**Installation Guide**

**Installing GraphMIC with QT, ITK and OpenCv**

**Required components**

* **Windows 7 Operating system**
* **Visual Studio 2017 or Visual Studio 2015 compiler.**
* **CMake GUI**
* **Qt 5.11.0 with Qt Creator**
* **ITK**
* **OpenCv 3.4.2**

With a successfully execution of the Installations guide steps, you should be able to build and run the project.

**Please read the instructions carefully and pay attention to the complementary images.**

**Contents**

[1. Installing Debugging Tools from the Windows SDK 2](#_Toc535410212)

[2. Installing Visual Studio 3](#_Toc535410213)

[3. Preparing CMake 4](#_Toc535410214)

[4. Installing Qt 5](#_Toc535410215)

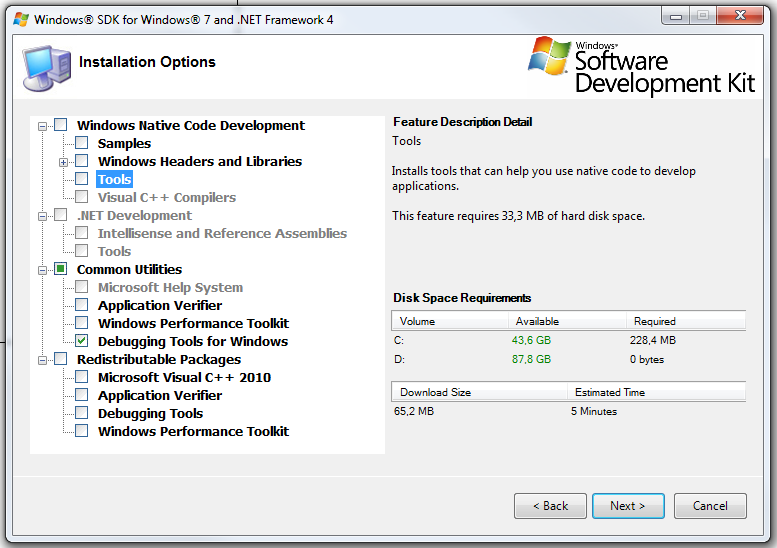
[5. Installing and Build ITK 7](#_Toc535410216)

[6. Installing and Build OpenCV: 9](#_Toc535410217)

[7. Download and configure the project (GraphMIC): 10](#_Toc535410218)

## Installing Debugging Tools from the Windows SDK

Windows SDK contains a CDB ***Debugger***, you will need it in case you want to debug the application on ***Qt***.

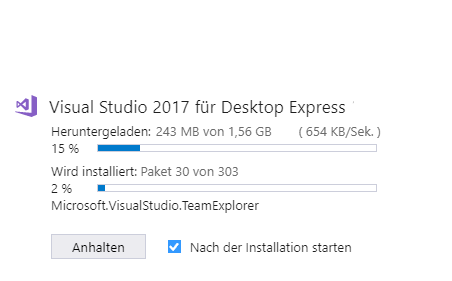
* 1. Go to the official Microsoft site <https://www.microsoft.com/en-us/download/details.aspx?id=8279>
  2. Click on ***Download***, to download the installer.
  3. ***Run*** the installer.
  4. Read the licence and select ***I Agree***
  5. Click next.
  6. ***Select*** the Options like on the image, click next.
  7. If the Installer Fails, you must close the installer, uninstall all versions of the Visual C++ 2010 Redistributable and try to install the Debugging Tools for windows again.

## Installing Visual Studio

Installing Visual Studio Express 2017 for Windows Desktop

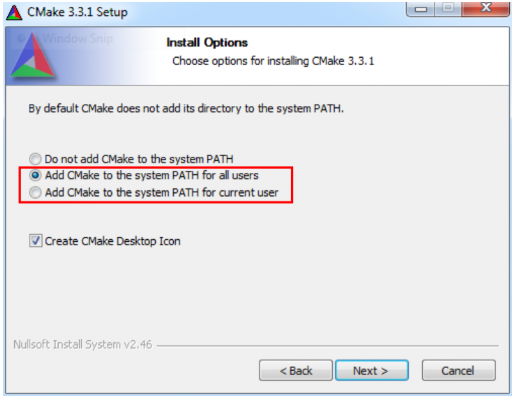
* + - Go to the Microsoft website [https://visualstudio.microsoft.com/de/vs/express/](https://visualstudio.microsoft.com/de/vs/express/%20)
    - Scroll to the bottom of the site and click on ***“Express 2017 for Windows Desktop”,*** the installer will be downloaded.



* + - Go to the downloads folder, select the installer and run it as administrator.
    - You may see a pop-up “We have to prepare some things...” click on ***continue.***
    - ***Click on install.***
    - Wait until the installer is done and close it.

## Preparing CMake

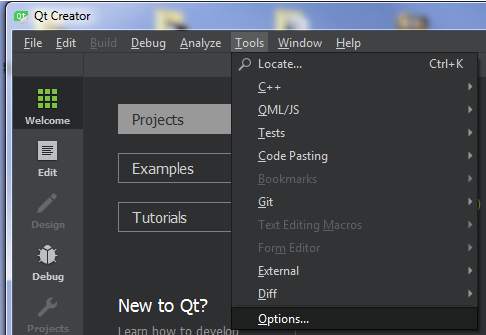
This step is needed if you do not have a pre-installed CMake version on your system.

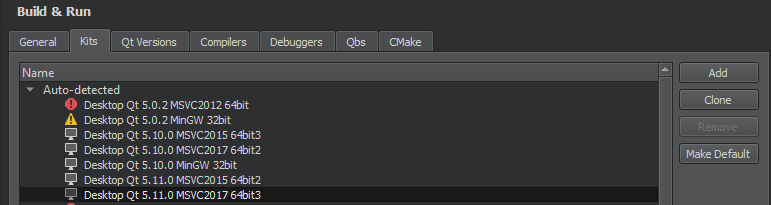
* 1. You can download a windows installer of CMake ([cmake-3.12.1-win64-x64.msi](https://cmake.org/files/v3.12/cmake-3.12.1-win64-x64.msi)) from the [CMake download page](https://cmake.org/download/). Follow the instructions provided on the [CMake installation](https://cmake.org/install/) page for downloading and installing the software

***Note: Add*CMake*to the system path.***

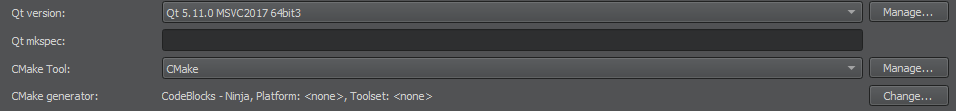
## Installing Qt

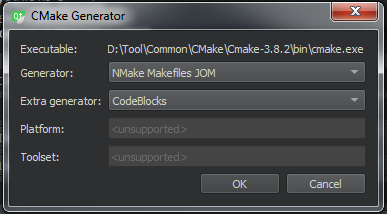
If you already have a Qt version on your system, just open the ***Qt Maintenance Tool***, click on ***next***, ***skip***, select ***add or remove components*** and ***continue, so continue with step 4.4***

* 1. Go to the download site <https://www.qt.io/download>, go to ***open source***, select ***go open source*** and then ***download.***
  2. Run the installer, click on ***next***, ***skip, if the installer throw an error like “Failed to connect to server”, so click on Options and select “no Proxy”***
  3. Select the path where you want to install Qt, ***recommended on*** ***“D:\lib\Qt”***, click ***continue.***
  4. Select the components you want to install, the ones you need are:
     + Under *Qt 5.11.0:* ***MSVC 2015 64-Bit*** or ***MSVC 2017 64-Bit***
     + Under Tools: ***Qt Creator 4.7.0 CBD Debugger Support 4.5.1***
     + You can add or delete Tools every time using the Qt Maintenance Tool.
  5. Finish the installation and ***not open*** QtCreator
  6. Add the ***D:\lib\Qt\Tools\QtCreator\bin path to the environments variables, like on step*** *Fehler! Verweisquelle konnte nicht gefunden werden.****.***
  7. ***Setting the QT Creator up (NMake Generator)***
     + Open the QtCreator, go to Tools and click Options
     + On the left pane, go to ***Build & Run***
     + On the right you will see different tabs, go to ***“Kits”,***
     + Select the ***MSVC precompiled tool***, you have installed on step 4.4 and want to use an on the right site click on ***clone***.

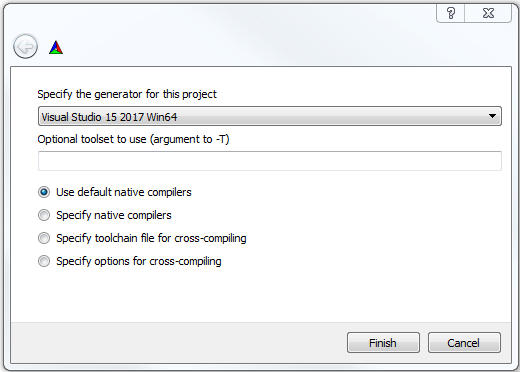
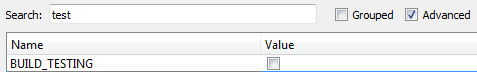
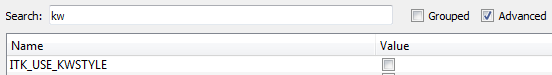
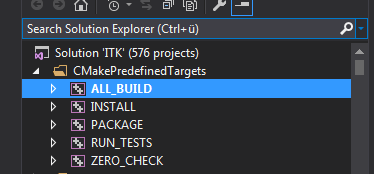
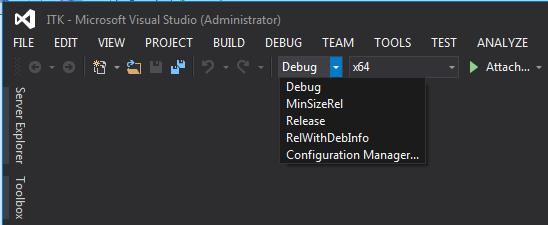


* + - You will see the ***cloned kit*** will appears on the bottom side of the window, ***select it*** and go to ***“CMake generator” and click on change:***



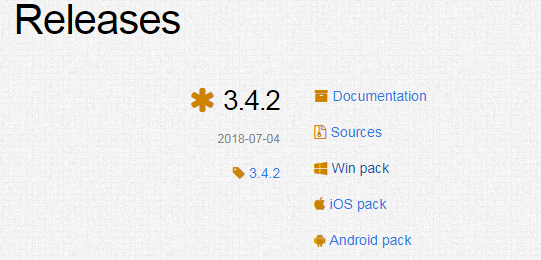
* + -  Select ***NMake MakeFiles JOM as generator, and CodeBlocks as Extra generator.***
    - Click on ***Apply*** and ***ok.***
    - ***If you want to debug, make sure you select the correct path to the debugger like on the follow snip.***
    - ***Close QtCreator.***
    - ***Add the following path to the env variables D:\lib\Qt\5.11.0\msvc2017\_64\bin***

## Installing and Build ITK

* 1. Create a folder **“*ITK*”** with a subfolder **“*bin*”** (recommended on D:\lib\)
  2. Go to <https://github.com/InsightSoftwareConsortium/ITK>
     + On the left site on ***Branch*** select “*release*” and on the right, click on clone or download, then click **download zip**
     + **Move** the Zip file to the *ITK* folder
     + Right-click the Zip file and select **unzip it here**, so it will be a folder “ITK-release” created, this will be the source folder.
  3. Open the CMake GUI
     + On “*Where is the source code*” set the path to the unzipped “***ITK-release***” folder
     + On *“Where to build the binaries”* set the path to the ***bin*** folder you created before
     +  Click on ***Generate*** so on the pop-up window select **“*Visual Studio 15 2017* Win64”** (see the image below, if you have Visual Studio 2015, then select **Visual studio 14 2015 Win64**) as generator and click finish, it will take a couple of minutes to configure the project.
     + **The output window will tell you the configuring is done,** so search for the component **BUILD\_TESTING and uncheck it! then click on configure:**
     + **The output window will tell you the configuring is done,** so search for the component **KWSTYLE and uncheck it! then click on configure:**
     + ****When the configuration is done, then ***click on generate***, so it will generate a Visual studio project.
  4. When it finishes the generation, so you can click on “Open Project” and it will be open with Visual studio.
  5. **On Visual Studio**
     + Wait until the project initializes and parses the files. (you can read it on the blue bar on the bottom)
     + You see on the solution explorer several project files, go to the ***“CMakePredefinedTargets*”** folder, so right-click on ***ALL\_BUILD*** and select ***“build”****, it will take several minutes to complete the process.*
     + Make sure you make this step for both **Debug** and **Release** configuration, so Visual studio will generate the appropriated libraries for each of them.
     + Right-click on ***INSTALL*** and select ***“build”****, m*ake sure you make this step for both **Debug** and **Release** configuration. (This step is needed for the CMake to find the properly paths to ITK)
     + **If Visual Studio fails to compile the Installation subproject, consider trying it again running visual studio as administrator.**

## **Installing and Build OpenCV:**

**This Step performs the OpenCv build and install, you can download and use the pre-compiled binaries from OpenCv online following the step 6.1 (Recommended) or you can build the binaries from yourself on step 6.2 using the same procedure like with ITK on step 5 but it will take a long time.**

* 1. **USING THE MSVC BINARIES: On the Website from OpenCv (**[**https://opencv.org/releases.html**](https://opencv.org/releases.html)**) you can find the already build binaries package for Windows or IOS:**
     + ****Select ***Win pack*** from the latest version, the download ***will take up to 2 Hours for each configuration***.
     + Go to the download folder and click on the opencv….exe you downloaded.
     + Set the path where you want to extract the package (***D:\lib\).***
     + Extract the files.
     + A ***opencv*** folder will be created on ***D:\lib\***
     + Make sure you add the build ***path to the environment variables*** ***(D:\lib\opencv\build).***
  2. **BUILDING OF OpenCv BINARIES with MSVC:**

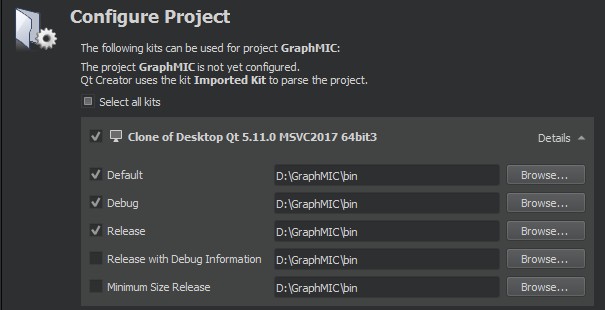
**Follow the same Steps like on 5 with following changes:**

* + - **For 5.1 create a Folder OpenCv** with a subfolder **“*bin*”** ***(recommended on D:\lib\).***
    - **For 5.2 Download the OpenCv from** [**https://opencv.org/releases.html**](https://opencv.org/releases.html), select sources.
    - **Move** the Zip file to the **OpenCv** folder
    - Right-click the Zip file and select **unzip it here**, so it will be a “opencv-3.4.2” folder created, this will be the source folder, ***rename*** it to “opencv”.
    - **For 5.3 omit to uncheck the KWSTYLE component on the CMake Configuration from OpenCv.**
    - **Configure and generate with Cmake and continue making the same procedure as step 5.3**

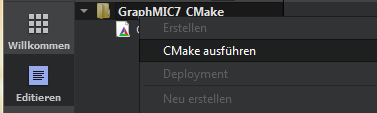
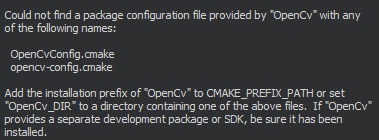
## Download and configure the project (GraphMIC):

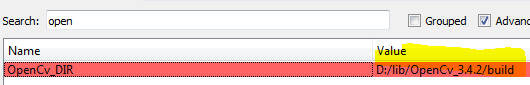
* 1. Clone the project from the following repository: <http://github.conti.de/CES/I_CES_MED_1__GraphMIC> to you folder ***(D:\GraphMIC)***
  2. Open the Qt Creator and click ***open project*** on the *Welcome* tab.
  3. Go to the path of the project ***(D:\GraphMIC)*** and select the ***CMakeLists.txt*** file
  4. Qt Creator will show you the possible configurations you can use for your project, **Select the kit you cloned on step 4.7 installing Qt.** then click on ***details*** select *Default, Debug and Release*

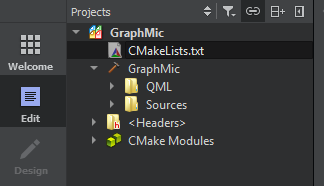
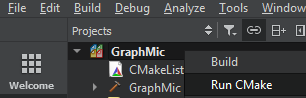
**and click on *“configure project”***

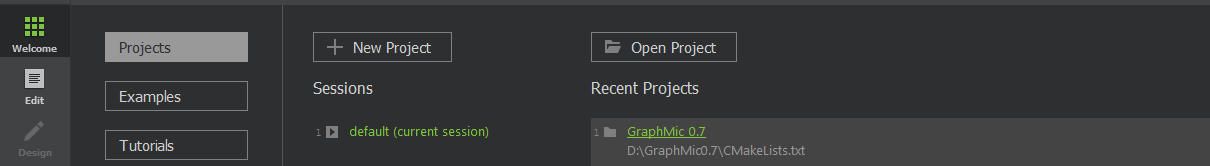


* 1. Make sure as build folder on ***Qt Creator*** is set the **bin** folder **of the project**, if not, then **change to it or create it** like on the following image and go to the tab ***edit*** right-click on the project folder and ***run CMake.***
  2. ***(Optional)***If you have another configuration and it fails, so make sure you delete the created ***CMakeCache.txt*** on the ***GraphMic0.7*\*bin*** folder and the ***CMakeLists.txt.user*** on the ***source*** folder before you try it again.
  3. **(This step can be optional if you add the *opencv build directory* to the environment path variables)** If the CMake ***don’t find the ITK or OpenCv*** libraries automatically, then you will see the error on the ***common output*** tab below like on the following image, so you must set the paths manually using the **CMake GUI:**



* + - **Open CMake Gui**, ***set the GraphMIC source folder and the binary*** folder, click on ***file*** and ***reload cache***
    - ***CMake GUI*** and set the correct build path of the ***OpenCv*** directory on the field ***OpenCv\_Dir*** and click on ***configure***
    - ***(Repeat the proces for ITK\_Dir) Go back to*** Set the correct build path of the ***ITK*** directory on the field ***ITK\_Dir*** and click on ***configure, click on reload cache.***
    - ***Go to Qt Creator, Edit tab*** and click on ***Run Cmake.***
    - ******(Optional)CMake might will throw an error, but the path should be already set, so ***you can ignore it.***
    - **Close CMake GUI**
  1. **Go to Qt Creator** to the tab ***edit*** right-click on the project folder and ***run CMake,*** *the project should be configured jet and* you will see all the folders and files corresponding to the project on the left tab:



* 1. You should be able to build and run the project, maybe you should restart the Qt Creator und re-open it and the project from the recent used projects pane.
  2. If you want to debug the application in C++ and QML go to ***Projects***, and on the ***Run*** option, click on ***enable QML.***

