overlays are loaded from a host machine using a system call.

`-fxscope`
: Enable support for xSCOPE tracing. The XN file of the target must contain an xSCOPE link. The option should be passed when compiling and linking.

`-funroll-loops`
: Unroll loops with small iteration counts. This is enabled at -O2 and above.

`-finline-functions`
: Integrate simple functions into their callers. This is enabled at -O2 and above and also at -Os.

`-pass-exit-codes`
: Returns the numerically highest error code produced by any phase of compilation. (By default XCC returns 1 if any phase of the compiler returns non-success, otherwise it returns 0.)

`-c`
: Compiles or assembles the source files, producing an object file for each source file, but does not link/map. By default the object filename is formed by replacing the source file suffix with `.o` (for example, `a.c` produces `a.o`).

`-S`
: Stops after compilation proper, producing an assembly code file for each nonassembly input file specified. By default the assembly filename is formed by replacing the source file suffix with .s.

Input files not requiring compilation are ignored.

`-E` Preprocesses the source files only, outputting the preprocessed source to stdout.

Input files not requiring preprocessing are ignored.

`-o`*file* Places output in *file*.

If `-o` is not specified, the executable file is placed in `a.xe`, the object file for `source.suffix` in `source.o`, its assembly code file in `source.s`, and all preprocessed C/C++/XC source on standard output.

`-v` Prints (on standard error) the commands executed at each stage of compilation. Also prints the version number of XCC, the preprocessor and the compiler proper.

`-###` The same as `-v` except that the commands are not executed and all command arguments are quoted.

`--help` Prints a description of the supported command line options. If the `-v` option is also specified, `--help` is also passed to the subprocesses invoked by XCC.

`--version` Displays the version number and copyrights.

2 Warning Options

Many specific warnings can be controlled with options beginning `-W`. Each of the following options has a negative form beginning `-Wno-` to turn off warnings.

`-fsyntax-only`
Checks the code for syntax errors only, then exits.

`-w` Turns off all warning messages.

`-Wbidirectional-buffered-port`
Warns about the use of buffered ports not qualified with either `in` or `out`. This warning is enabled by default.

`-Wchar-subscripts`
Warns if an array subscript has type char.

`-Wcomment` Warns if a comment-start sequence `/*` appears in a `/*` comment, or if a backslash-newline appears in a `//` comment. This is default.

`-Wimplicit-int`
Warns if a declaration does not specify a type. In C also warns about function declarations with no return type.

`-Wmain` Warns if the type of `main` is not a function with external linkage returning `int`. In XC also warns if main does not take zero arguments. In C also warns if `main` does not take either zero or two arguments of appropriate type.

`-Wmissing-braces`
Warns if an aggregate or union initializer is not fully bracketed.

`-Wparentheses`
Warns if parentheses are omitted when there is an assignment in a context where a truth value is expected or if operators are nested whose precedence people often find confusing.

`-Wreturn-type`
Warns if a function is defined with a return type that defaults to `int` or if a return statement returns no value in a function whose return type is not `void`.

`-Wswitch-default`
Warns if a `switch` statement does not have a `default` case.

`-Wswitch-fallthrough`
(XC only) Warns if a case in a `switch` statement with at least one statement can have control fall through to the following case.

`-Wtiming` Warns if timing constraints are not satisfied. This is default.

`-Wtiming-syntax`
Warns about invalid syntax in timing scripts. This is default.

`-Wunused-function`
Warns if a static function is declared but not defined or a non-inline static function is unused.

`-Wunused-parameter`
Warns if a function parameter is unused except for its declaration.

`-Wunused-variable`
Warns if a local variable or non-constant static variable is unused except for its declaration.

`-Wunused` Same as `-Wunused-function`, `-Wunused-variable` and `-Wno-unused-parameter`.

`-Wall` Turns on all of the above `-W` options.

The following `-W...` options are not implied by `-Wall`.

`-Wextra`
`-W` Prints extra warning messages for the following:

- A function can return either with or without a value (C, C++ only).
- An expression statement or left-hand side of a comma expression contains no side effects. This warning can be suppressed by casting the unused expression to `void` (C, C++ only).
- An unsigned value is compared against zero with < or <=.
- Storage-class specifiers like `static` are not the first things in a declaration (C, C++ only).
- A comparison such as `x<=y<=z` appears (XC only).
- The return type of a function has a redundant qualifier such as `const`.
- Warns about unused arguments if `-Wall` or `-Wunused` is also specified.
- A comparison between signed and unsigned values could produce an incorrect result when the signed value is converted to unsigned. (Not warned if `-Wno-sign-compare` is also specified.)
- An aggregate has an initializer that does not initialize all members.
- An initialized field without side effects is overridden when using designated initializers (C, C++ only).
- A function parameter is declared without a type specifier in K&R-style functions (C, C++ only).
- An empty body occurs in an `if` or `else` statement (C, C++ only).
- A pointer is compared against integer zero with <, <=, >, or >=. (C, C++ only).
- An enumerator and a non-enumerator both appear in a conditional expression. (C++ only).
- A non-static reference or non-static const enumerator and a non-enumerator both appear in a conditional expression (C++ only).
- Ambiguous virtual bases (C++ only).
- Subscripting an array which has been declared `register` (C++ only).
- Taking the address of a variable which has been declared `register` (C++ only).
- A base class is not initialized in a derived class' copy constructor (C++ only).

`-Wconversion`

Warns if a negative integer constant expression is implicitly converted to an unsigned type.

`-Wdiv-by-zero`

Warns about compile-time integer division by zero. This is default.

`-Wfloat-equal`
: Warns if floating point values are used in equality comparisons.

`-Wlarger-than-`*len*
: Warns if an object of larger than *len* bytes is defined.

`-Wpadded`
: Warns if a structure contains padding. (It may be possible to rearrange the fields of the structure to reduce padding and thus make the structure smaller.)

`-Wreinterpret-alignment`
: Warns when a reinterpret cast moves to a larger alignment.

`-Wshadow`
: Warns if a local variable shadows another local variable, parameter or global variable or if a built-in function is shadowed.

`-Wsign-compare`
: Warns if a comparison between signed and unsigned values could produce an incorrect result when the signed value is converted to unsigned.

`-Wsystem-headers`
: Prints warning messages for constructs found in system header files. This is not default. See §7.

`-Wundef`
: Warns if an undefined macro is used in a `#if` directive.

`-Werror`
: Treat all warnings as errors.

`-Werror=`*option*
: Turns a warning message into an error. The option should be one of the warning options to the compiler that can be prefixed with `-W`.

  By default, the flag `-Werror=timing-syntax` is set. Turning this warning into an error implies that timing warnings (`-Wtiming`) are also errors and vice versa.

3 Debugging Options

`-g`
: Produces debugging information.

`-fxta-info`
: Produces timing information for use with XTA. This is default.

`-fresource-checks`
: Produces code in the executable that traps if a resource allocation fails. This causes resource errors to be detected as early as possible.

`-save-temps`
: Saves the intermediate files. These files are placed in the current directory and named based on the source file.

`-fverbose-asm`
: Produces extra compilation information as comments in intermediate assembly files.

`-dumpmachine`
: Prints the target machine and exit.

`-dumpversion`
: Prints the compiler version and exit.

`-print-multi-lib`
: Prints the mapping from multilib directory names to compiler switches that enable them. The directory name is seperated from the switches by ';', and each switch starts with a '@' instead of the '-', without spaces between multiple switches.

`-print-targets`
: Prints the target platforms supported by the compiler. The target names correspond to strings accepted by the `-target` option.

4 Optimization Options

Turning on optimization makes the compiler attempt to improve performance and/or code size at the expense of compilation time and the ability to debug the program.

`-O0`
: Do not optimize. This is the default.

`-O`
`-O1`
: Optimize. Attempts to reduce execution time and code size without performing any optimizations that take a large amount of compilation time.

`-O2`
: Optimize more. None of these optimizations involve a space-speed tradeoff.

`-O3`
: Optimize even more. These optimizations may involve a space-speed tradeoff; high performance is preferred to small code size.

`-Os`
: Optimize for the smallest code size possible.

`-fschedule`
: Attempt to reorder instructions to increase performance. This is not default at any optimization level.

5 Preprocessor Options

The following options control the preprocessor.

`-E` Preprocesses only, then exit.

`-D`*name* Predefines *name* as a macro with definition `1`.

`-D`*name*`=`*definition*
Tokenizes and preprocesses the contents of *definition* as if it appeared in a `#define` directive.

`-U`*name* Removes any previous definition of *name*.

`-D` and `-U` options are processed in the order given on the command line.

`-MD` Outputs to a file a rule suitable for `make` describing the dependencies of the source file. The default name of the dependency file is determined based on whether the `-o` option is specified. If `-o` is specified, the filename is the basename of the argument to `-o` with the suffix `.d`. If `-o` is not specified, the filename is the basename of the input file with the suffix `.d`. The name of the file may be overriden with `-MF`.

`-MMD` The same as `-MD` expect that dependencies on system headers are ignored.

`-MF` *file* Specifies the file to write dependency information to.

`-MP` Emits phony targets for each dependency of the source file. Each phony target depends on nothing. These dummy rules work around errors `make` gives if header files are removed without updating the `Makefile` to match.

`-MT` *file* Specifies the target of the rule emitted by dependency generation.

6 Linker And Mapper Options

The following options control the linker/mapper.

`-l`*library* Searches the library library when linking. The linker searches and processes libraries and object files in the order specified. The actual library name searched for is `lib`*library*`.a`.

The directories searched include any specified with `-L`.

Libraries are archive files whose members are object files. The linker scans the archive for its members which define symbols that have so far been referenced but not defined.

`-nostartfiles`
: Do not link with the system startup files.

`-nodefaultlibs`
: Do not link with the system libraries.

`-nostdlib` Do not link with the system startup files or system libraries.

`-s` Removes all symbol table and relocation information from the executable.

`-default-clkblk` *clk*
: Use *clk* as the default clock block. The clock block may be specified by its name in `<xs1.h>` or by its resource number.

  The startup code turns on the default clock block, configures it to be clocked off the reference clock with no divide and puts it into a running state. Ports declared in XC are initially attached to the default clock block. If this option is unspecified, the default clock block is set to `XS1_CLKBLK_REF`.

`-Wm,`*option* Passes *option* as an option to the linker/mapper. If *option* contains commas, it is split into multiple options at the commas.

  To view the full set of advanced mapper options, type `xmap --help`.

`-Xmapper` *option*
: Passes option as an option to the linker/mapper. To pass an option that takes an argument use `-Xmapper` twice.

`-report` Prints a summary of resource usage.

7 Directory Options

The following options specify directories to search for header files and libraries.

`-I`*dir* Adds *dir* to the list of directories to be searched for header files.

`-isystem`*dir* Searches *dir* for header files after all directories specified by `-I`. Marks it as a system directory.

  The compiler suppresses warnings for header files in system directories.

`-iquote`*dir* Searches *dir* only for header files requested with `#include "file"` (not with `#include <file>`) before all directories specified by `-I` and before the system directories.

`-L`*dir* Adds *dir* to the list of directories to be searched for by `-l`.

8 Environment Variables Affecting XCC

The following environment variables affect the operation of XCC. Multiple paths are separated by an OS-specific path separator (';' for Windows, ':' for Mac and Linux).

`XCC_INCLUDE_PATH`
: A list of directories to be searched as if specified with `-I`, but after any paths given with `-I` options on the command line.

`XCC_XC_INCLUDE_PATH`
`XCC_C_INCLUDE_PATH`
`XCC_CPLUS_INCLUDE_PATH`
`XCC_ASSEMBLER_INCLUDE_PATH`
: Each of these environment variables applies only when preprocessing files of the named language. The variables specify lists of directories to be searched as if specified with `-isystem`, but after any paths given with `-isystem` options on the command line.

`XCC_LIBRARY_PATH`
: A list of directories to be searched as if specified with `-L`, but after any paths given with `-L` on the command line.

`XCC_DEVICE_PATH`
: A list of directories to be searched for device configuration files.

`XCC_EXEC_PREFIX`
: If set, subprograms executed by the compiler are prefixed with the value of this environment variable. No directory seperated is added when the prefix is combined with the name of a subprogram. The prefix is not applied when executing the assembler or the mapper.

`XCC_DEFAULT_TARGET`
: The default target platform, to be located as if specified with `-target=`. The default target platform is used if no target is specified with `-target=` and no XN file is passed.

9 Board Support Provided by `<platform.h>`

During compilation of a program, the compiler generates a temporary header file named `platform.h` that contains variable and macro definitions, as defined by the target XN file, which includes:

- Declarations of variables of type `tileref` (see XM-000929-PC).
- Macro definitions of port names (see XM-000929-PC).