**Создание статического qt**

ОС Windows 7, 64 bit. MSVC 2017

1. Загружаем Microsoft Visual Studio Community 2017. Устанавливаем компилятор, стандартные библиотеки и заголовочные файлы.
2. В комплекте не будет дебаггера. По сути он не нужен, т.к. мы ориентируемся на версию для выпуска, а не для отладки, но лучше установить отдельно WindowsSDK, в котором будет X64 Debuggers And Tools-x64\_en\_us. (не помню, как качал SDK, у меня набор инсталляторов. Сейчас только ISO – образы в загрузке с оф. сайта MSDN <https://www.microsoft.com/en-us/download/details.aspx?id=8442>, если загрузить этот образ, смонтировать его на виртуальный привод и перейти в директорию Setup, то можно найти WinSDKDebuggingTools\_ia64 и WinSDKDebuggingTools\_amd64, установить тот, что подходит для процессора(у меня заработал 2-й вариант). По умолчанию директория для установки C:\Program Files\Debugging Tools for Windows (x64) )
3. Понадобится Qt Creator 64 бит. Скачивается отдельно с <https://www.qt.io/offline-installers>. Раздел QtCreator. Далее находим 64 – битную версию и производим установку.
4. Загружаем исходники qt: <https://download.qt.io/archive/qt/5.13/5.13.1/single/>. Я выбрал версию последнюю на 14.09.2019 версию. Скачиваем архив: [qt-everywhere-src-5.13.1.zip](https://download.qt.io/archive/qt/5.13/5.13.1/single/qt-everywhere-src-5.13.1.zip) и распаковываем его в нужное место.
5. После чего нужно открыть Командную строку Native Tools x64 для VS 2017. Заходим в **пуск->все программы->Visual Studio 2017->Visual Studio Tools->VC-> Командная строка Native Tools x64 для VS 2017**.
6. Переходим в директорию, где располагаются распакованные файлы Qt:

**cd G:\QtStatic\qt-everywhere-src-5.13.1**

После этой команды смена диска не произойдет, поэтому нужно еще раз выбрать диск, на который хотим перейти:

**G:**

Как только рабочей директорией будет G:\QtStatic\qt-everywhere-src-5.13.1, можем выполнить конфигурацию qt:

**configure.bat –release –static –opengl desktop –platform win32-msvc –prefix “G:\QtStatic\qt-static” –skip webengine –skip qtdeclarative –nomake tests –nomake examples**

В строке выше –prefix отвчеает за то место, куда будет происходить установка qt. –release говорит о том, что будет только релизная версия. Параметр –skip позволяет пропустить установку определенных модулей. webengine мне был просто не нужен (не знаю, устанавливается или нет), а qtdeclarative не смог установиться, поэтому я его исключил. Подобным образом можно поступать со всеми модулями, которые по каким-либо причинам не устанавливаются.

*Upd: 14\_02\_2020: для установки qtdeclarative нужно иметь установленный python, также лучше установить perl, в противном случае не выполнить nmake –clean. Если при установке что-то пошло не так, лучше удалить весь каталог, залить новый из архива qt-everywhere-src-5.13.1.rar, сконфигурировать заново и запустить nmake.*

Далее нужно запустить команду **nmake**. Как только сборка будет завершена, запускаем **nmake install**.

Процедура сборки и установки длительная. В случае успешного завершения (не выдано никаких записей, содержащих Error) директория G:\QtStatic\qt-static должна заполнится.

1. Теперь нужно настроить QtCreator.
   1. Заходим в **Инструменты->параметры**. Выбираем **Комплекты**. Далее идем на вкладку **Профили Qt.** Добавляем новый профиль. Добавляем новый профиль. Нажимаем **Добавить** и находим qmake.exe, располагающийся в директории bin установленного qt. Например, G:\QtStatic\qt-static\bin
   2. Далее нужно задать компиляторы. В случае установки QtCreator после Visual Studio, компиляторы должны обнаружится самостоятельно. Если нет, то добавляем вручную. Не знаю, где именно располагаются компиляторы, возможно в C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\VC\Tools\MSVC\14.16.27023\bin\Hostx64\x64. Если не получится найти, можно удалить QtCreator, а затем поставить заново (уже после Visual Studio Community), тогда компиляторы будут обнаружены автоматически.
   3. Следом можно задать отладчик (тот, который из WindowsSDK) во вкладке **Отладчики.** Нажимаем **Добавить**, далее указываем имя отладчика и путь к нему. У меня C:\Program Files\Debugging Tools for Windows (x64)\cdb.exe
   4. Переходим во вкладку **Комплекты**. Нажимаем **Добавить**, далее задаем **название**, выбираем **тип устройства** Desktop**, компилятор** для С и для С++ (для обоих задал Microsoft Visual C++ Compiler 15.0 (amd64)). Можно указать **отладчик**. Следует выбрать **Профиль Qt.** Если никаких ошибок нет (иконка рядом с названием комплекта в общем списке комплектов не содержит предупреждений или ошибок, а предупреждение будет в случае отсутствия заданного отладчика), то можно выбрать рабочий профиль, произвести настройку директории версии для выпуска (в левой панели вкладка **Проекты**, далее выбрать нужный профиль в **Сборке** и в **Запуске** выбрать нужную директорию). После чего в дереве проекта нажать правой кнопкой мыши на название проекта, выбрать **запустить qmake**, если ошибок не будет, то следует собрать проект. Если этот шаг завершится успешно, то в директории для выпуска будет располагаться исполняемый файл, содержащий в себе все необходимые для работы библиотеки.