

Building 3D Slicer

MACbioIDi – February – March 2018

Carlos Luque





Outline



- Introduction
- Environment Setup in Windows
- Slicer Source
- Build Slicer
- Run Slicer
- Testing
- Package Slicer

This tutorial is based on

https://www.slicer.org/wiki/Documentation/Nightly/Developers/Build_Instructions



Introduction 3D Slicer



- Open source software
- Target
 - Medical image informatics,
 - Image processing,
 - Three-dimensional visualization
- Multi-platform
 - Windows
 - Linux
 - Mac OS X



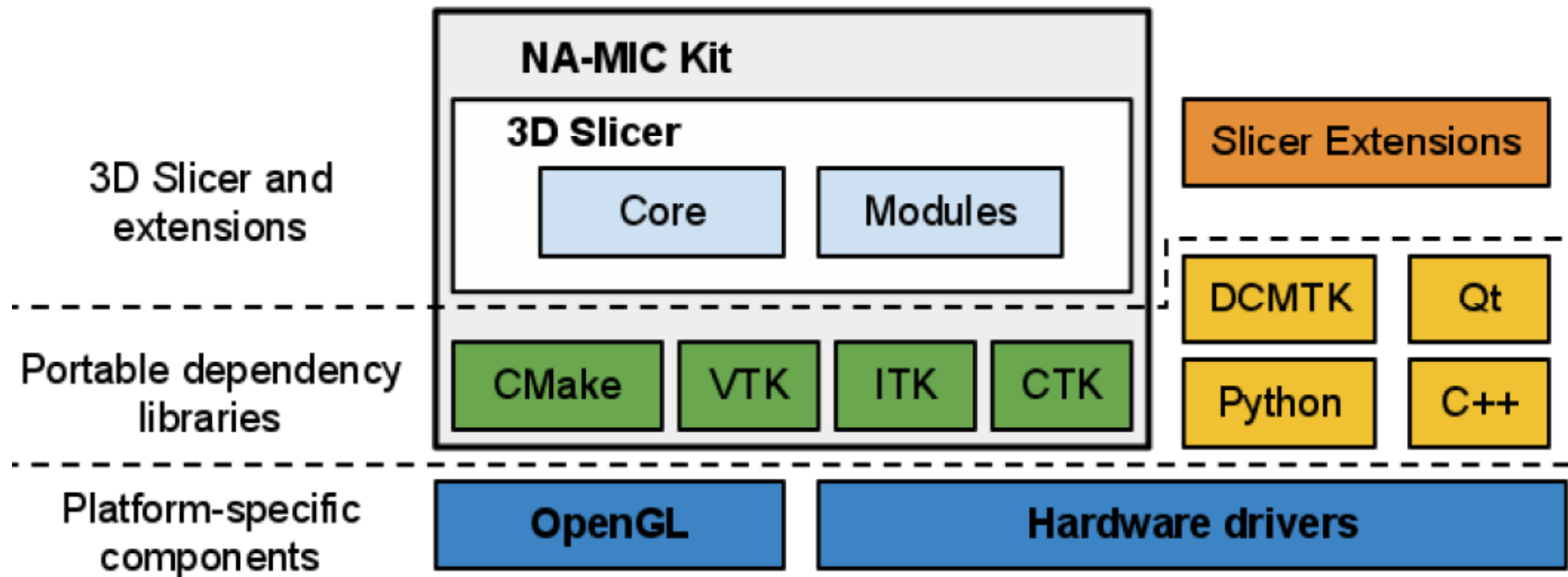
Introduction 3D Slicer



- 3D Slicer Architecture
 - Main application framework (Core)
 - Written in C++ and Python
 - GUI using Qt library
 - Plugins (modules)
 - Written in C++ and Python
 - Custom functionality (external modules)
 - Slicer extensions
 - One or more nodules



Introduction 3D Slicer: Architecture



Article: [3D Slicer as an image computing platform for the Quantitative Imaging Network](#)
A Fedorov, R Beichel, J Kalpathy-Cramer, J Finet... - Magnetic resonance imaging, 2012



Environment Setup



- Cmake \geq 3.8.2
- Version control system
 - Git
 - Subversion (svn)
- Compiler
 - Visual Studio Community 2013
 - Official
 - Visual Studio Community 2015
 - Experimental



Environment Setup



- Graphical user interface (GUI)
 - Qt 4.8.7 for Slicer 4.8.1
 - Qt 5.9.1 for Slicer 4.9 or posterior
- NSIS (optional): Needed if packaging Slicer
 - If you want to create your Slicer distribution



cmake



- Open-source, cross-platform family of tools designed to build, test and package software
- [Download](#)
 - Version 3.10.1 (Latest release version)



Version control system



- Subversion (svn)
 - Clients:
 - GUI: [TortoiseSVN](#)
 - Command Line: [sliksvn](#)
- Git
 - Clients:
 - GUI: [TortoiseGit](#)
 - Command Line: [git](#)



Compiler: Visual Studio Community



- Create Microsoft account
 - [Registration](#)
 - Get a new email address
 - Use your email
- Choose a subscriber access
- Download
 - Visual Studio 2013 with Update 5
 - Visual Studio Community 2013 with Update 5 (.exe)
 - Visual Studio 2015 with Update 3
 - Visual Studio Community 2015 with Update 3 (.exe)



Visual Studio Community 2013



- Installer
 - Optional features to install:
 - Microsoft Foundation Classes for C++



Error: VS2012 x64 Cross tools Command Prompt



- If run the command prompt and you have the error message:
 - ERROR: Cannot determine the location of the VS Common Tools folder.
- Fix:
 - Click right mouse in VS2012 x64 Cross tools command prompt
 - In Target: Switch it to installed visual studio Path

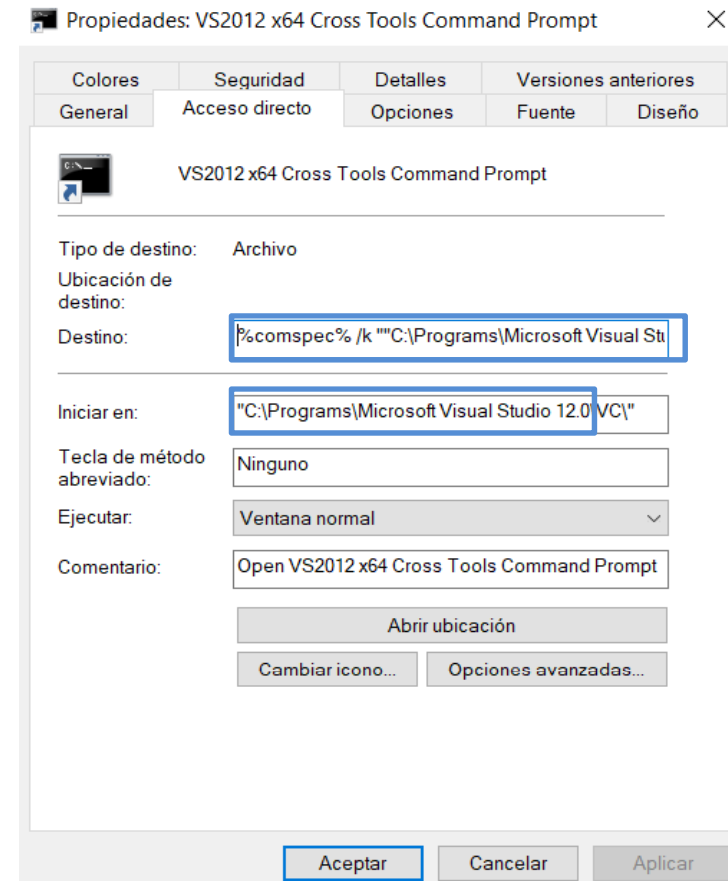
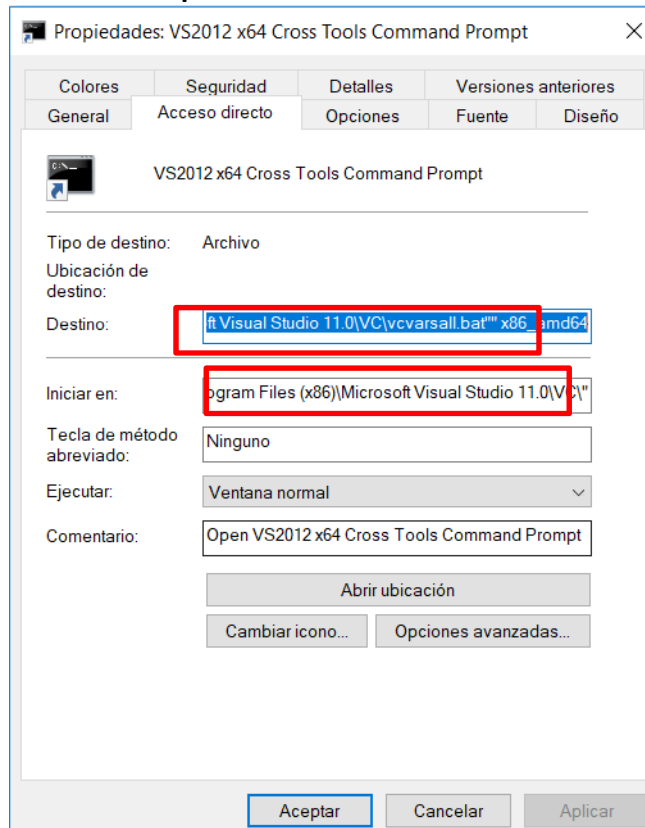


Error: VS2012 x64 Cross tools Command Prompt



Error path: C:\Program Files (x86)\Visual Studio 11.0, because Visual studio is installed in other path

Fix error, assuming the installed visual studio path is c:\Programs\Visual Studio 12.0





Graphical user interface: QT



- A cross-platform application framework that is used for developing application software
- Building Qt 4.8.7 for Slicer version 4.8.1
 - Run VS2012 x64 Cross tools Command Prompt
 - Execute the command line

```
@powershell -Command "$destDir='C:\D\Qt487';$buildType='Release';$qtPlatform='win32-msvc2013';$bits='64';iex ((new-object net.webclient).DownloadString('https://raw.githubusercontent.com/jcfr/qt-easy-build/4.8.7/windows_build_qt.ps1'))"
```
 - If all is correct, the message should be shown
“-- Building Qt – ok”



Slicer Source



- Repositories:
 - Git: [git://github.com/Slicer/Slicer.git](https://github.com/Slicer/Slicer.git)
 - Svn: <http://svn.slicer.org/Slicer4/trunk>
- Git repository is used for:
 - Getting the latest changes
 - Developing Slicer core
- Svn repository is used for:
 - Getting the Slicer release
 - Testing your developing extensions for a specified Slicer release



Slicer Source



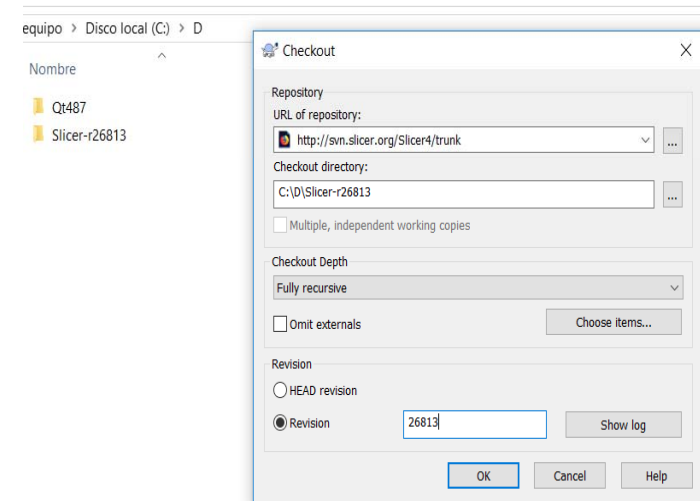
- Windows paths are limited to 250 characters
 - Short path name is better
 - No more than 50 characters
- I recommend the following directories in units
C:/ or D:/
 - D -> Developing
 - Put all source codes
 - B -> Building
 - Put all building directory



Slicer Source



- Download Slicer source
 - Release version (Slicer 4.8.1)
 - Create a folder for the code:
 - For example C:\D\Slicer-r26813
 - Select the code folder
 - C:\D\Slicer-r26813
 - Select SVN checkout
 - url of repository
 - » <http://svn.slicer.org/Slicer4/trunk>
 - Checkout directory
 - » C:\D\Slicer-r26813
 - Revision 26813
 - OK





Slicer Source: Svn revision



Release	SVN revision
4.8.1	26813
4.8.0	26489
4.6.2	25516



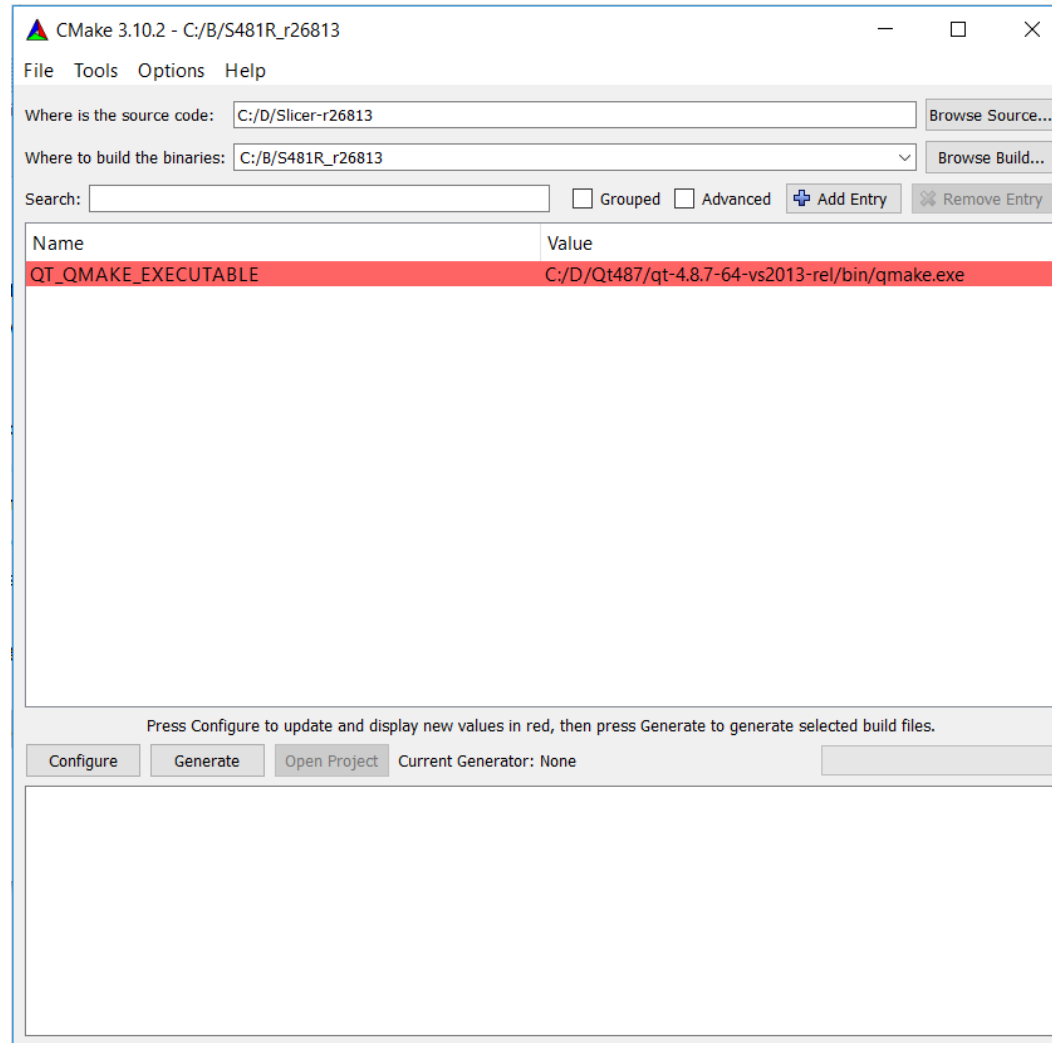
Building Slicer: Create solution



- Make the building via cmake-gui for Slicer 4.8.1
 - Create a building folder:
 - For example C:\B\S481R_r26813
 - Run cmake-gui
 - Select source code folder in “Where is source code”
 - For example C:\D\Slicer-r26813
 - Select the building folder in “Where to build binaries”
 - For example C:\B\S481R_r26813
 - Set the variable “QT_QMAKE_EXECUTABLE ” to the path of the qmake.exe
 - For example C:\D\Qt487\qt-4.8.7-64-vs2013-rel\bin\qmake.exe



Building Slicer: Create solution





Building Slicer: Create solution



- Run cmake-gui (continue)
 - Click the button “Configure”
 - Select generator
 - Visual Studio 12 2013 Win64
 - If building in release mode (this presentation)
 - Set the variable “CMAKE_CONFIGURATION_TYPES” to “Release”
 - If building in debug mode
 - The variable “Slicer_USE_SimpleITK” must be turned off
- Click “Generate”
- The same process for developing Slicer core or latest slicer version



Building slicer



- You can click Open Project from cmake-gui
- Or
- Start Windows Explorer
 - Open the solution in Visual Studio
 - For example C:\B\S481R_r26813\Slicer.sln
 - Select build configuration. Usually **Release** or **Debug**
 - This presentation: Select Release in x64
 - Select menu Build -> Build Solution
- If no error in compilation, the 3D Slicer is built



RUN Slicer



- Start Windows Explorer
 - Run Slicer.exe
 - For example: C:\B\S481R_r26813\Slicer-build\Slicer.exe



Testing



- Start Windows Explorer.
 - Open the solution of slicer in Slicer-build
 - For example: C:\B\S481R_r26813\Slicer-build\Slicer.sln in Visual Studio
 - Select build configuration.
Usually **Release** or **Debug**
 - This presentation: Select Release in x64
 - In the Solution Explorer, right click on **RUN_TESTS** project and then select **Build**



Packaging



- Start Windows Explorer.
 - Open the solution of slicer in Slicer-build
 - For example: C:\B\S481R_r26813\Slicer-build\Slicer.sln in Visual Studio
 - Select **Release** build configuration.
 - In the Solution Explorer, right click on **PACKAGE** project in group **CMakePredefinedTargets** and then select **Build**

Building 3D Slicer

Annex





Visual Studio Community 2015



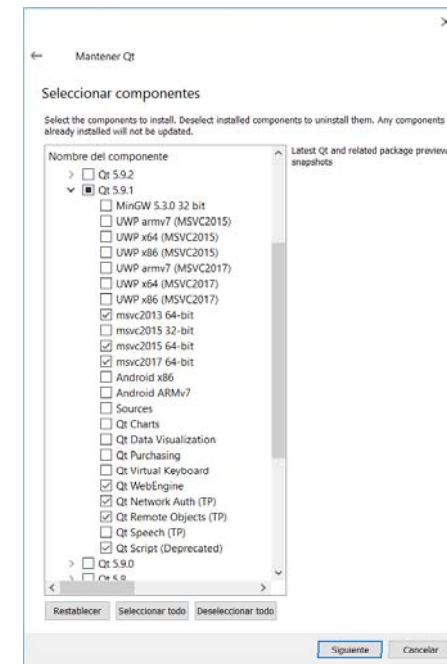
- Download from [subscriber access](#) in Microsoft webpage
- Installer
 - Select in windows “Features”
 - Programming Languages
 - Visual C++
 - Windows and Web Development
 - PowerShell Tools for Visual Studio
 - Common Tools
 - Git for Windows
 - Visual Studio Extensibility Tools Update 3



Graphical user interface: QT5.9.1



- Run a online installer manager
- Specify the directory to install
- Choose Qt 5.9.1 in Select Component window
 - Select (see figure)
 - msvc2013 64-bit
 - msvc2015 64-bit
 - Qt Webengine
 - Qt Script





Graphical user interface: QT



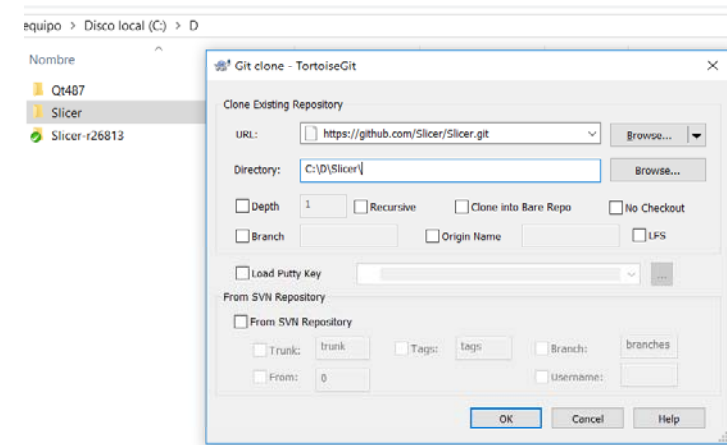
- Install Qt 5.9.1 for Slicer version 4.9 or posterior versions
 - [Download](#)
 - <https://www.qt.io/>
 - link
 - » Developers
 - » Download
 - Choose Open Source
 - Download an online installer manager to choose the Qt version:
 - » 5.9.1



Slicer Source: Download



- Download Slicer source
 - Latest version (Slicer 4.9 nightly)
 - Create a folder for the code:
 - For example C:/D/Slicer
 - Select the code folder
 - C:/D/Slicer
 - Select GIT Clone
 - URL
 - » <https://github.com/Slicer/Slicer.git>
 - Directory
 - » C:\D\Slicer
 - OK





Slicer Source:Developing core (optional)



- Setup the development environment:
 - Run git bash
 - Go to Slicer source folder
 - For example C:/D/Slicer
 - Run `./Utilities/SetupForDevelopment.sh`



Slicer Source: Developing core (optional)



- Configure the git svn bridge to ensure the mapping with svn revision:
 - Run git bash
 - Go to Slicer source folder
 - For example: `cd C:/D/Slicer`
 - Run
 - `git svn init http://svn.slicer.org/Slicer4/trunk`
 - `git update-ref refs/remotes/git-svn refs/remotes/origin/master`
 - `git checkout master`
 - `git svn rebase`