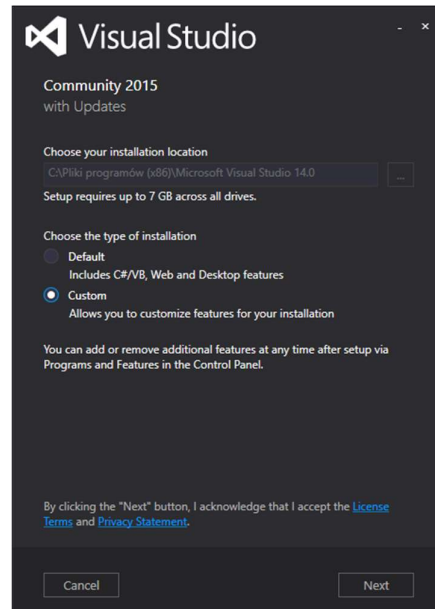


INSTALACJA VISUAL STUDIO, Qt I VTK

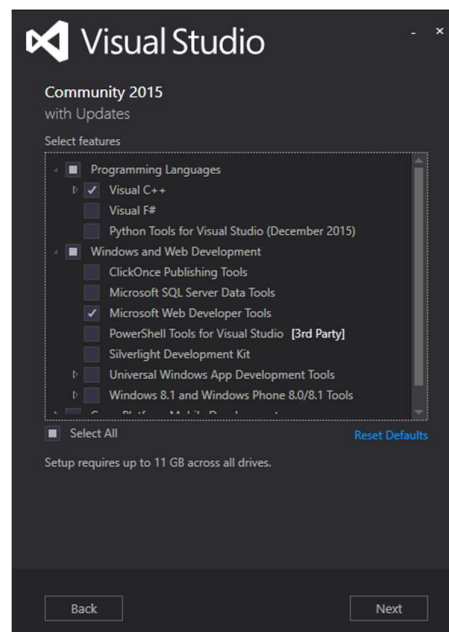
Zainstaluj Microsoft Visual Studio 2015. Możesz pobrać je stąd:

<https://www.visualstudio.com/pl/vs/older-downloads/>

Wybierz typ instalacji Custom:



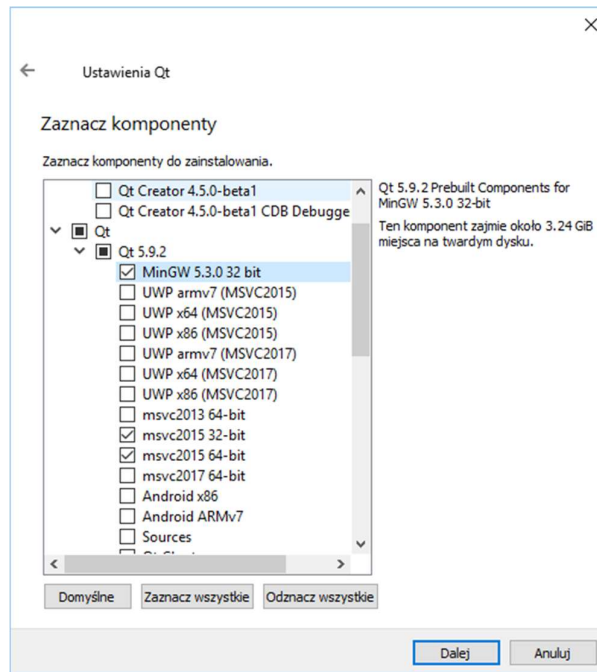
I zaznacz język programowania Visual C++:



Zainstaluj Qt5. Możesz pobrać je stąd:

<https://www1.qt.io/download-open-source/?hsCtaTracking=f977210e-de67-475f-a32b-65cec207fd03%7Cd62710cd-e1db-46aa-8d4d-2f1c1ffdacea>

Wybierz Qt5.9.2 i kompilatory msvc 2015:



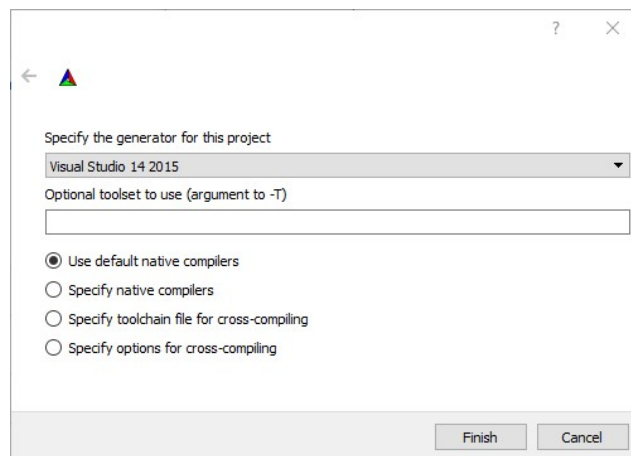
Pobierz VTK z repozytorium:

```
git clone git://vtk.org/VTK.git
```

Umieść pliki do katalogu np. C:\VTK\VTK.

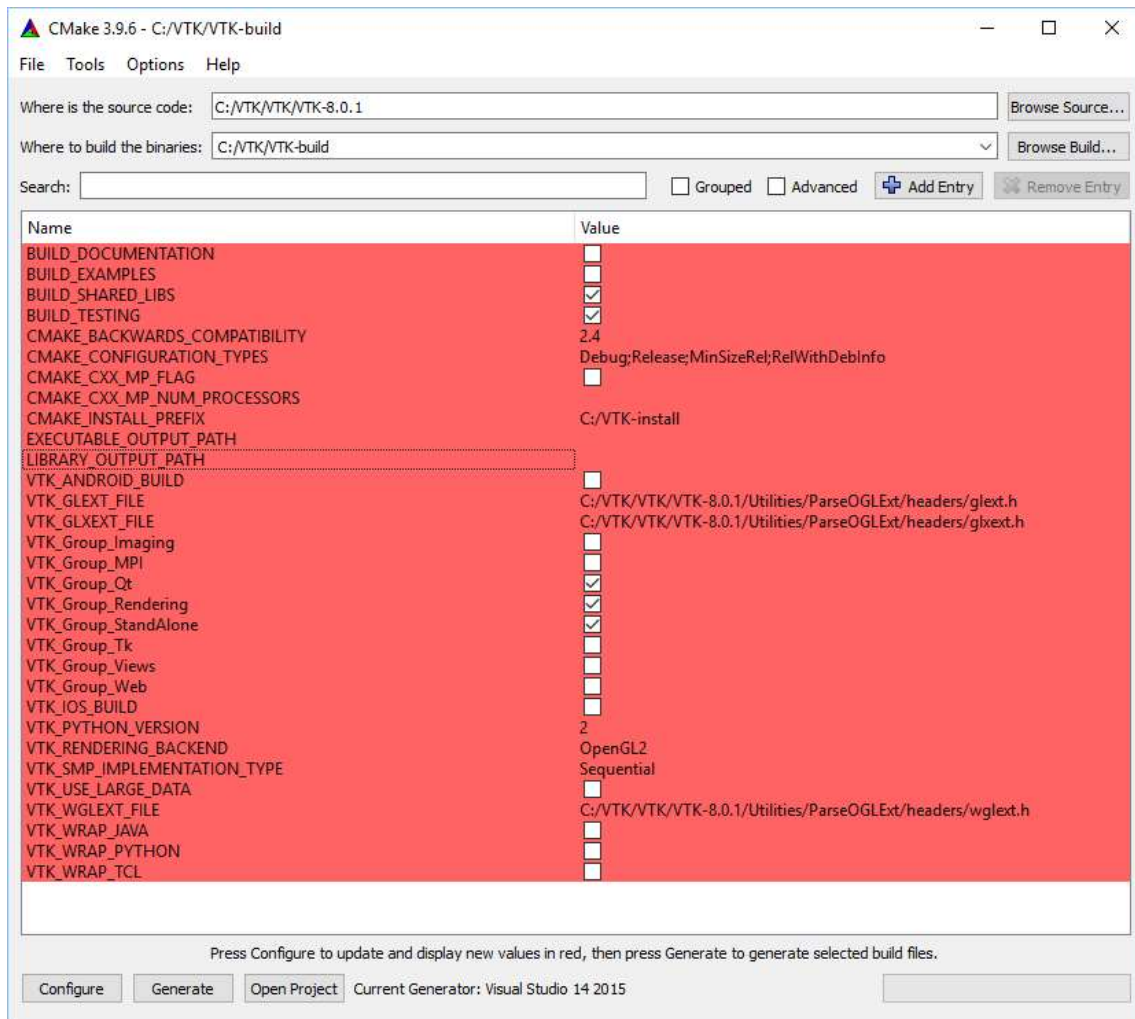
Pobierz program Cmake: <https://cmake.org/download/> (wersja 3.9.6) i go zainstaluj.

Uruchom program Cmake. W pasku *Where is the source code* podaj ścieżkę do plików VTK (C:/VTK/VTK), a w pasku *Where to build the binaries* podaj ścieżkę, gdzie zostaną utworzone pliki do instalacji VTK (np. C:/VTK/VTK-build). Naciśnij przycisk *Configure* i wybierz generator Visual Studio 14 2015:

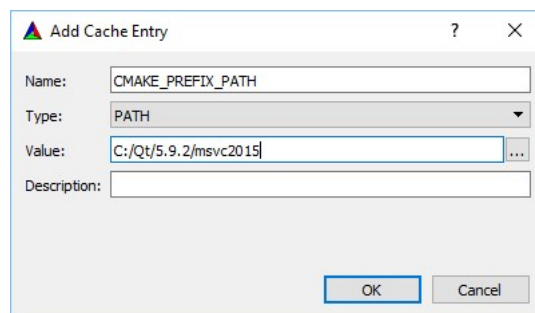


Naciśnij przycisk Finish.

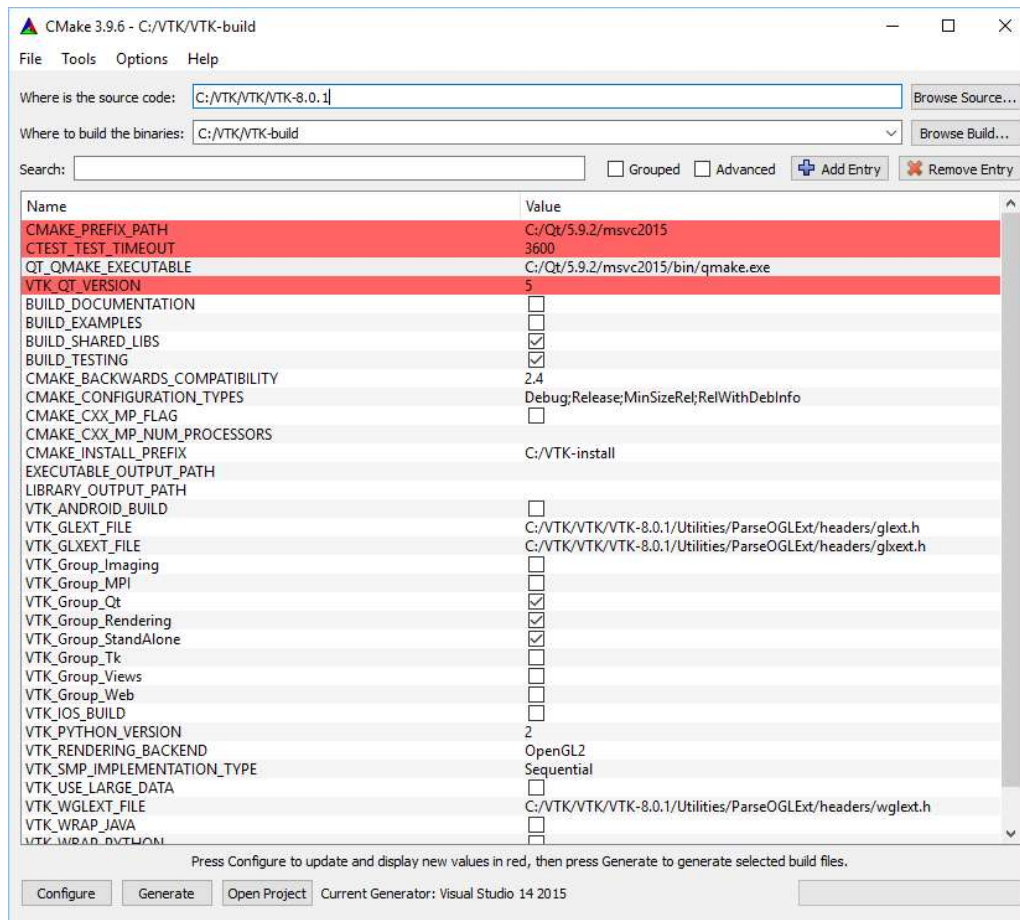
Skonfiguruj VTK. Zaznacz BUILD_SHARED_LIBS, BUILD_TESTING, VTK_Group_Qt. CMAKE_INSTALL_PREFIX zmień na folder, gdzie masz prawa do zapisywania (np. C:/VTK-install) (zauważ, że w Cmake w ścieżkach używa się '/' zamiast '\') i naciśnij przycisk *Configure*:



Zmień VTK_QT_VERSION:5, QT_QMAKE_EXECUTABLE: ścieżka do pliku qmake.exe (np. C:/Qt/5.9.2/msvc2015/bin/qmake.exe). Naciśnij przycisk *Add Entry* aby utworzyć zmienną CMAKE_PREFIX_PATH i podaj ścieżkę do kompilatora (np. C:/Qt/5.9.2/msvc2015) i naciśnij przycisk *Configure*:



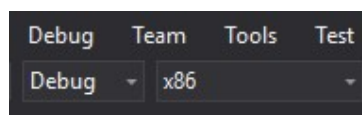
Jeżeli natomiast brakuje Qt5_DIR, podaj ścieżkę do pliku Qt5Config.cmake (np. C:/Qt/5.9.2/msvc2015/lib/cmake/Qt5).



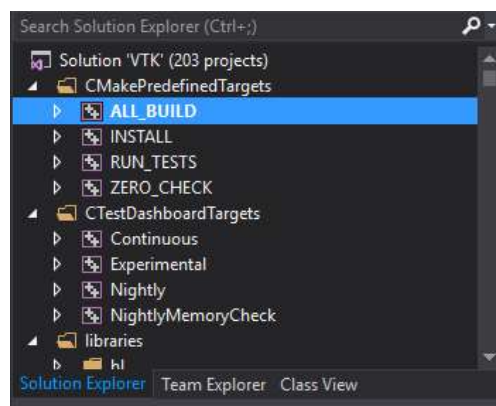
Po ukończonej konfiguracji, naciśnij przycisk *Generate*.

Uruchom Visual Studio. Wybierz File->Open->Project/Solution... i otwórz wygenerowaną solucję VTK.sln (C:\VTK\VTK-build\VTK.sln).

Ustal tryb Debug (w tym trybie też uruchamiaj aplikacje):



Naciśnij prawym przyciskiem myszy na ALL_BUILD i wybierz Build (ten proces może zająć dużo czasu):



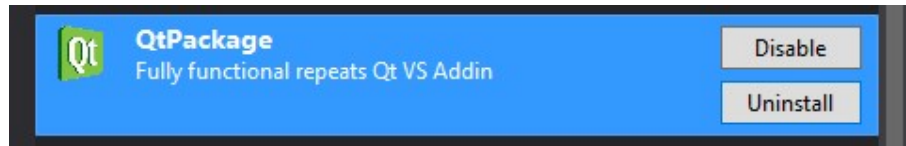
Następnie, po ukończonym budowaniu, naciśnij prawym przyciskiem myszy na INSTALL, następnie wybierz Project Only -> Build Only INSTALL. Gratulacje, właśnie zainstalowałeś VTK 😊

KONFIGURACJA PROJEKTU Qt Z VTK W VISUAL STUDIO

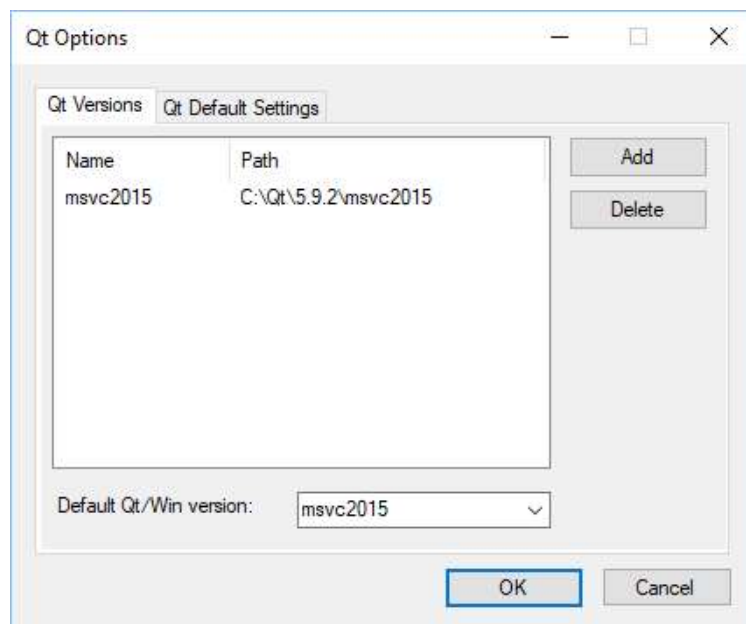
Poniższy filmik przedstawia przebieg instalacji pluginu Qt w Visual Studio oraz tworzenia nowego projektu (od 0:40):

<https://www.youtube.com/watch?v=P6Mg8FpFPS8>

Jedyna różnica jest taka, że rozszerzenie nazywa się QtPackage:



Oraz, że korzystamy z kompilatora msvc2015:



Dodaj pliki dll VTK znajdujące się w folderze z instalacją (np. C:\VTK-install\bin) do zmiennej środowiskowej PATH:

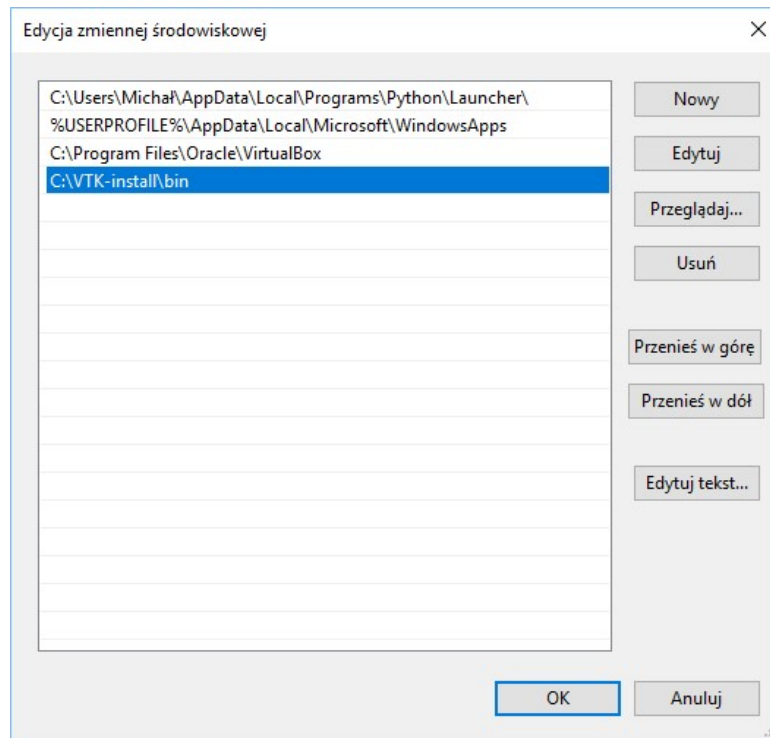
Windows 10 i Windows 8:

- W polu wyszukiwania wpisz „Panel sterowania”, po czym wybierz „System”.
- Kliknij na łączu **Zaawansowane ustawienia systemu**.
- Naciśnij przycisk **Zmienne środowiskowe**. W obszarze **Zmienne systemowe** odszukaj zmienną środowiskową Path, po czym ją wybierz. Naciśnij przycisk **Edytuj**. Jeśli zmienna środowiskowa Path nie istnieje, naciśnij przycisk **Nowa**.
- Dodaj ścieżkę do folderu z plikami dll (np. C:\VTK-install\bin) i naciśnij przycisk **OK**. Zamknij wszystkie pozostałe okna naciskając przycisk **OK**.

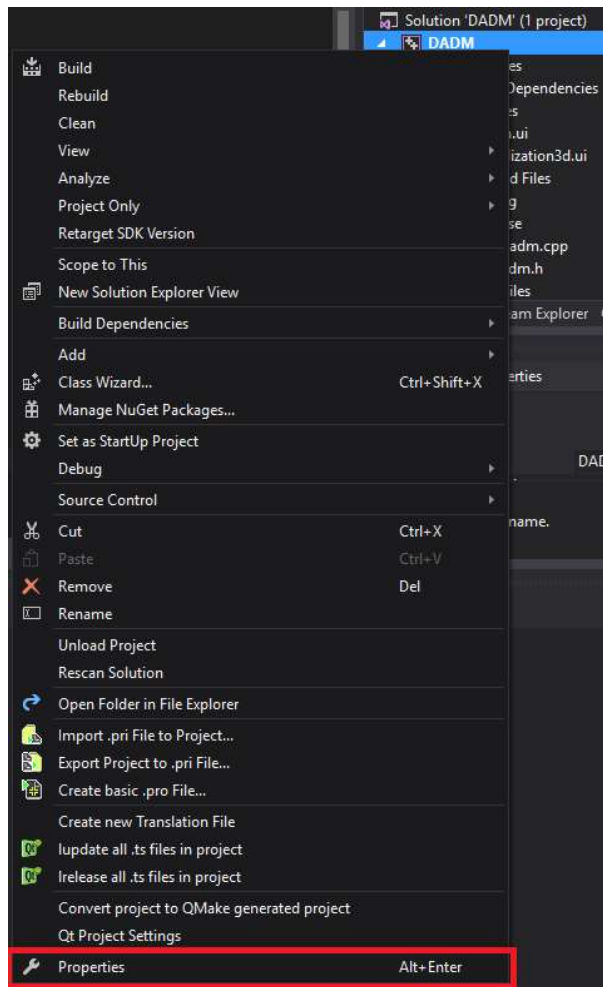
Windows 7

- Kliknij prawym przyciskiem myszy na wyświetlanej na pulpicie ikonie **Komputer**.

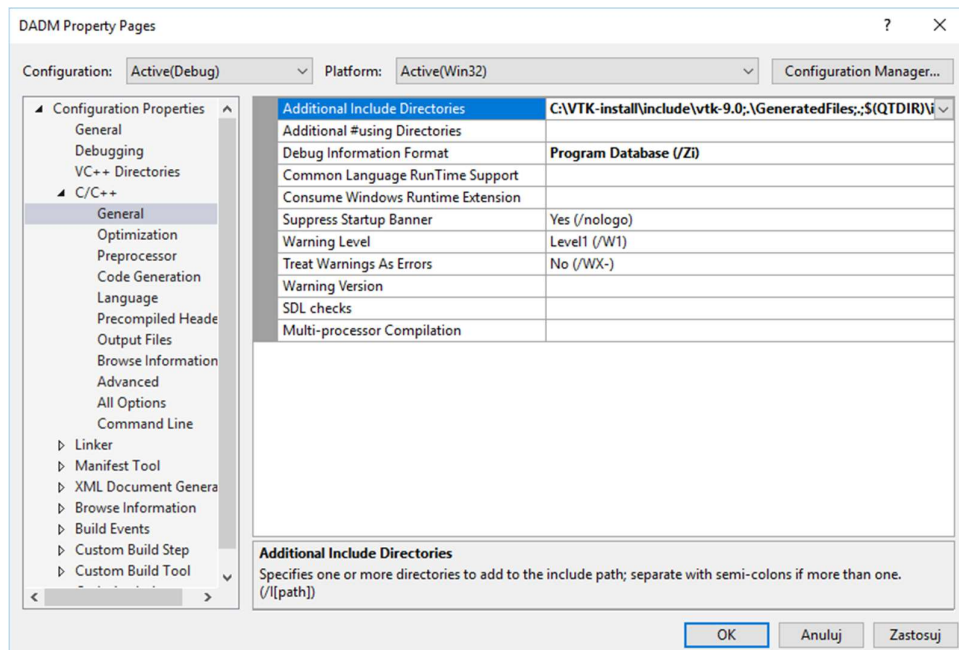
- Z menu podręcznego wybierz opcję **Właściwości**.
- Kliknij na łączu **Zaawansowane ustawienia systemu**.
- Naciśnij przycisk **Zmienne środowiskowe**. W obszarze **Zmienne systemowe** odszukaj zmienną środowiskową Path, po czym ją wybierz. Naciśnij przycisk **Edytuj**. Jeśli zmienna środowiskowa Path nie istnieje, naciśnij przycisk **Nowa**.
- Dodaj ścieżkę do folderu z plikami dll (np. C:\VTK-install\bin) i naciśnij przycisk **OK**. Zamknij wszystkie pozostałe okna naciskając przycisk **OK**.



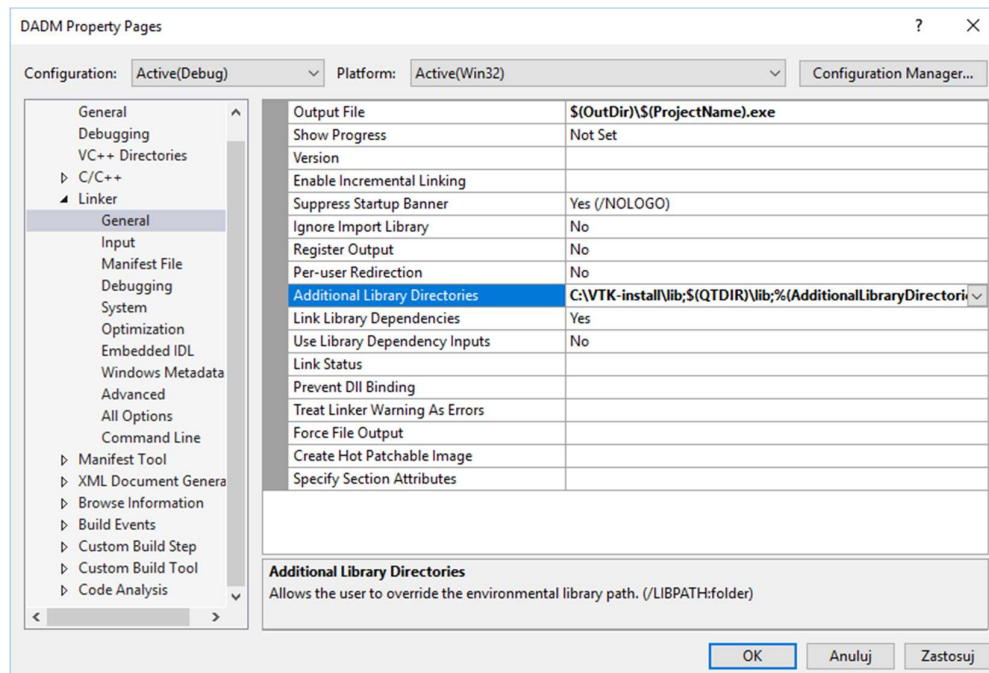
Aby dodać biblioteki VTK do projektu Qt, kliknij prawym przyciskiem myszy na nazwę projektu i wybierz *Properties*:



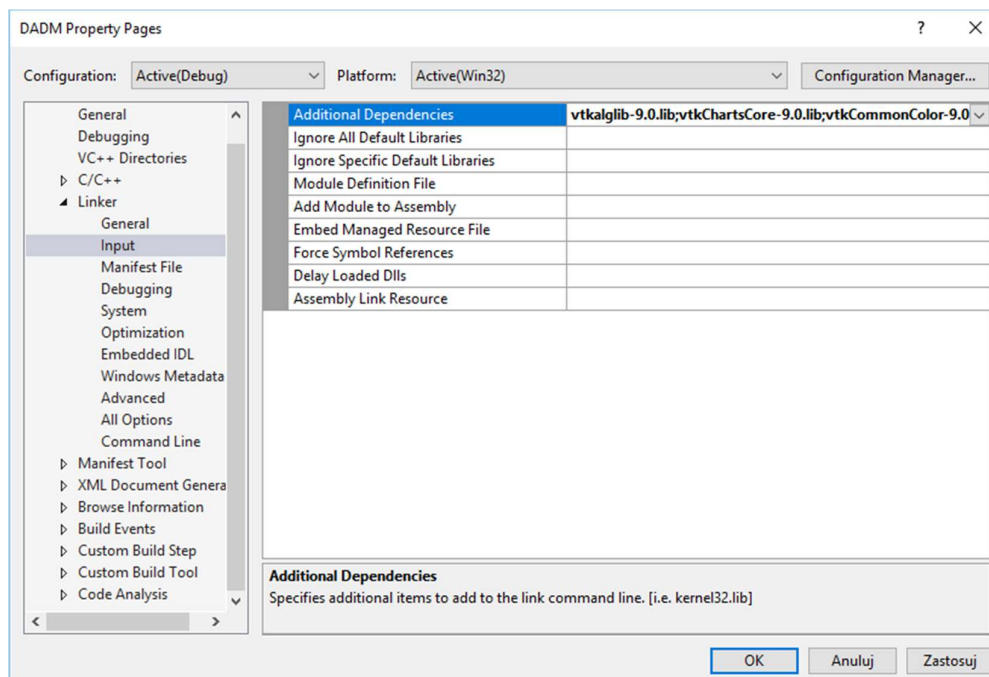
W zakładce C/C++ -> General w polu *Additional Include Directories* dodaj folder zawierający pliki nagłówkowe VTK (znajduje się on w folderze z instalacją, np. C:\VTK-install\include\vtk-9.0):

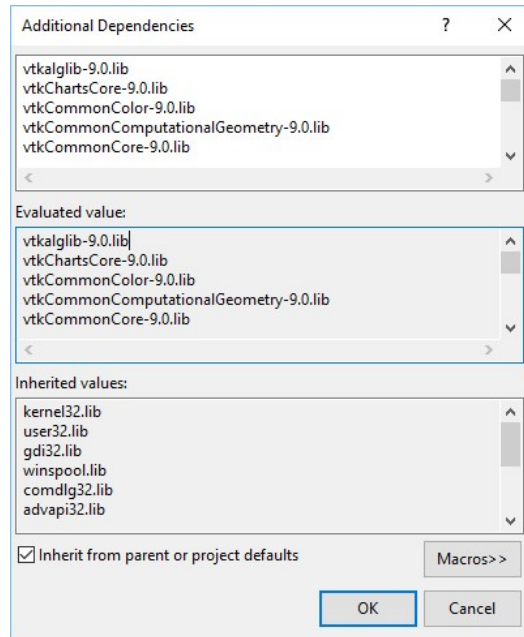


W zakładce *Linker* -> *General* w polu *Additional Library Directories* dodaj folder zawierający biblioteki VTK (znajduje się on w folderze z instalacją, np. C:\VTK-install\lib):




W zakładce *Linker* -> *Input* w polu *Additional Dependencies* dodaj wszystkie biblioteki VTK:

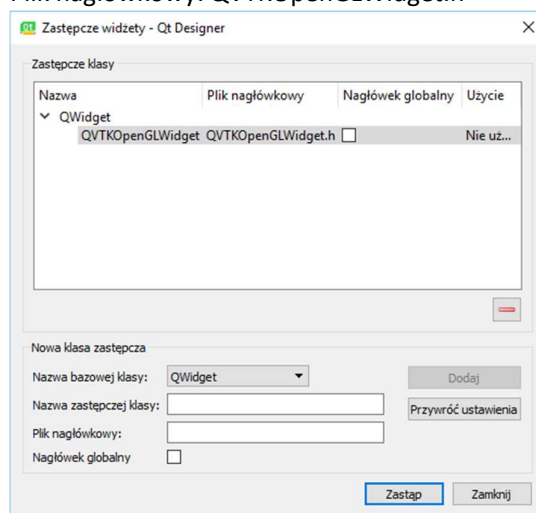




Nazwy wszystkich plików .lib znajdują się w pliku libs.txt w repozytorium, wystarczy że je przekopiujesz do powyższego okienka.

Aby dodać Widget VTK do okienka programu:

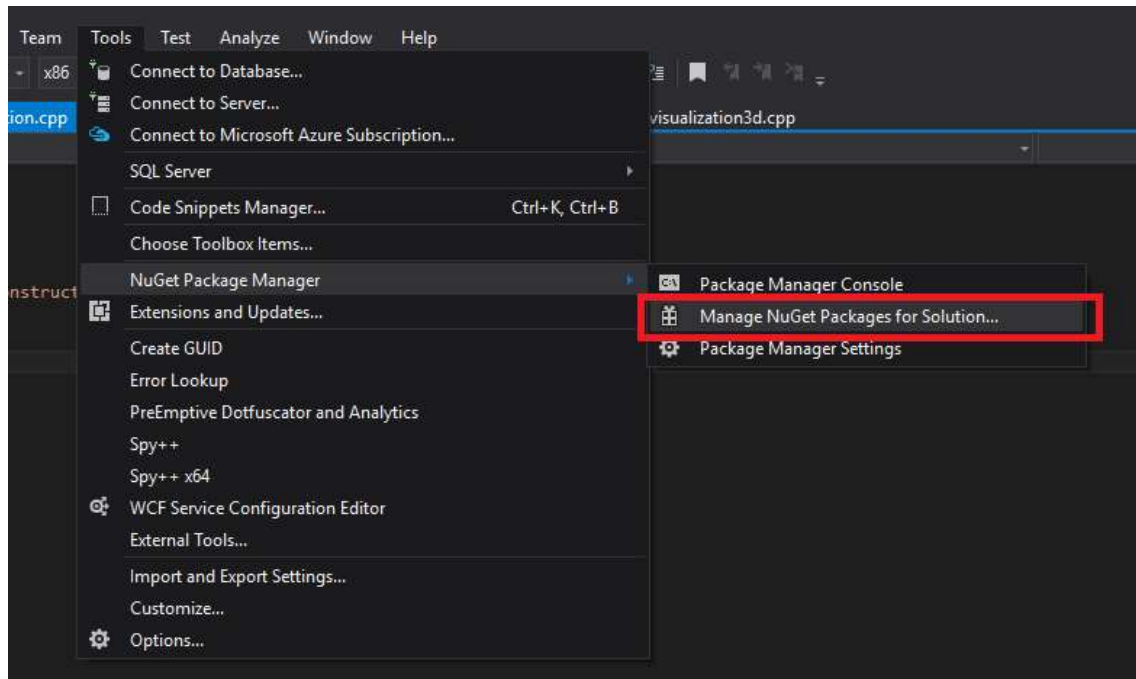
- Wybierz plik .ui
- Umieść w oknie QWidget
-  Widget
- Naciśnij na niego prawym przyciskiem myszy
- W menu kontekstowym naciśnij *Zastąp*
- Dodaj nową klasę zastępczą:
 - Nazwa bazowej klasy: QWidget
 - Nazwa zastępczej klasy: QVTKOpenGLWidget
 - Plik nagłówkowy: QVTKOpenGLWidget.h



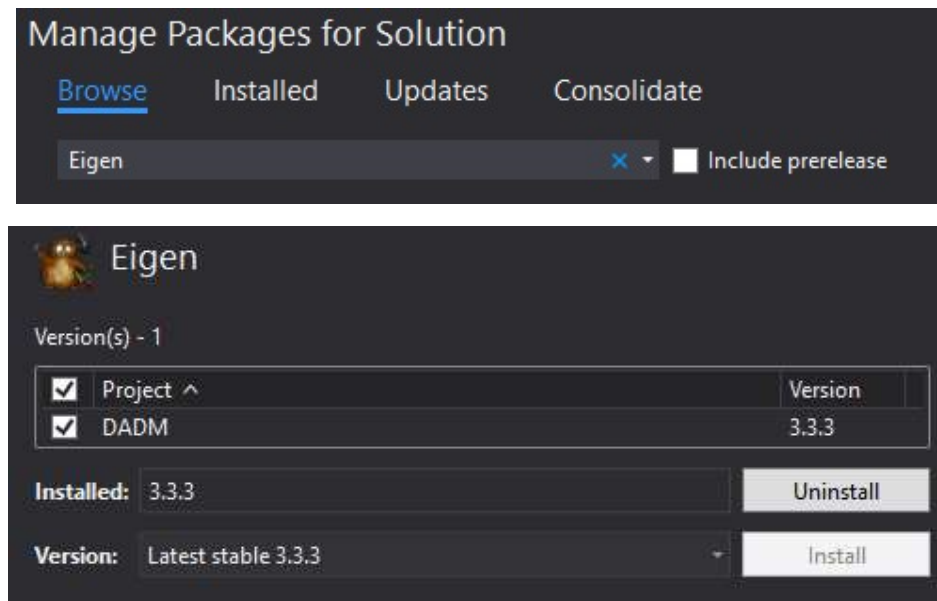
- Naciśnij przycisk *Zastąp*
- Możesz teraz odnosić się do obiektu QVTKOpenGLWidget

DODAWANIE EIGEN DO PROJEKTU

W Visual Studio wybierz *Tools -> NuGet Package Manager -> Manage NuGet Packages for Solution...*



Wyszukaj Eigen, zaznacz projekt i naciśnij przycisk *Install*

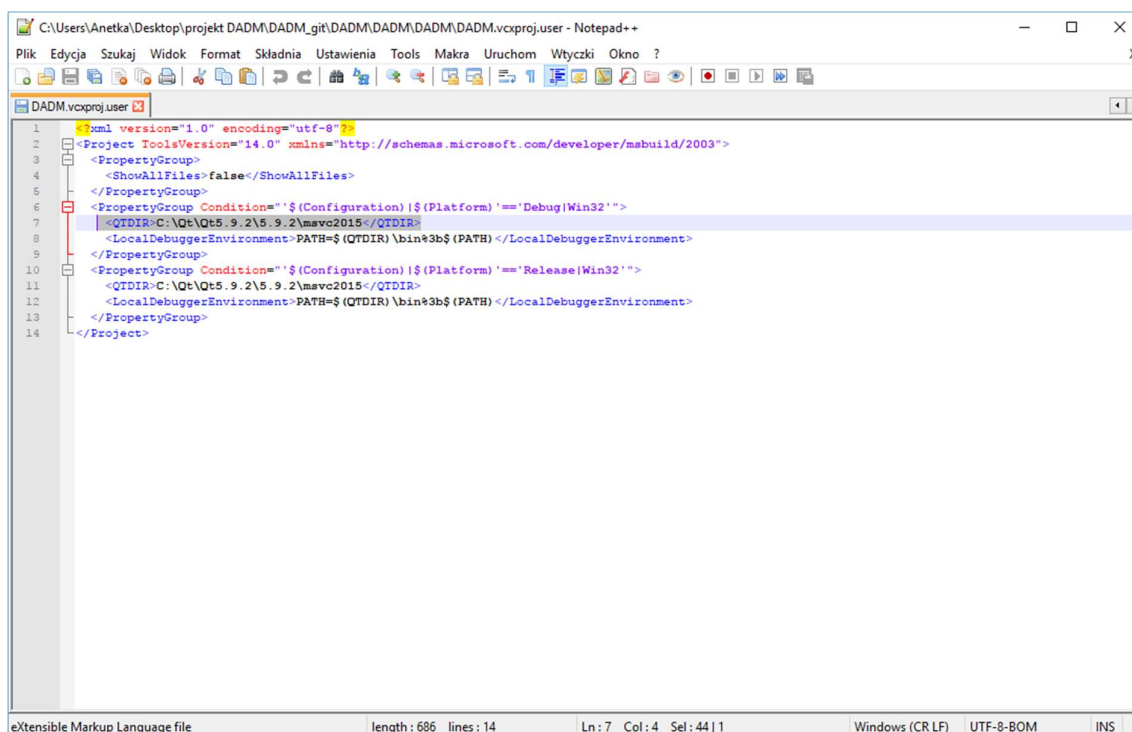


FAQ:

1. Podczas kompilacji, kompilator „nie widzi” funkcji Qt

```
cannot open source file "QMainWindow"
cannot open source file "QMessageBox"
class "DADM" has no member "DADM"
identifier "QWidget" is undefined
identifier "parent" is undefined
expected a ':'
expected a declaration
cannot open source file "QWidgets/QMainWindow"
DADM    helpermethods.h  3
DADM    dadm.cpp        5
DADM    dadm.cpp        7
DADM    dadm.cpp        7
DADM    dadm.cpp        7
DADM    dadm.cpp        7
DADM    dadm.cpp        7
DADM    dadm.cpp        21
DADM    dadm.h          4
```

W pliku projektu o rozszerzeniu .vsxproj.user zmień ścieżkę do Qt do Twojej instalacji:



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Project ToolsVersion="14.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3   <PropertyGroup>
4     <ShowAllFiles>false</ShowAllFiles>
5   </PropertyGroup>
6   <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|Win32'">
7     <QDIR>C:\Qt\Qt5.9.2\5.9.2\mavc2015</QDIR>
8     <LocalDebuggerEnvironment>PATH=$(QDIR)\bin\3b$(PATH) </LocalDebuggerEnvironment>
9   </PropertyGroup>
10  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Release|Win32'">
11    <QDIR>C:\Qt\Qt5.9.2\5.9.2\mavc2015</QDIR>
12    <LocalDebuggerEnvironment>PATH=$(QDIR)\bin\3b$(PATH) </LocalDebuggerEnvironment>
13  </PropertyGroup>
14 </Project>
```

2. Dodawałem nowe kontrolki do UI w Visual Studio i nie mam do nich dostępu w projekcie

Kliknij prawym przyciskiem myszy na plik .ui i wybierz Compile. Następnie w Visual Studio wybierz Project -> Rescan Solution

3. Failed to run MSBuild command w Cmake'u

```
CMake Error at CMakeLists.txt:14 (project):
  Failed to run MSBuild command:

    C:/Program Files (x86)/MSBuild/14.0/bin/MSBuild.exe
```

Prawdopodobnie nie zainstalowałeś bibliotek Visual C++ podczas instalacji Visual Studio. Pobierz Build Tools i zainstaluj: <http://landinghub.visualstudio.com/visual-cpp-build-tools>

4. Na stronie internetowej z pobieraniem Visual Studio dostaję błąd: Niestety, nie udało nam się znaleźć dla Ciebie żadnych plików do pobrania.

Najprawdopodobniej nie posiadasz subskrypcji Visual Studio Dev Essentials. Możesz ją uzyskać pod tym linkiem: <https://www.visualstudio.com/pl/dev-essentials/>