CompilingOpenSSLwithMinGW

<https://wiki.qt.io/Compiling_OpenSSL_with_MinGW>

# Compiling OpenSSL with MinGW

Qt 5 / QtWebkit can make use of OpenSSL, which MinGW distributions do not ship.

**Contents**

 [[hide](https://wiki.qt.io/Compiling_OpenSSL_with_MinGW)]

* [1 Pre-built packages](https://wiki.qt.io/Compiling_OpenSSL_with_MinGW#Pre-built_packages)
* [2 Compiling on your own](https://wiki.qt.io/Compiling_OpenSSL_with_MinGW#Compiling_on_your_own)
  + [2.1 Requirements](https://wiki.qt.io/Compiling_OpenSSL_with_MinGW#Requirements)
  + [2.2 How to build](https://wiki.qt.io/Compiling_OpenSSL_with_MinGW#How_to_build)
* [3 Errors](https://wiki.qt.io/Compiling_OpenSSL_with_MinGW#Errors)
* [4 Using it](https://wiki.qt.io/Compiling_OpenSSL_with_MinGW#Using_it)

Pre-built packages

The [Win32OpenSSL project](http://slproweb.com/products/Win32OpenSSL.html) provides pre-compiled libraries for both 32 and 64 bit. However, these depend on the Microsoft Visual C++ 2008 runtime being installed.

Inside [MSYS2](https://wiki.qt.io/MSYS2) shell, pre-built openssl can be obtained from MSYS2 repo, with this command: pacman -S openssl

Compiling on your own

**Requirements**

* Either use MSYS, or use [MSYS2](https://wiki.qt.io/MSYS2) :
  + MSYS shell command prompt (an sh shell + some UNIX tools, get it from <http://sourceforge.net/projects/mingw/files/MSYS/Base/msys-core/msys-1.0.11/MSYS-1.0.11.exe/download?use_mirror=garr>. It's also an (optional) part of mingw.org installer). Install it in C:drive in "msys" folder.
  + [MSYS2](https://wiki.qt.io/MSYS2) shell command prompt, it can be obtained from <https://msys2.github.io/> Install it in C:drive in "msys2" folder.
* Working MinGW, or MinGW-w64 toolchain:
  + Either get & load MinGW, or, get & load MinGW-w64 toolchains (from their respective project website), or, use "MinGW-w64" toolchains from MSYS2 repo, inside the MSYS2-shell.
  + NOTE: Do not use (or try to avoid using) different type of toolchains for building different sub-components for the same/main project, when targeted for same platform/OS. If you are using MinGW-w64 based toolchains from MSYS2\mingw32 or MSYS2\mingw64, then for main project & all sub-components, use same MSYS2 toolchains. If you are using MinGW toolchain, then for all of your sub-components & main project, use same MinGW toolchain. If you are using MinGW-w64 toolchains from "MinGW-builds", then for all sub-components & main project, use same MinGW-builds toolchains.

**How to build**

* Download latest+stable [OpenSSL](https://www.openssl.org/) (You must use HTTPS secure encrypted connection to obtain source file) from <https://www.openssl.org/source> (e.g. openssl-1.0.1c.tar.gz)
* Either start MSYS-shell, or start MSYS2-shell:
  + Start a MSYS shell command prompt (C:.0\msys.bat)
  + Or, start MSYS2 shell command prompt (C:2\msys2\_shell.bat)
    - See inside this [MSYS2](https://wiki.qt.io/MSYS2) page, how to prepare MSYS2 with build/compile related toolchains, tools & dependencies
* Extract source tar.gz using MSYS tar or MSYS2 tar (and ignore the symlink warnings)
  + *don't use 7zip or other apps, since they fail to set up any symlinks! tar will also complain about symlinks, but compilation will still succeed.*
* Unzip/decompress source:

$ tar xvzf openssl-1.0.1c.tar.gz

$ cd openssl-1.0.1c

* Check that gcc is in PATH, otherwise add it: change below command to match with your-side directory's exact name & letter-case & location:
  + If using MinGW-builds toolchain (which is usually located outside of MSYS or MSYS2), then execute this in MSYS or in MSYS2:

$ export PATH="/c/Mingw-builds/bin:$PATH"

* + If using MSYS shell, and if you have installed MinGW toolchain into C:32 directory, then execute this:

$ export PATH="/c/mingw32/bin:$PATH"

* + If using MSYS2 shell, execute this, if you are building for 32bit:

$ export PATH="/c/msys2/mingw32/bin:$PATH"

* + If using MSYS2 shell, execute this, if you are building for 64bit:

$ export PATH="/c/msys2/mingw64/bin:$PATH"

* for MinGW (32 bit) do:

$ ./Configure --prefix=$PWD/dist no-idea no-mdc2 no-rc5 shared mingw

* for MinGW-w64 do:

$ ./Configure --prefix=$PWD/dist no-idea no-mdc2 no-rc5 shared mingw64

* Compile & install:

$ make depend && make && make install

Errors

* If you run into

perl asm/sha1-x86\_64.pl mingw64 > sha1-x86\_64.s

gcc -I.. -I../.. -I../modes -I../asn1 -I../evp -I../../include -D\_WINDLL -DOPENSSL\_PIC -DOPENSSL\_TH

READS -D\_MT -DDSO\_WIN32 -DL\_ENDIAN -O3 -Wall -DWIN32\_LEAN\_AND\_MEAN -DUNICODE -D\_UNICODE -DOPENSSL\_IA

32\_SSE2 -DOPENSSL\_BN\_ASM\_MONT -DOPENSSL\_BN\_ASM\_MONT5 -DOPENSSL\_BN\_ASM\_GF2m -DSHA1\_ASM -DSHA256\_ASM-

DSHA512\_ASM -DMD5\_ASM -DAES\_ASM -DVPAES\_ASM -DBSAES\_ASM -DWHIRLPOOL\_ASM -DGHASH\_ASM -c -o sha1-x86\_

64.o sha1-x86\_64.s

sha1-x86\_64.s: Assembler messages:

sha1-x86\_64.s:1824: Warning: end of file not at end of a line; newline inserted

sha1-x86\_64.s:2183: Error: number of operands mismatch **for** `rol'

make[2]: '''\* [sha1-x86\_64.o] Error 1

make[2]: Leaving directory `/d/dev/tmp/openssl-1.0.1c/crypto/sha'

make[1]:'''\* [subdirs] Error 1

make[1]: Leaving directory `/d/dev/tmp/openssl-1.0.1c/crypto'

make: '''\* [build\_crypto] Error 1

**check out**[**http://openssl.6102.n7.nabble.com/Compile-error-with-MinGW-w64-td36657.html**](http://openssl.6102.n7.nabble.com/Compile-error-with-MinGW-w64-td36657.html) **\* Use perl inside MSYS or inside MSYS2** **\* or change crypto/perlasm/x86\_64-xlate.pl as mentioned in the e-mail thread.**

Using it

* If you are using cmd.exe (Cmd-shell), then: add the bin, include, lib folders to your compilation environment:

C:\> set PATH=PATH;C:–1.0.1c\dist\bin

C:\> set INCLUDE=INCLUDE;C:–1.0.1c\dist\include

C:\> set LIB=LIB;C:–1.0.1c\dist\lib

* If you are using MSYS2/MSYS-shell, then: add the bin, include, lib folders to your compilation environment:

$ export PATH="$PATH:/c/openssl-1.0.1c/dist/bin"

$ export INCLUDE="$INCLUDE:/c/openssl-1.0.1c/dist/include"

$ export LIB="$LIB:/c/openssl-1.0.1c/dist/lib"

When you now run Qt's configure.exe openssl should be detected, and Qt links against the libraries.