

Forum:[Hack Forums](#)

Thread: [Optimize your Visual Studio compile settings for Filesize \(C++\)](#)

WhiteTV - February 17, 2025, 7:25 am

Generally in today's world you usually don't care anymore how big your compiled binary is. But in our domain, especially when it comes to Red Teaming and Malware development, you want the smallest possible filesize.

Apart from optimized code, without external dependencies (looking at you OpenSSL with 5MB of size), you can optimize your Compile settings for reduced size.

The following Visual Studio (2022) Settings are the ones I use. Compared to the default settings, these reduced the statically linked binary size in my case from 480kb to 300kb.

Note that the following settings are for WinAPI applications. If you want a console window, skip the SubSystem: Windows (/SUBSYSTEM:WINDOWS) setting

Also some settings might not be available depending which VS or C++ versions you are using.

Project Properties -> C/C++ -> Optimization: Optimization: Maximum Optimization (Favor Size) (/O1)

Favor Size or Speed: Favor Small Code (/Os)

Whole Program Optimization: Yes (/GL)

Enable Function-Level Linking: Yes (/Gy)

Omit Frame Pointers: Yes (/Oy)

Project Properties -> C/C++ -> Code Generation: Enable String Pooling: Yes (/GF)

Runtime Library: Multi-threaded (MT) for static linking

Buffer Security Check: No (/GS-)

Control Flow Guard: No

Project Properties -> C/C++ -> Preprocessor: Add: NDEBUG;_RELEASE

Project Properties -> Linker -> Optimization:

Link Time Code Generation: Use Link Time Code Generation (/LTCG)

References: Eliminate Unreferenced Data (/OPT:REF)

Enable COMDAT Folding: Remove Redundant COMDATs (/OPT:ICF)

Project Properties -> Linker -> Debugging: Generate Debug Info: No

Strip Private Symbols: Yes

Project Properties -> Linker -> System: SubSystem: Windows (/SUBSYSTEM:WINDOWS)

Enable Large Addresses: No

Project Properties -> Linker -> Command Line: Add these additional options:

Code /EMITPOGOPHASEINFO:NO /INCREMENTAL:NO /RELEASE

Project Properties -> C/C++ -> Advanced: Disable Specific Warnings: 4711;4710 (to disable inline expansion warnings)

Inline Function Expansion: Only __inline (/Ob1)

Project Properties -> C/C++ -> General: Debug Information Format: None

Project Properties -> Build Events -> Post-Build Event: Add command to strip debug symbols:

Code strip "\$(TargetPath)"

Edit: To further analyze your binary, SizeBench from Windows can be used to exactly show you which libraries are taking up a lot of space in your compiled Binary.

<https://github.com/microsoft/SizeBench>

