Building Bitcoin Core with Visual Studio

Introduction

Solution and project files to build Bitcoin Core with `msbuild` or Visual Studio can be found in the `build_msvc` directory. The build has been tested with Visual Studio 2019 (building with earlier versions of Visual Studio should not be expected to work).

To build Bitcoin Core from the command-line, it is sufficient to only install the Visual Studio Build Tools component.

Building with Visual Studio is an alternative to the Linux based cross-compiler build.

Prerequisites

To build dependencies (except for Qt), the default approach is to use the vcpkg package manager from Microsoft:

1. Install vcpkg.
2. By default, vcpkg makes both `release` and `debug` builds for each package. To save build time and disk space, one could skip `debug` builds (example uses PowerShell):

```
Add-Content -Path "vcpkg\triplets\x64-windows-static.cmake" -Value "set(VCPKG_BUILD_TYPE rel
```

Qt

To build Bitcoin Core with the GUI, a static build of Qt is required.

1. Download a single ZIP archive of Qt source code from https://download.qt.io/official_releases/qt/ (e.g., `qt-everywhere-opensource-src-5.15.3.zip`), and expand it into a dedicated folder. The following instructions assume that this folder is `C:\dev\qt-source`.
2. Open "x64 Native Tools Command Prompt for VS 2019", and input the following commands:

```
cd C:\dev\qt-source
mkdir build
cd build
..\configure -release -silent -opensource -confirm-license -opengl desktop -static -static-r
nmake
nmake install
```

One could speed up building with `jom`, a replacement for `nmake` which makes use of all CPU cores.

To build Bitcoin Core without Qt, unload or disable the `bitcoin-qt`, `libbitcoin_qt` and `test_bitcoin-qt` projects.

Building

1. Use Python to generate `*.vcxproj` from Makefile:

```
PS >py -3 msvc-autogen.py
```

2. An optional step is to adjust the settings in the `build_msvc` directory and the `common.init.vcxproj` file. This project file contains settings that are common to all projects such as the runtime library version and target Windows SDK version. The Qt directories can also be set. To specify a non-default path to a static Qt package directory, use the `QTBASEDIR` environment variable.

3. To build from the command-line with the Visual Studio 2019 toolchain use:

```
msbuild -property:Configuration=Release -maxCpuCount -verbosity:minimal bitcoin.sln
```

Alternatively, open the `build_msvc/bitcoin.sln` file in Visual Studio 2019.

Security

Base address randomization is used to make Bitcoin Core more secure. When building Bitcoin using the `build_msvc` process base address randomization can be disabled by editing `common.init.vcproj` to change `RandomizedBaseAddress` from `true` to `false` and then rebuilding the project.

To check if `bitcoind` has `RandomizedBaseAddress` enabled or disabled run

```
.\dumpbin.exe /headers src/bitcoind.exe
```

If is it enabled then in the output `Dynamic base` will be listed in the `DLL characteristics` under `OPTIONAL HEADER VALUES` as shown below

```
            8160 DLL characteristics
                   High Entropy Virtual Addresses
                   Dynamic base
                   NX compatible
                   Terminal Server Aware
```

This may not disable all stack randomization as versions of windows employ additional stack randomization protections. These protections must be turned off in the OS configuration.