# Установка Qt Creator и требуемых для сборки компонентов Visual Studio

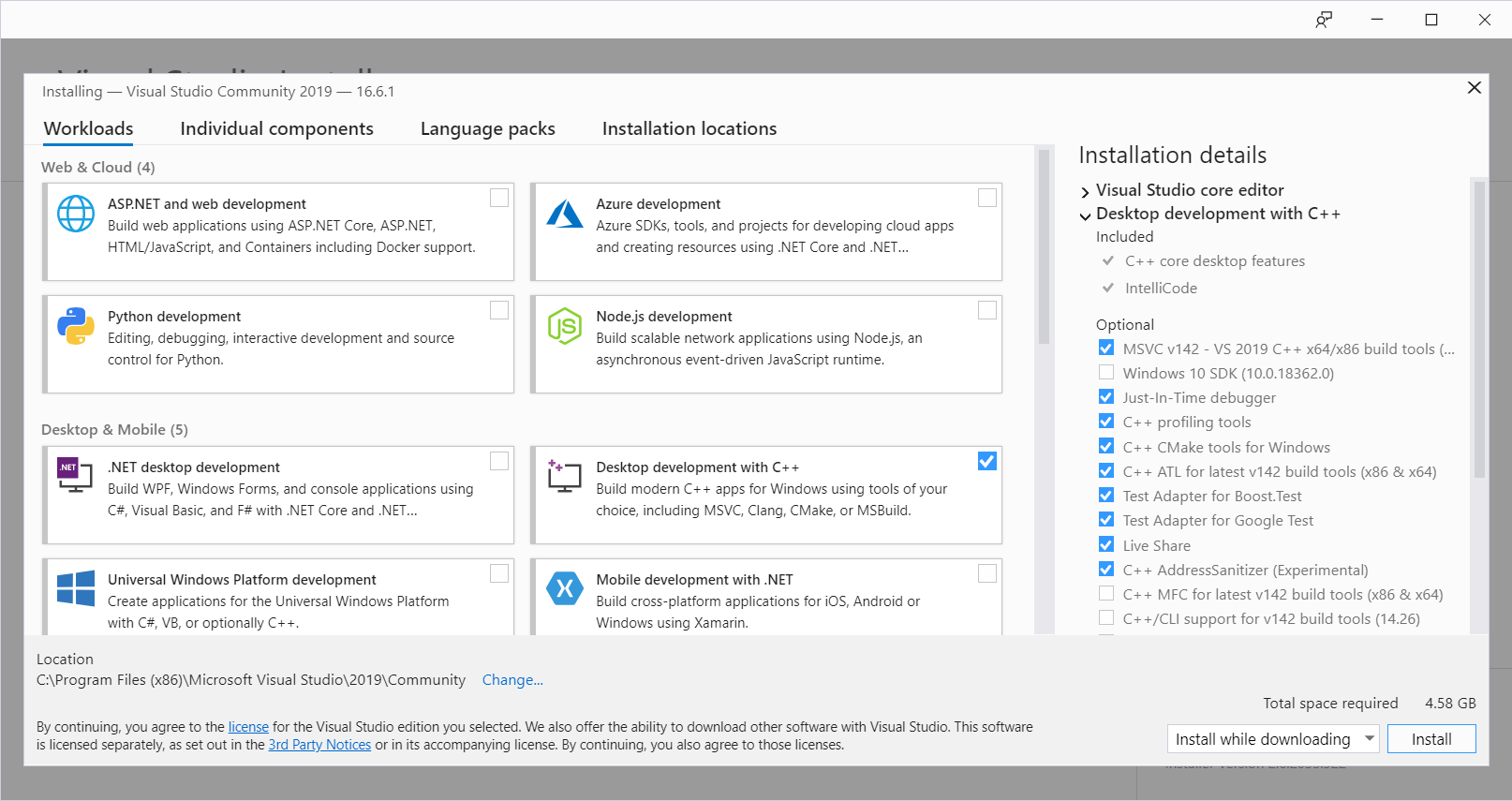
## Версия Windows

Ставил под: Microsoft Windows [Version 10.0.19041.208], сейчас обновил до: Microsoft Windows [Version 10.0.19041.630]

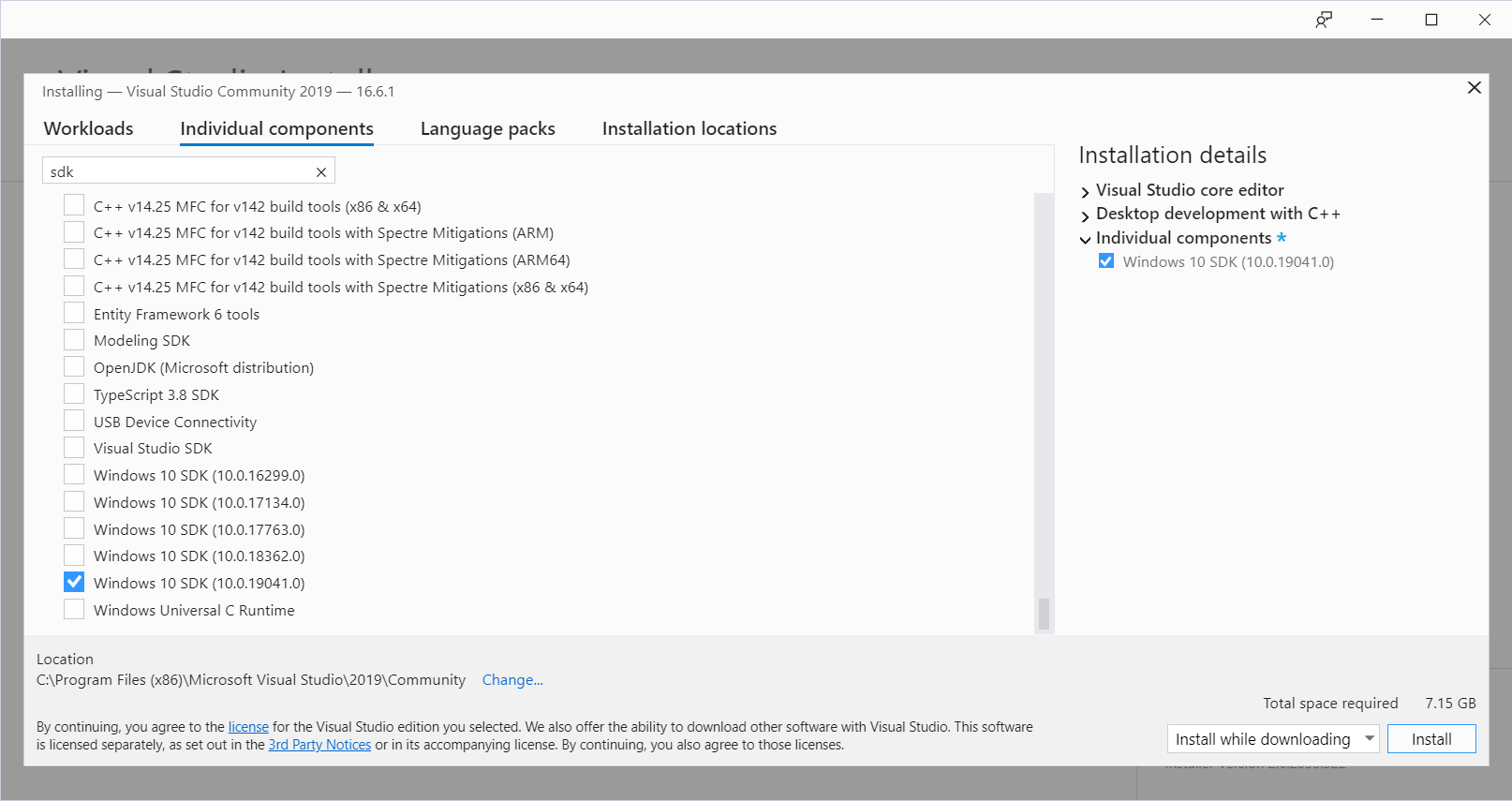
## Установка Visual Studio Community 2019

Установил: Visual Studio Community 2019 16.6.1

При установке выбрал в Workloads: **Desktop development with C++**



Выбрал: **Windows SDK 10.0.19041.0**



## Установка Qt 5.9.5

Скачал Qt 5.9.5 с:

<https://download.qt.io/official_releases/qt/5.9/5.9.5/>

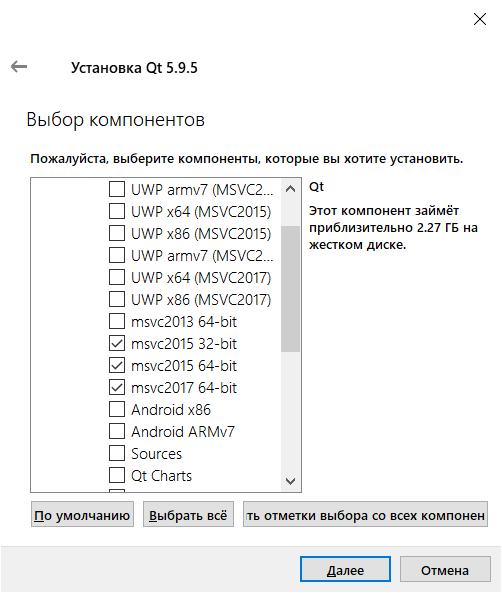
А именно 2 файла:

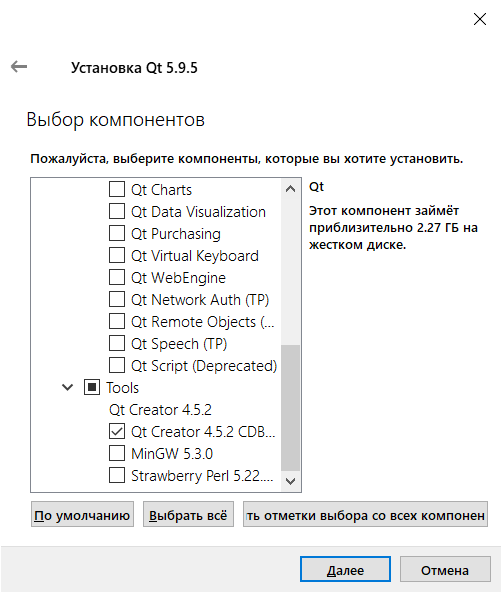
<https://download.qt.io/official_releases/qt/5.9/5.9.5/qt-opensource-windows-x86-5.9.5.exe>

<https://download.qt.io/official_releases/qt/5.9/5.9.5/qt-opensource-windows-x86-pdb-files-desktop-5.9.5.zip>

Установил Qt 5.9.5 со следующими параметрами:

* msvc2015 32-bit
* msvc2015 64-bit
* msvc2017 64-bit



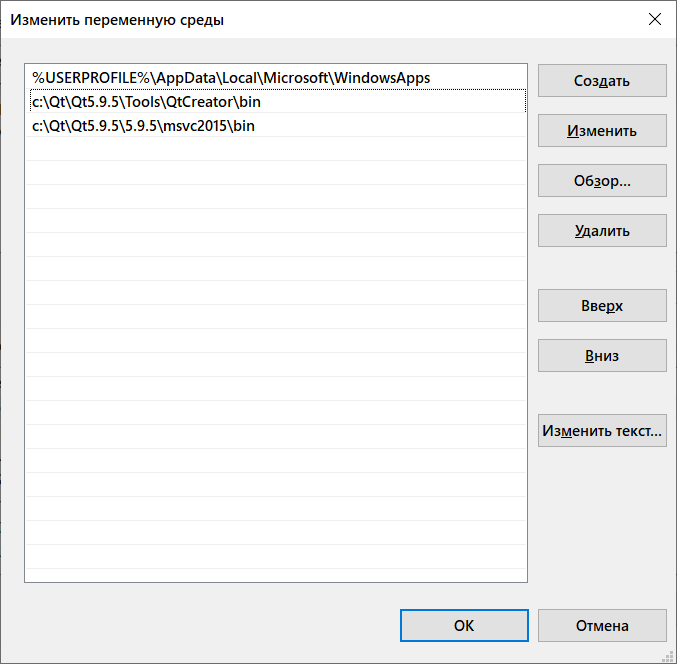


## Задание переменной окружения PATH

Добавил в переменную окружения PATH пользователя каталоги:

c:\Qt\Qt5.9.5\Tools\QtCreator\bin

c:\Qt\Qt5.9.5\5.9.5\msvc2015\bin



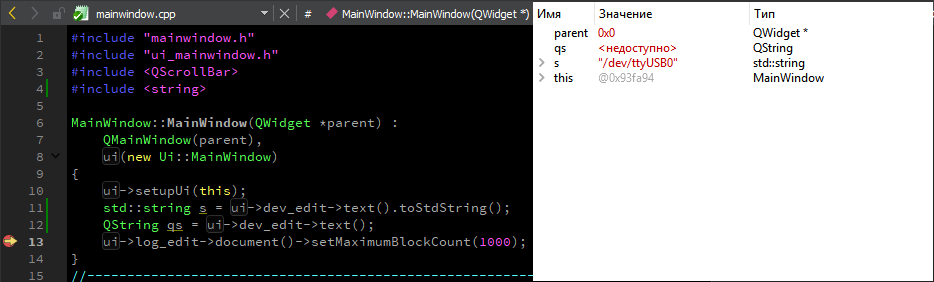
## Добавление PDB-файлов

В настоящее время PDB-файлы, необходимые для отладки, предоставляются в отдельном zip-файле.

Этот zip-файл содержит файлы 7zip, которые, в свою очередь, должны быть извлечены в правильную подпапку установки Qt.

Без PDB-файлов, например, не отображаются значения композитных переменных в отладчике, например QString.

В окне: "Locals and Expressions" в значениях данных переменных выводится: <недоступно> или <not accessible>



При установке последней версии Qt, через online installer, можно выбрать: debug info.

Подробнее можно посмотреть QTBUG-67587 с резолюцией - Won't Do:

**PDB files of windows online installer are provided in a unmanageable huge compressed folder**

<https://bugreports.qt.io/browse/QTBUG-67587>

Ну и почитать, как к этому относятся другие разработчики:

**some Qt types in debugger shown as 'not accessible'**

<https://bugreports.qt.io/browse/QTCREATORBUG-19797>

[Permalink](https://bugreports.qt.io/browse/QTCREATORBUG-19797?focusedCommentId=427681&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel" \l "comment-427681" \o "Right click and copy link for a permanent link to this comment.)

[David Schulz](https://bugreports.qt.io/secure/ViewProfile.jspa?name=davschul) added a comment - 19 Oct '18 09:05

Unfortunately the pdb files aren't part of the offline packages anymore. Feel free to leave a vote on  if this causes trouble for you.

[Permalink](https://bugreports.qt.io/browse/QTCREATORBUG-19797?focusedCommentId=429087&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel" \l "comment-429087" \o "Right click and copy link for a permanent link to this comment.)

[Permalink](https://bugreports.qt.io/browse/QTCREATORBUG-19797?focusedCommentId=434219&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel" \l "comment-434219" \o "Right click and copy link for a permanent link to this comment.)

[David Schulz](https://bugreports.qt.io/secure/ViewProfile.jspa?name=davschul) added a comment - 22 Nov '18 06:44

Feel free to leave a vote on  if this causes trouble for you. I also think this is more than uncomfortable.

Т.к. мы устанавливали Qt из offline installer, надо вручную скопировать данные файлы.

Разархивировал, скачанный ранее zip файл с pdb файлами:

<http://download.qt.io/official_releases/qt/5.9/5.9.5/qt-opensource-windows-x86-pdb-files-desktop-5.9.5.zip>

Для MSVC2015 X86\_64 надо выполнить следующее:

Отобрал .7z файлы по маске:

\*Windows-Windows\_10-MSVC2015-Windows-Windows\_10-X86\_64\*.\*

Разархивировал в один каталог с помощью Total Commander.

Выделил файлы.

Выполнил команду: Файлы -> Распаковать … (Alt+F9)

Скопировал в:

c:\Qt\Qt5.9.5\5.9.5\msvc2015\_64\

Аналогично, для 32-разрядной версии надо скопировать соответствующие PDB-файлы в: c:\Qt\Qt5.9.5\5.9.5\msvc2015

## WebKit

В Qt 5.6.0 WebKit был удален.

<https://github.com/goldendict/goldendict#building-under-windows-with-ms-visual-studio>

**Building under Windows with MS Visual Studio**

Note: In Qt 5.6.0 and later the Webkit module was removed from official release builds. You should to build it from sources to compile GoldenDict.

Собирать из исходников я не стал.  
На: <https://github.com/qtwebkit/qtwebkit/releases>, существуют скомпилированные qtwebkit только под версии:

Qt 5.14.1, Qt 5.13, Qt 5.9

Скачал скомпилированный QtWebKit под Qt 5.9:

QtWebKit 5.212.0 Alpha 2

<https://github.com/qtwebkit/qtwebkit/releases/tag/qtwebkit-5.212.0-alpha2>

<https://github.com/qtwebkit/qtwebkit/releases/download/qtwebkit-5.212.0-alpha2/qtwebkit-5.212.0_alpha2-qt59-msvc2015-x86.zip>

<https://github.com/qtwebkit/qtwebkit/releases/download/qtwebkit-5.212.0-alpha2/qtwebkit-5.212.0_alpha2-qt59-msvc2015-x64.zip>

MinGW есть только 32-bit:

<https://github.com/qtwebkit/qtwebkit/releases/download/qtwebkit-5.212.0-alpha2/qtwebkit-5.212.0_alpha2-qt59-mingw530-x86.zip>

Развернул архив в папку:

c:\Qt\Qt5.9.5\5.9.5\msvc2015

c:\Qt\Qt5.9.5\5.9.5\msvc2015\_64

c:\Qt\Qt5.9.5\5.9.5\mingw53\_32

## Запуск Qt Creator

Перед запуском Qt Creator надо задать переменные окружения для VS 2015.

Запустить командную строку через ярлык:

VS2015 x64 Native Tools Command Prompt

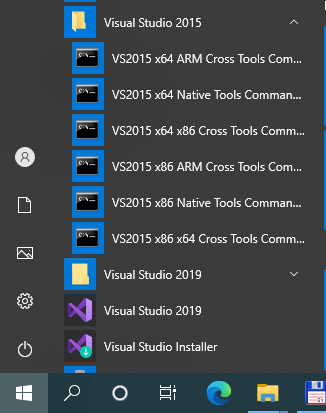
или

VS2015 x86 Native Tools Command Prompt, соответственно для 32bit,

которые соответственно выполнят команды:

%comspec% /k ""C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\vcvarsall.bat"" amd64

%comspec% /k ""C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\vcvarsall.bat"" x86



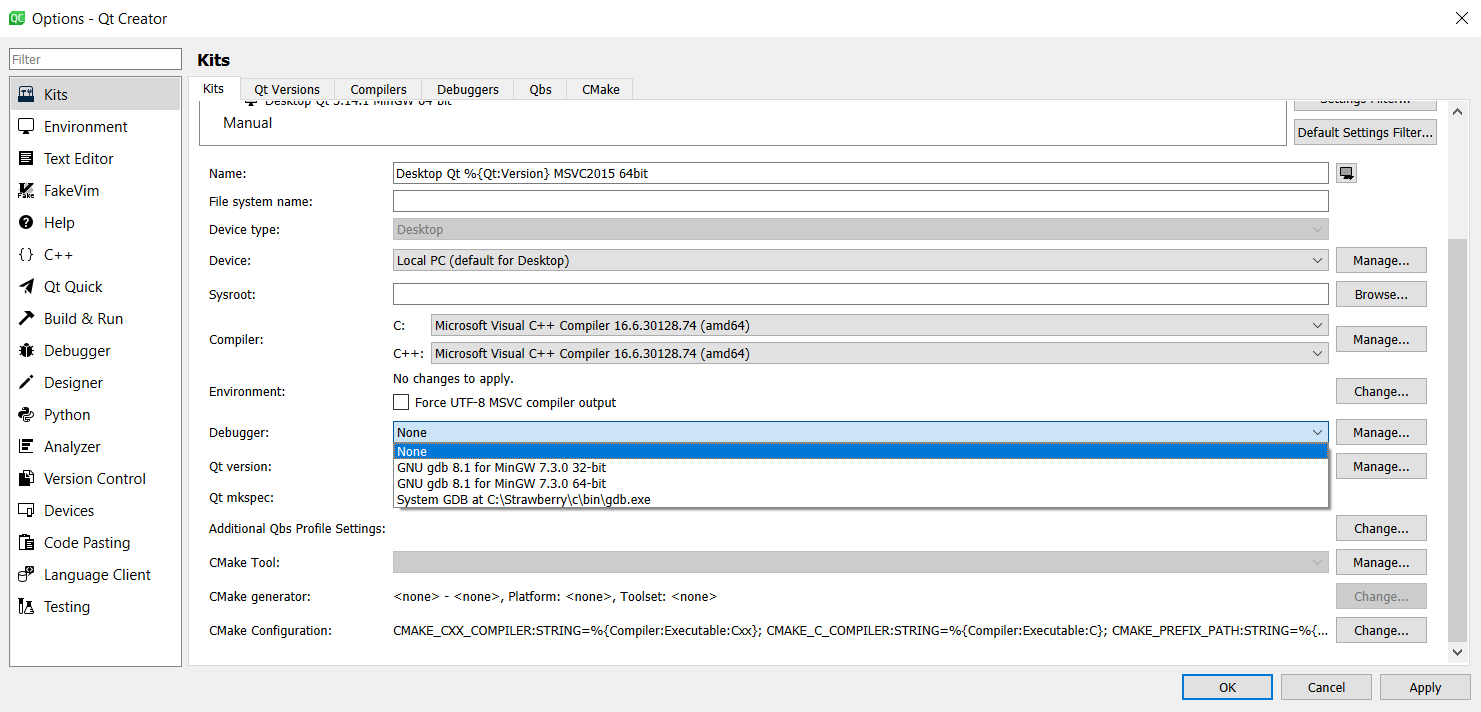
Затем в открывшейся командной строке запустить Qt Creator:

C:\Qt\Qt5.9.5\Tools\QtCreator\bin\qtcreator.exe

## Отладчик cdb.exe недоступен

Необходимо задать поля для Компилятор (Compiler) и Отладчик (Debugger).

Запустил Qt Creator, но Отладчик (Debugger) cdb.exe оказался недоступен.



## Windows 10 SDK

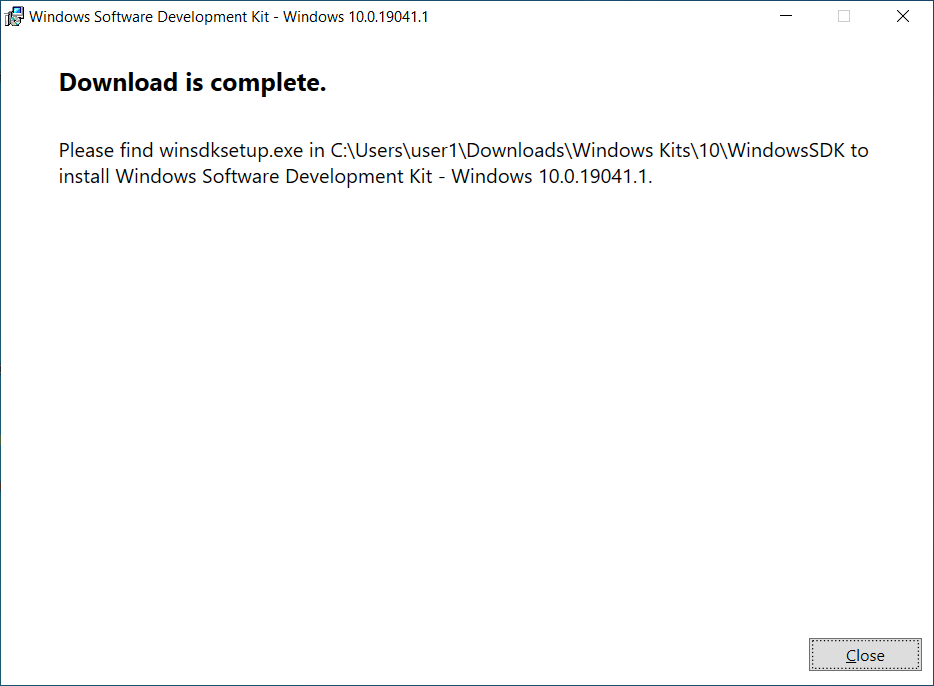
Скачал Windows 10 SDK (10.0.19041.0)

<https://developer.microsoft.com/ru-ru/windows/downloads/windows-10-sdk/>

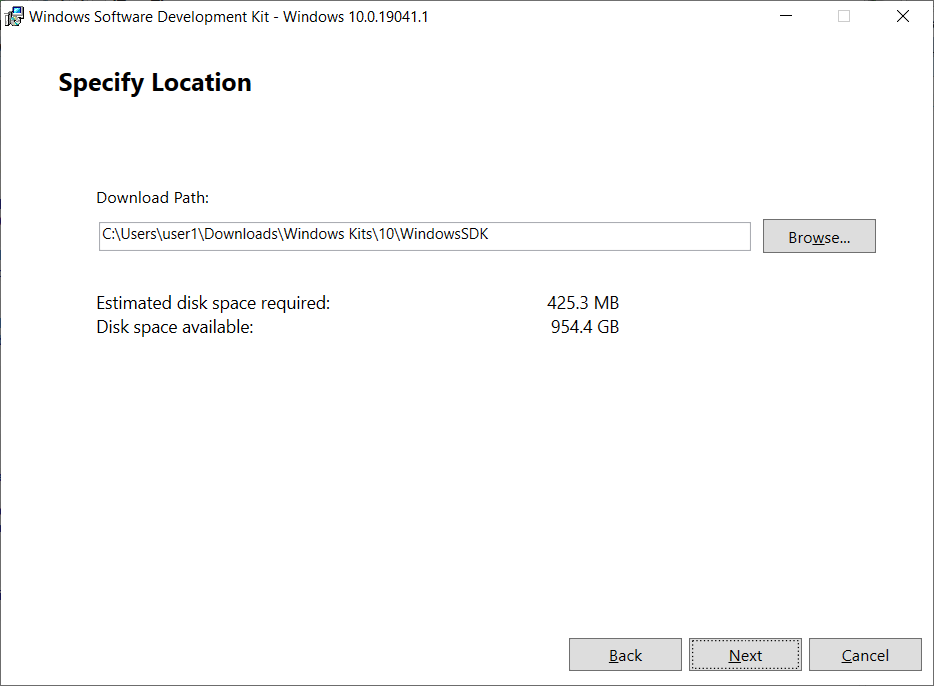
Установил только отладчики, из 2-х msi-файлов:

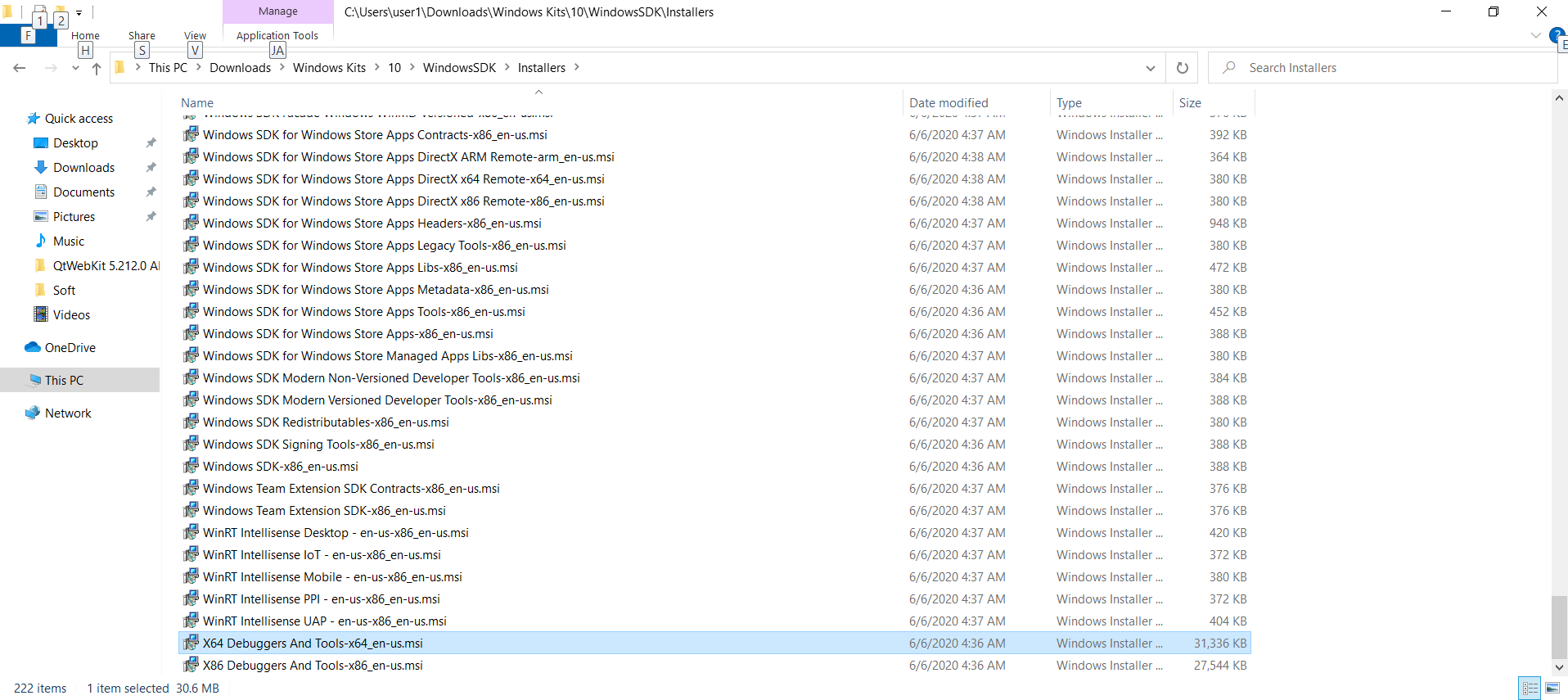
X64 Debuggers And Tools-x64\_en-us.msi

X86 Debuggers And Tools-x86\_en-us.msi



Каталог: C:\Users\user1\Downloads\Windows Kits\10\WindowsSDK

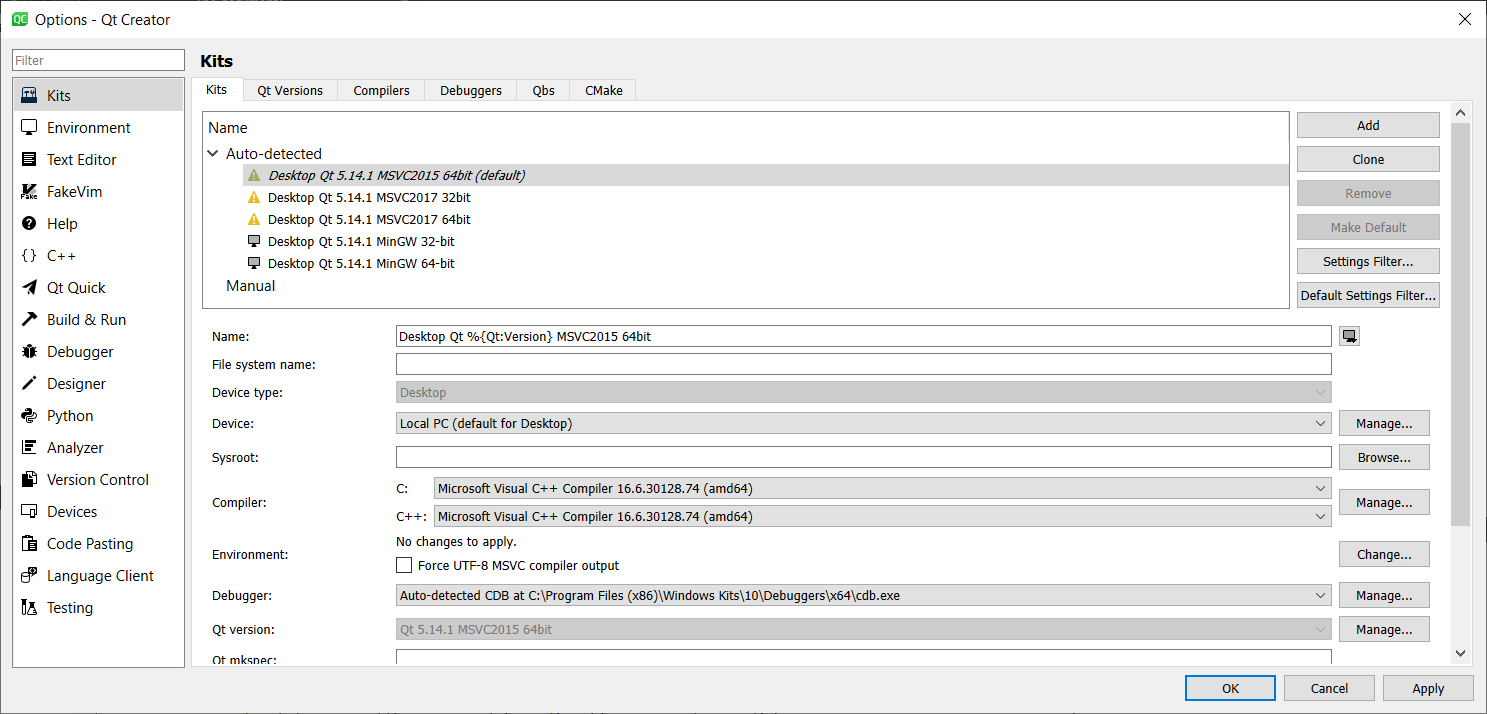




Установил:

X64 Debuggers And Tools-x64\_en-us.msi

Отладчик появился:



Подробнее:

[https://doc.qt.io/qtcreator/creator-debugger-engines.html#debugging-tools-for-windows](https://doc.qt.io/qtcreator/creator-debugger-engines.html" \l "debugging-tools-for-windows)

### Debugging Tools for Windows

To use the CDB debugger, you must install the *Debugging tools for Windows*. You can download them from

[Download and Install Debugging Tools for Windows](https://developer.microsoft.com/windows/downloads/windows-10-sdk) as part of the Windows SDK.

**Note:**Visual Studio does not include the Debugging tools needed, and therefore, you must install them separately.

In addition, you must select **Qt Creator CDB Debugger Support** (in **Qt** > **Tools** > **Qt Creator**) when you install Qt or the stand-alone Qt Creator.

### Добавление Symbol Server provided by Microsoft в symbol search path for Windows debuggers

Необязательный пункт.

Чтобы получить отладочную информацию для библиотек операционной системы, для отладки приложений Windows,

добавьте сервер символов, предоставленный корпорацией Майкрософт, в путь поиска символов отладчика:

В результате чтения

<https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/symbol-path>

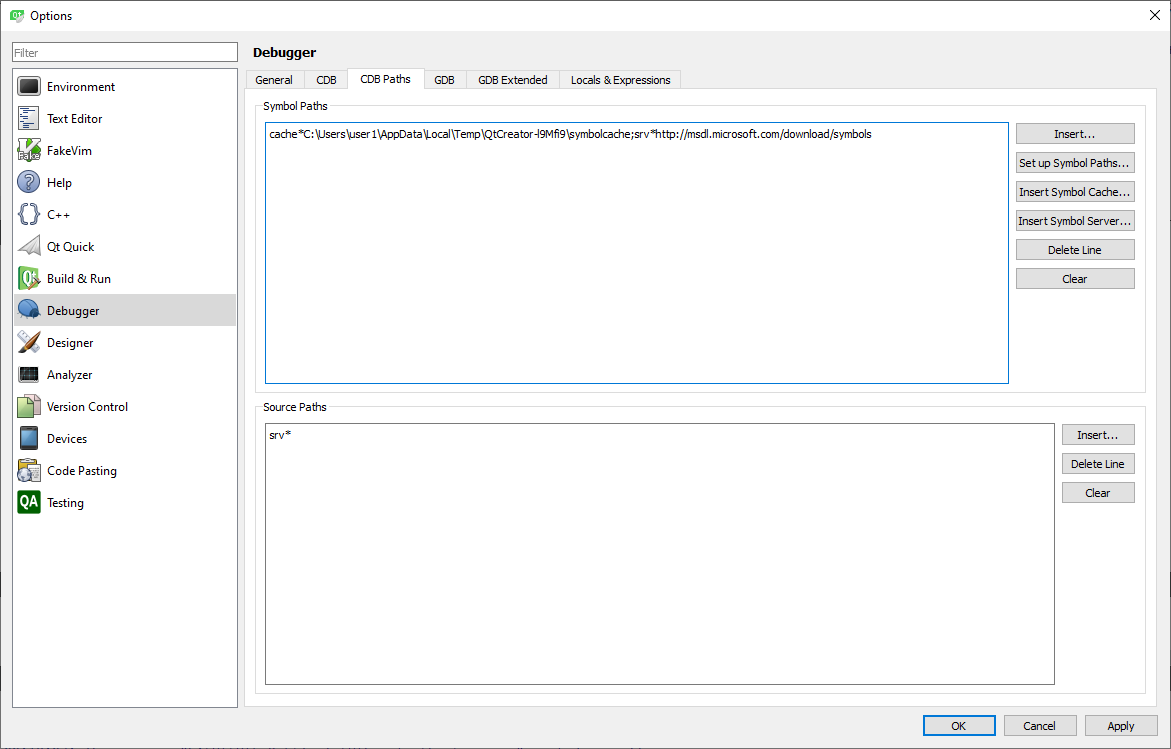
задал:

**cache\*C:\Users\user1\AppData\Local\Temp\QtCreator-l9Mfi9\symbolcache;srv\*http://msdl.microsoft.com/download/symbols**

Как в примере:

dbgcmdCopy

.sympath cache\*c:\MySymbols;srv\*https://msdl.microsoft.com/download/symbols



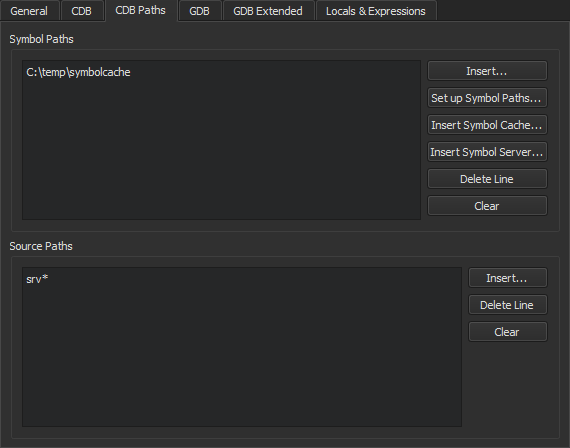
Подробнее:

[https://doc.qt.io/qtcreator/creator-debugger-engines.html#setting-cdb-paths-on-windows](https://doc.qt.io/qtcreator/creator-debugger-engines.html" \l "setting-cdb-paths-on-windows)

## Setting CDB Paths on Windows

To obtain debugging information for the operating system libraries for debugging Windows applications, add the Symbol Server provided by Microsoft to the symbol search path of the debugger:

1. Select **Tools** > **Options** > **Debugger** > **CDB Paths**.



1. In the **Symbol Paths** group, select **Insert**.
2. Select the directory where you want to store the cached information.

Use a subfolder in a temporary directory, such as C:\temp\symbolcache.

1. Select **OK**.

**Note:**Populating the cache might take a long time on a slow network connection.

To use the Source Server infrastructure for fetching missing source files directly from version control or the web, enter the following string in the **Source Paths** field: srv\*.

## Git

Установил Git

<https://git-scm.com/download/win>