Compiler Options Listed by Category

Visual Studio 2015

The new home for Visual Studio documentation is Visual Studio 2017 Documentation on docs.microsoft.com.

The latest version of this topic can be found at Compiler Options Listed by Category.

This article contains a categorical list of compiler options. For an alphabetical list, see Compiler Options Listed Alphabetically.

Optimization

Option	Purpose
/O1	Creates small code.
/O2	Creates fast code.
/Ob	Controls inline expansion.
/Od	Disables optimization.
/Og	Deprecated. Uses global optimizations.
/Oi	Generates intrinsic functions.
/Os	Favors small code.
/Ot	Favors fast code.
/Ox	Uses maximum optimization (/Ob2gity /Gs).
/Oy	Omits frame pointer. (x86 only)
/favor	Produces code that is optimized for a specified architecture, or for a range of architectures.

Code Generation

Option	Purpose
/arch	Use SSE or SSE2 instructions in code generation. (x86 only)
/clr	Produces an output file to run on the common language runtime.
/EH	Specifies the model of exception handling.

Option	Purpose
/fp	Specifies floating-point behavior.
/GA	Optimizes for Windows applications.
/Gd	Uses the `__cdecl` calling convention. (x86 only)
/Ge	Deprecated. Activates stack probes.
/GF	Enables string pooling.
/Gh	Calls hook function `_penter`.
/GH	Calls hook function `_pexit`.
/GL	Enables whole program optimization.
/Gm	Enables minimal rebuild.
/GR	Enables run-time type information (RTTI).
/Gr	Uses the `__fastcall` calling convention. (x86 only)
/GS	Checks buffer security.
/Gs	Controls stack probes.
/GT	Supports fiber safety for data allocated by using static thread-local storage.
/guard:cf	Adds control flow guard security checks.
/Gv	Uses the `__vectorcall` calling convention. (x86 and x64 only)
/Gw	Enables whole-program global data optimization.
/GX	Deprecated. Enables synchronous exception handling. Use /EH instead.
/Gy	Enables function-level linking.
/GZ	Deprecated. Enables fast checks. (Same as /RTC1)
/Gz	Uses the `__stdcall` calling convention. (x86 only)
/homeparams	Forces parameters passed in registers to be written to their locations on the stack upon function entry. This compiler option is only for the x64 compilers (native and cross compile).
/hotpatch	Creates a hotpatchable image.
/Qfast_transcendentals	Generates fast transcendentals.

Option	Purpose
Qlfist	Deprecated. Suppresses the call of the helper function `_ftol` when a conversion from a floating-point type to an integral type is required. (x86 only)
/Qimprecise_fwaits	Removes `fwait` commands inside `try` blocks.
/Qpar	Enables automatic parallelization of loops.
/Qpar-report	Enables reporting levels for automatic parallelization.
/Qsafe_fp_loads	Uses integer move instructions for floating-point values and disables certain floating point load optimizations.
/Qvec-report (Auto-Vectorizer Reporting Level)	Enables reporting levels for automatic vectorization.
/RTC	Enables run-time error checking.
/volatile	Selects how the volatile keyword is interpreted.

Output Files

Option	Purpose
/doc	Processes documentation comments to an XML file.
/FA	Configures an assembly listing file.
/Fa	Creates an assembly listing file.
/Fd	Renames program database file.
/Fe	Renames the executable file.
/Fi	Specifies the preprocessed output file name.
/Fm	Creates a mapfile.
/Fo	Creates an object file.
/Fp	Specifies a precompiled header file name.
/FR /Fr	Generates browser files.

Preprocessor

Option	Purpose
/AI	Specifies a directory to search to resolve file references passed to the #using directive.
/C	Preserves comments during preprocessing.
/D	Defines constants and macros.
/E	Copies preprocessor output to standard output.
/EP	Copies preprocessor output to standard output.
/FI	Preprocesses the specified include file.
/FU	Forces the use of a file name, as if it had been passed to the #using directive.
/Fx	Merges injected code with the source file.
/I	Searches a directory for include files.
/P	Writes preprocessor output to a file.
/U	Removes a predefined macro.
/u	Removes all predefined macros.
/X	Ignores the standard include directory.

Language

Option	Purpose
/openmp	Enables #pragma omp in source code.
/vd	Suppresses or enables hidden `vtordisp` class members.
/vmb	Uses best base for pointers to members.
/vmg	Uses full generality for pointers to members.
/vmm	Declares multiple inheritance.
/vms	Declares single inheritance.
/vmv	Declares virtual inheritance.
/Z7	Generates C 7.0–compatible debugging information.

Option	Purpose
/Za	Disables language extensions.
/Zc	Specifies standard behavior under /Ze.
/Ze	Deprecated. Enables language extensions.
/ZI	Includes debug information in a program database compatible with Edit and Continue. (x86 only)
/Zi	Generates complete debugging information.
/Zl	Removes the default library name from the .obj file.
/Zp *n*	Packs structure members.
/Zs	Checks syntax only.
/ZW	Produces an output file to run on the Windows Runtime.

Linking

Option	Purpose
/F	Sets stack size.
/LD	Creates a dynamic-link library.
/LDd	Creates a debug dynamic-link library.
/link	Passes the specified option to LINK.
/LN	Creates an MSIL module.
/MD	Compiles to create a multithreaded DLL, by using MSVCRT.lib.
/MDd	Compiles to create a debug multithreaded DLL, by using MSVCRTD.lib.
/MT	Compiles to create a multithreaded executable file, by using LIBCMT.lib.
/MTd	Compiles to create a debug multithreaded executable file, by using LIBCMTD.lib.

Precompiled Header

Option	Purpose

Option	Purpose
/Y-	Ignores all other precompiled-header compiler options in the current build.
/Yc	Creates a precompiled header file.
/Yd	Places complete debugging information in all object files.
/Yu	Uses a precompiled header file during build.

Miscellaneous

Option	Purpose
/?	Lists the compiler options.
@	Specifies a response file.
/analyze	Enables code analysis.
/bigobj	Increases the number of addressable sections in an .obj file.
/c	Compiles without linking.
/cgthreads	Specifies number of cl.exe threads to use for optimization and code generation.
/errorReport	Enables you to provide internal compiler error (ICE) information directly to the Visual C++ team.
/FC	Displays the full path of source code files passed to cl.exe in diagnostic text.
/FS	Forces writes to the program database (PDB) file to be serialized through MSPDBSRV.EXE.
/H	Deprecated. Restricts the length of external (public) names.
/HELP	Lists the compiler options.
/J	Changes the default `char` type.
/kernel	The compiler and linker will create a binary that can be executed in the Windows kernel.
/MP	Builds multiple source files concurrently.
/nologo	Suppresses display of sign-on banner.
/sdl	Enables additional security features and warnings.
/showIncludes	Displays a list of all include files during compilation.

Option	Purpose
/Tc /TC	Specifies a C source file.
/Tp /TP	Specifies a C++ source file.
/V	Deprecated. Sets the version string.
/w	Disables all warnings.
/W0, /W1, /W2, /W3, /W4	Sets output warning level.
/w1, /w2, /w3, /w4	Sets warning level for the specified warning.
/Wall	Enables all warnings, including warnings that are disabled by default.
/wd	Disables the specified warning.
/we	Treats the specified warning as an error.
/WL	Enables one-line diagnostics for error and warning messages when compiling C++ source code from the command line.
/wo	Displays the specified warning only once.
/Wp64	Obsolete. Detects 64-bit portability problems.
/Wv	Disables warnings introduced by later versions of the compiler.
/WX	Treats warnings as errors.
/Yd	Deprecated. Places complete debugging information in all object files. Use /Zi instead.
/Yl	Injects a PCH reference when creating a debug library.
/Zm	Specifies the precompiled header memory allocation limit.

Deprecated and Removed Compiler Options

Option	Purpose
/clr:noAssembly	Deprecated. Use /LN (Create MSIL Module) instead.
/Fr	Deprecated. Creates a browse information file without local variables.
/Ge	Deprecated. Activates stack probes. On by default.

Option	Purpose
/GX	Deprecated. Enables synchronous exception handling. Use /EH instead.
/GZ	Deprecated. Enables fast checks. Use /RTC1 instead.
/H	Deprecated. Restricts the length of external (public) names.
/Og	Deprecated. Uses global optimizations.
QIfist	Deprecated. Once used to specify how to convert from a floating-point type to an integral type.
/V	Deprecated. Sets the .obj file version string.
/Yd	Deprecated. Places complete debugging information in all object files. Use /Zi instead.
/Zc:forScope-	Deprecated. Disables conformance in for loop scope.
/Ze	Deprecated. Enables language extensions.
/Zg	Removed in Visual C++ 2015. Generates function prototypes.

See Also

C/C++ Building Reference
Compiler Options
Setting Compiler Options