# Compile XLiFE++ libraries for Visual Studio on Windows, using cmake

**Eric Lunéville**

# Requirements

- **XLiFE++ source installed**

  *http://uma.ensta-paristech.fr/soft/XLiFE++/?module=main&action=dl*
- **Visual Studio installed**

  *v14 2015 community free : https://www.visualstudio.com/fr-fr/downloads/download-visual-studio-vs.aspx*
- **cmake installed (>2.8.2)** *https://cmake.org/download/*
- **paraview installed** *http://www.paraview.org/download/*
- **gmsh installed** *http://gmsh.info/#Download*

**Extra libraries (optionnal)**

**blas, lapack, suitesparse, arpack**

# Run cmake


CMake 3.2.2 - E:/Eric/Eric_bureau/xlife++_release/vs15

File Tools Options Help

Where is the source code: E:/Eric/Eric_bureau/xlife++_release Browse Source...

Where to build the binaries: E:/Eric/Eric_bureau/xlife++_release/vs15 Browse Build...

Search: Grouped Advanced Add Entry Remove Entry

Name Value

set XLiFE++ install dir and buil dir

ask to create if build_dir does not exist

Create Directory

Build directory does not exist, should I create it?

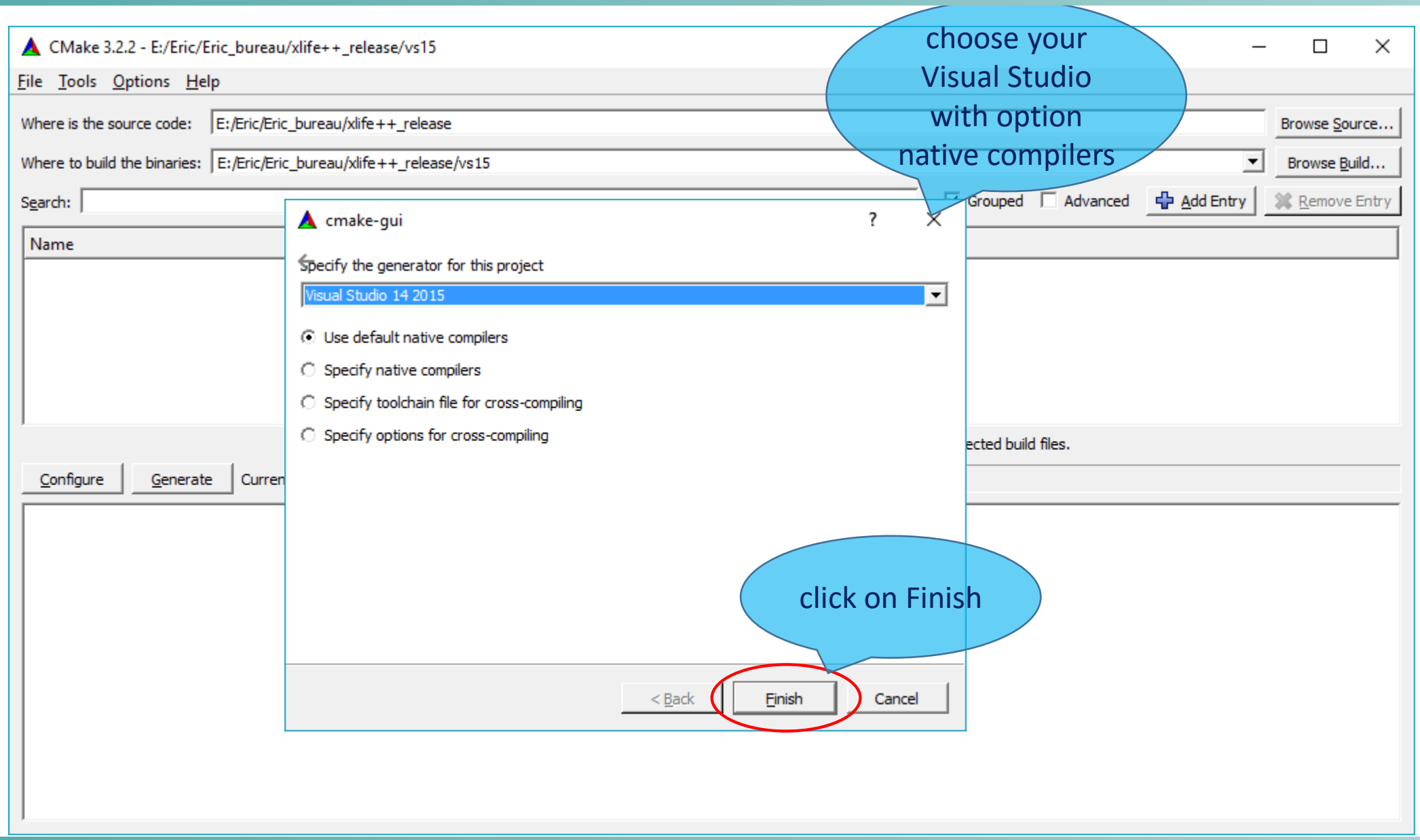
Directory: E:/Eric/Eric_bureau/xlife++_release/vs15

Yes No

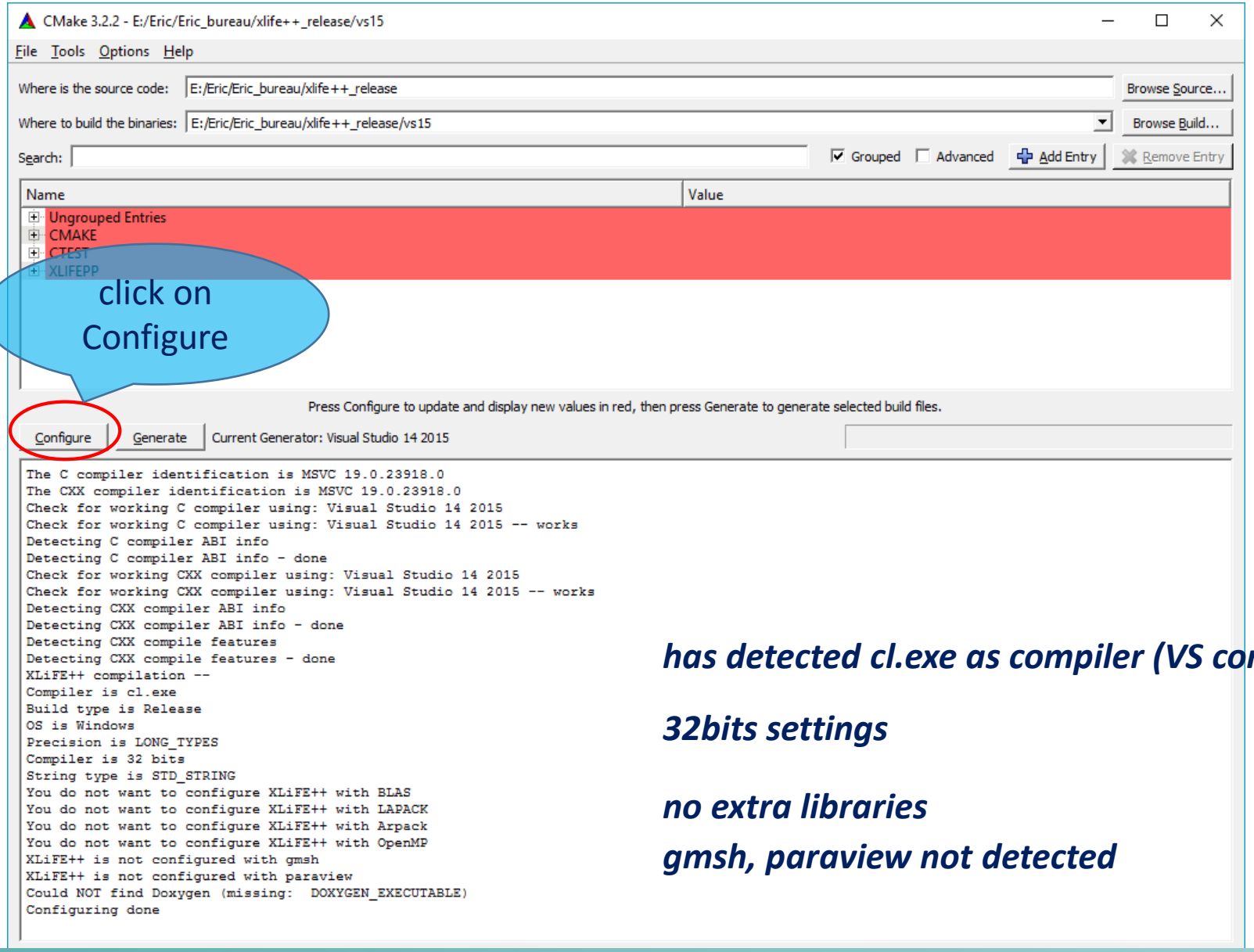
Press Configu... nerate selected build files.

Configure Generate Current Generator: None

# Choose Visual Studio as builder



# First configure



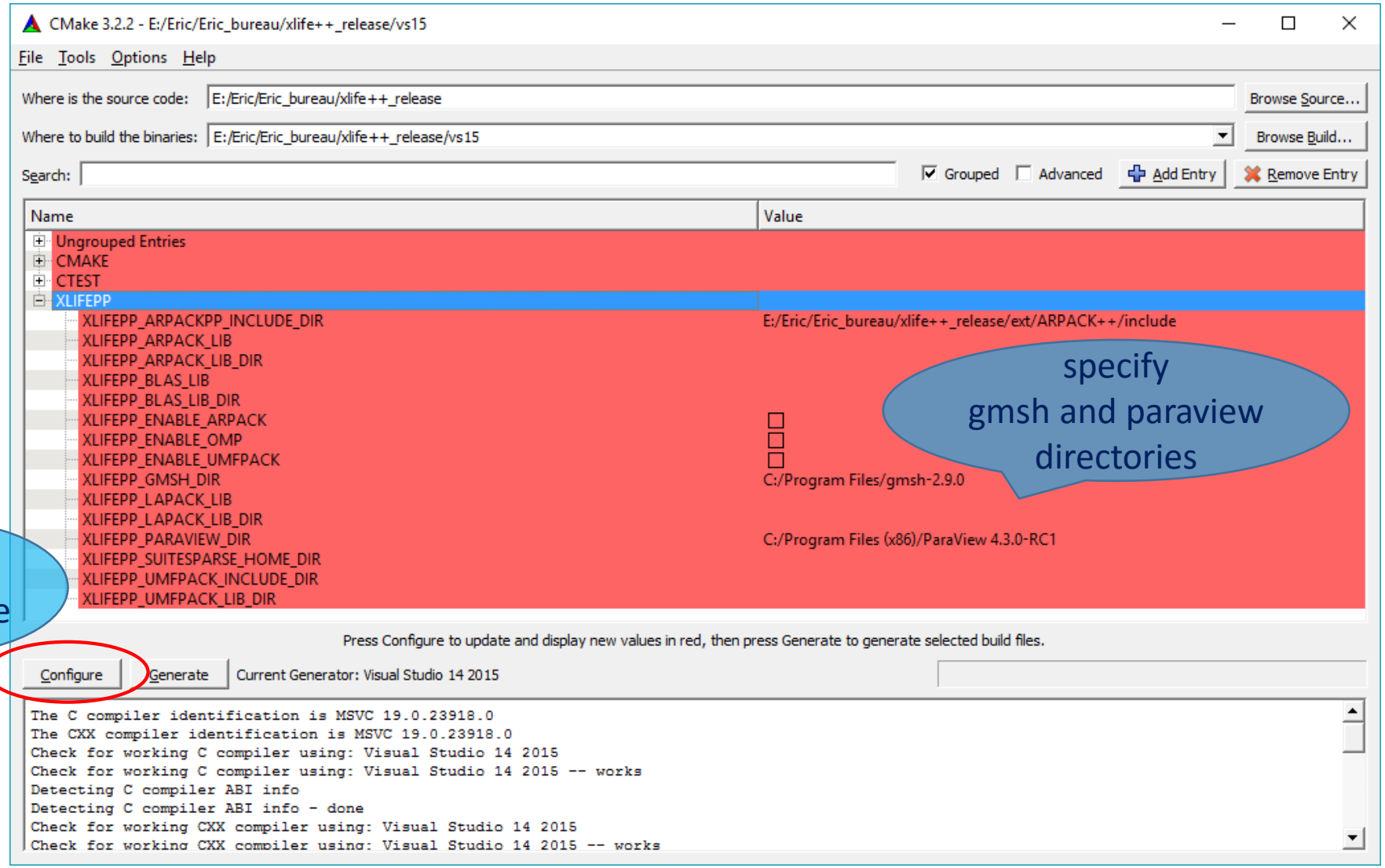

***has detected cl.exe as compiler (VS compiler)***

***32bits settings***

***no extra libraries***

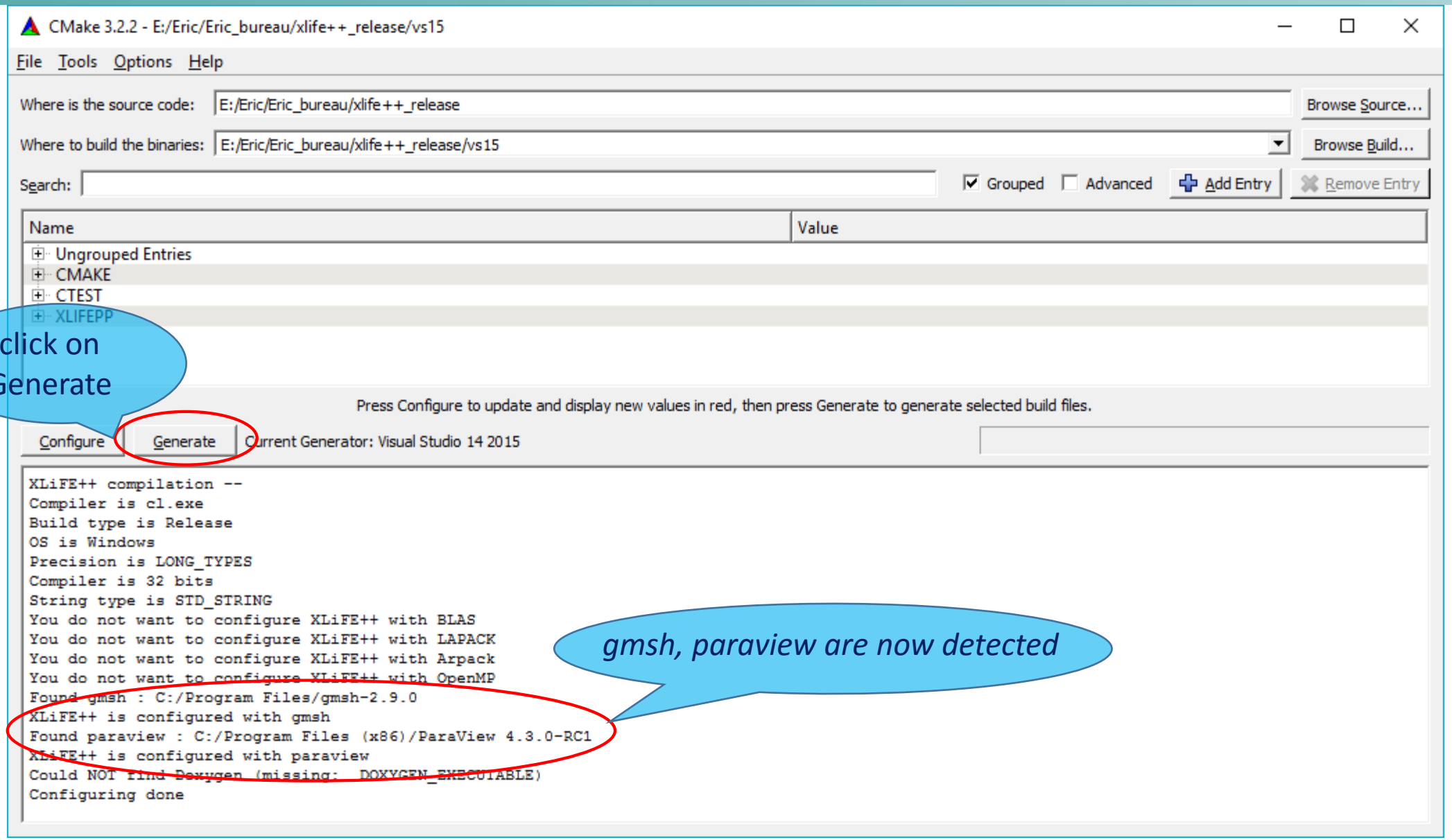
***gmsh, paraview not detected***

# Second configure

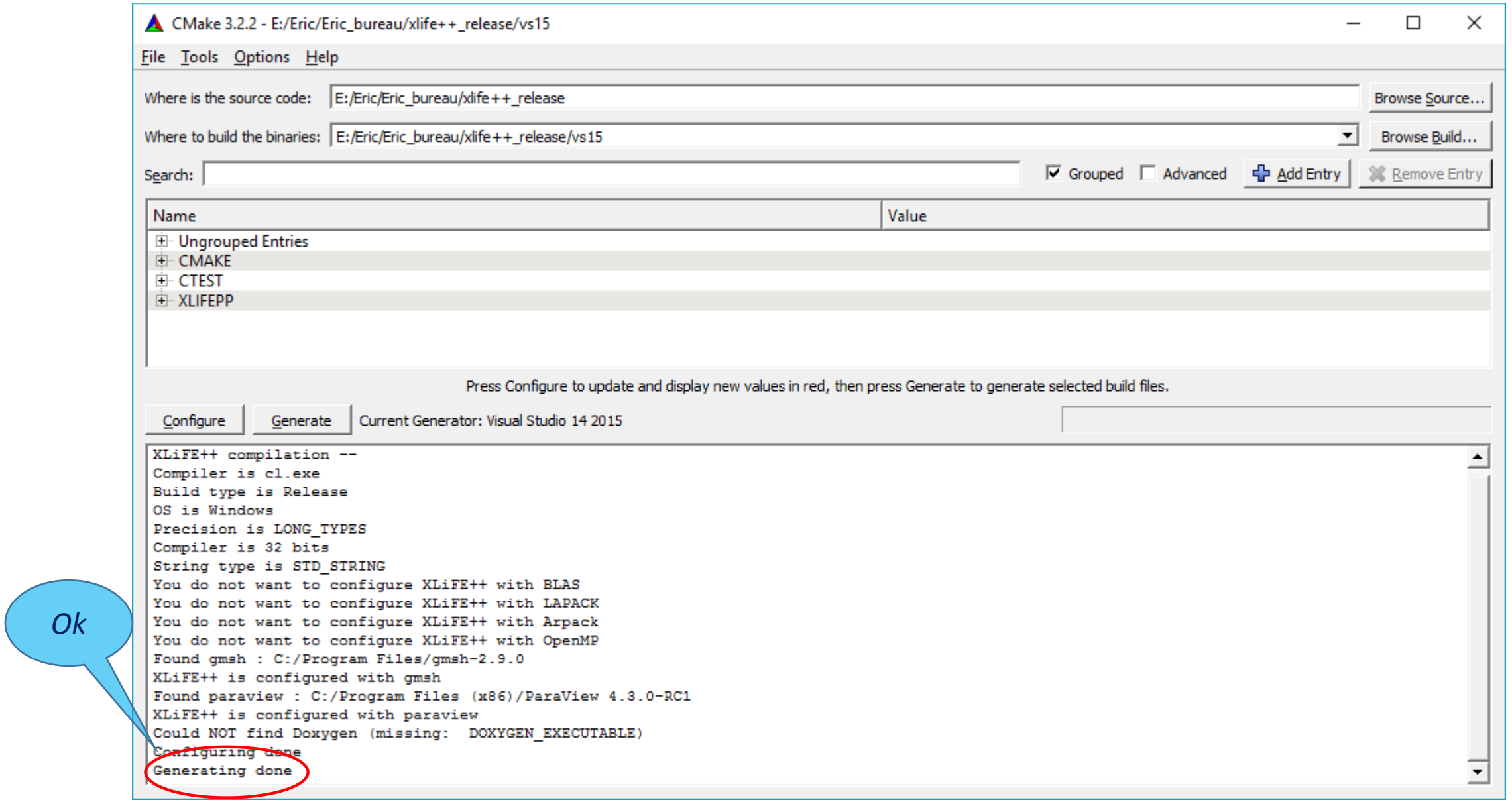



click again on Configure

# Now generate



# Build is completed



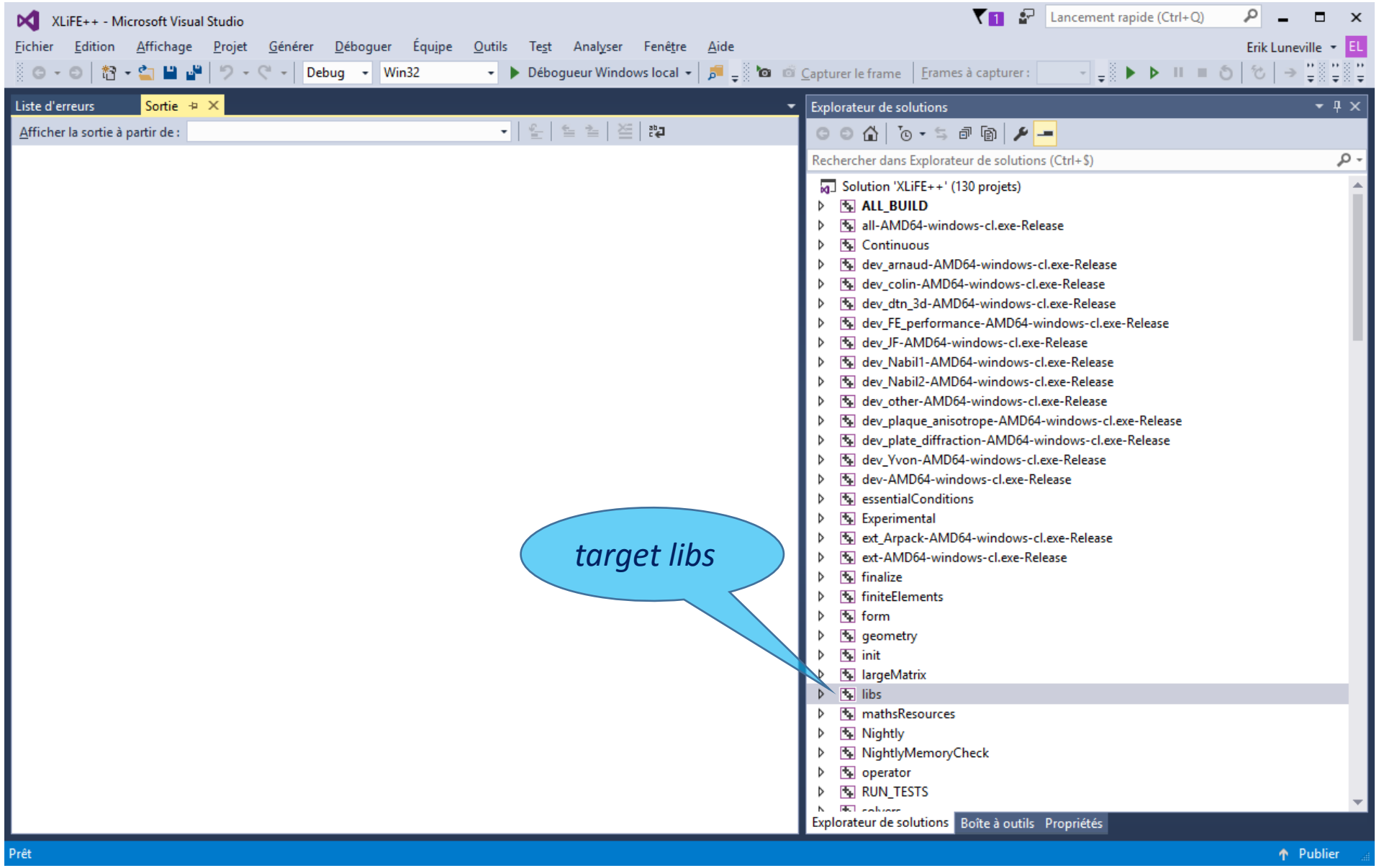
# Build is completed

Newton › Users (E:) › Eric › Eric_bureau › xlife++_release › vs15 ›

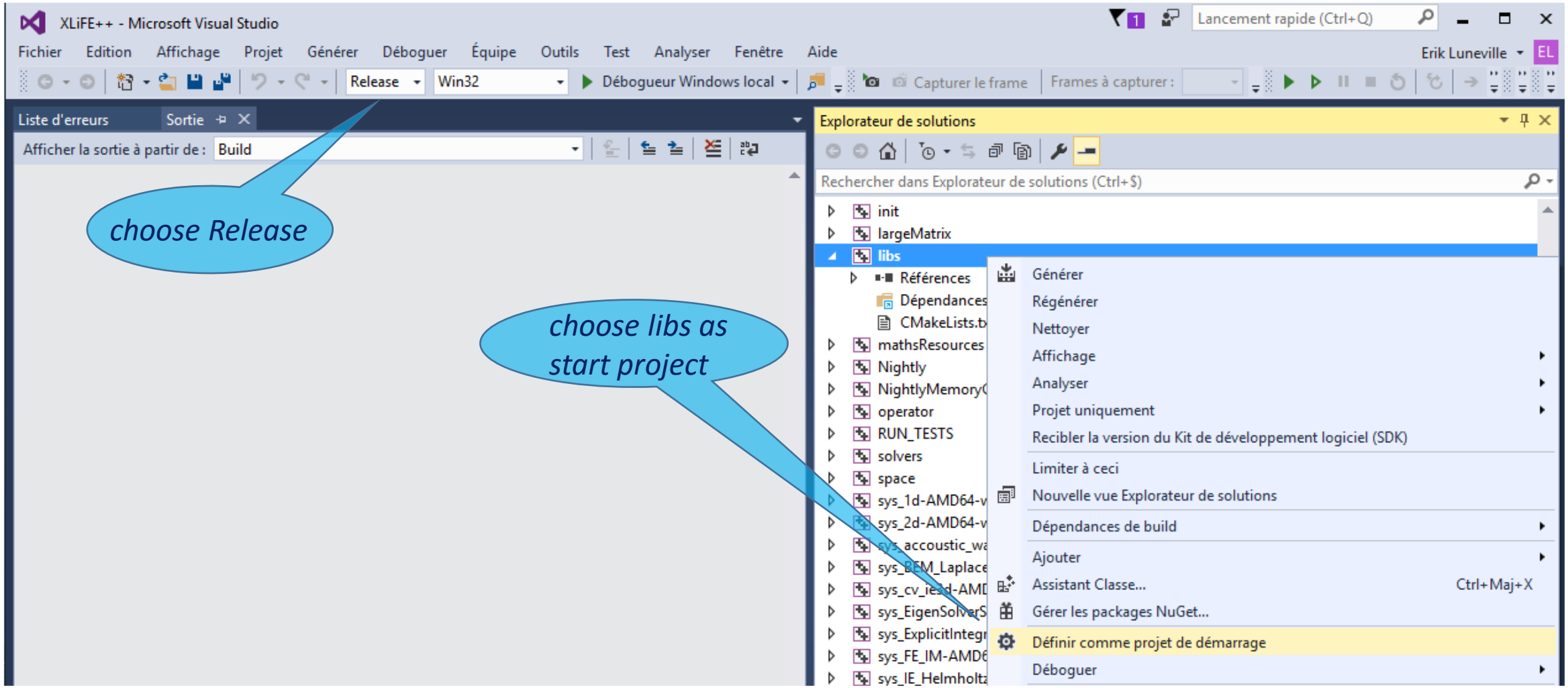
| | | |
|---|---|---|
| CMakeFiles | finiteElements.vcxproj | NightlyMemoryCheck.vcxproj.filters |
| Testing | finiteElements.vcxproj.filters | operator.vcxproj |
| tests | form.vcxproj | operator.vcxproj.filters |
| ALL_BUILD.vcxproj | form.vcxproj.filters | RUN_TESTS.vcxproj |
| ALL_BUILD.vcxproj.filters | geometry.vcxproj | RUN_TESTS.vcxproj.filters |
| cmake_install.cmake | geometry.vcxproj.filters | setup.hpp |
| CMakeCache.txt | init.vcxproj | solvers.vcxproj |
| Continuous.vcxproj | init.vcxproj.filters | solvers.vcxproj.filters |
| Continuous.vcxproj.filters | largeMatrix.vcxproj | space.vcxproj |
| CTestTestfile.cmake | largeMatrix.vcxproj.filters | space.vcxproj.filters |
| DartConfiguration.tcl | libs.vcxproj | term.vcxproj |
| essentialConditions.vcxproj | libs.vcxproj.filters | term.vcxproj.filters |
| essentialConditions.vcxproj.filters | mathsResources.vcxproj | utils.vcxproj |
| Experimental.vcxproj | mathsResources.vcxproj.filters | utils.vcxproj.filters |
| Experimental.vcxproj.filters | Nightly.vcxproj | XLiFE++.sln |
| finalize.vcxproj | Nightly.vcxproj.filters | ZERO_CHECK.vcxproj |
| finalize.vcxproj.filters | NightlyMemoryCheck.vcxproj | ZERO_CHECK.vcxproj.filters |

Visual Studio project

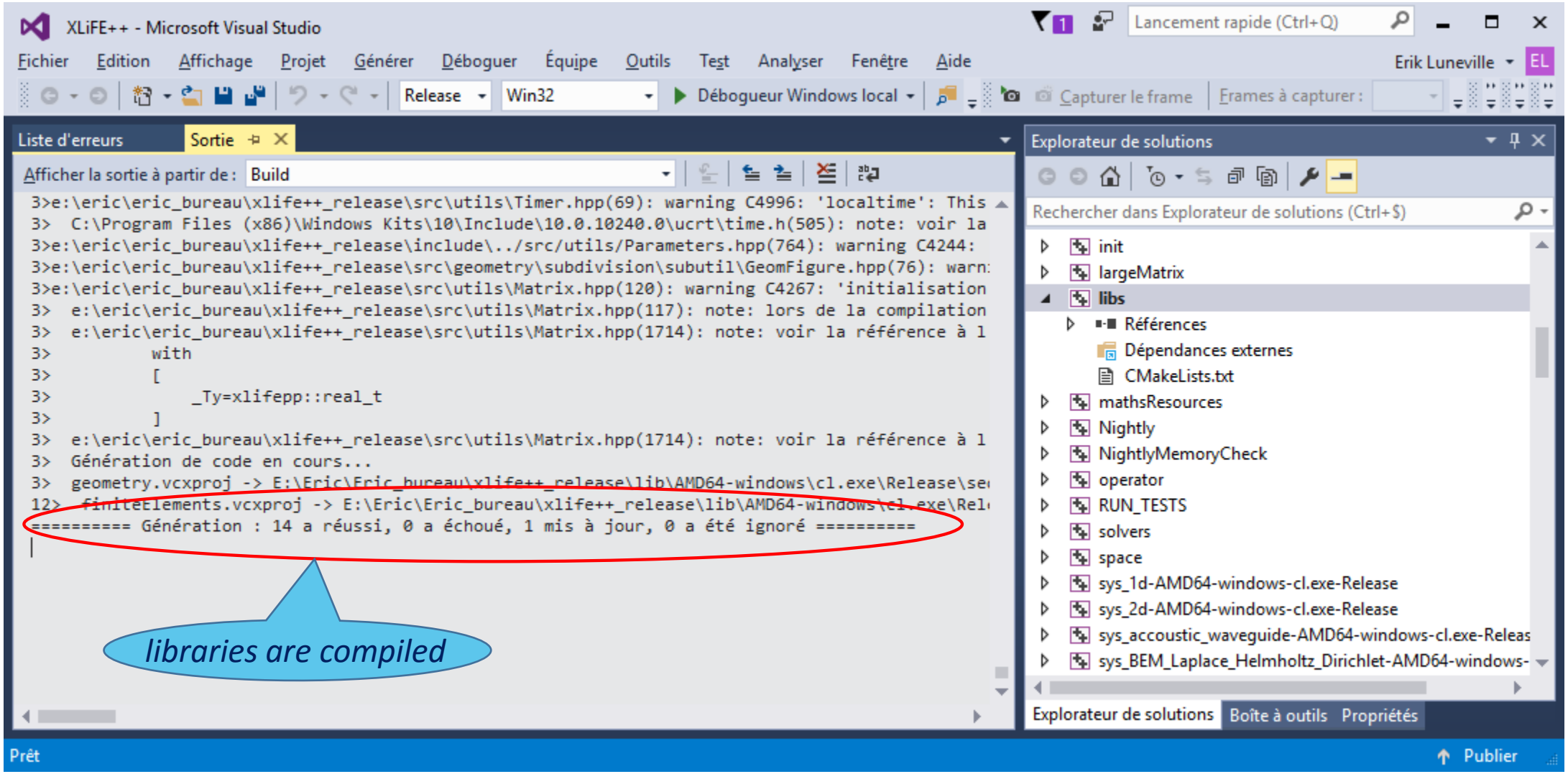
# Open Visual Studio Project



# Compile libraries

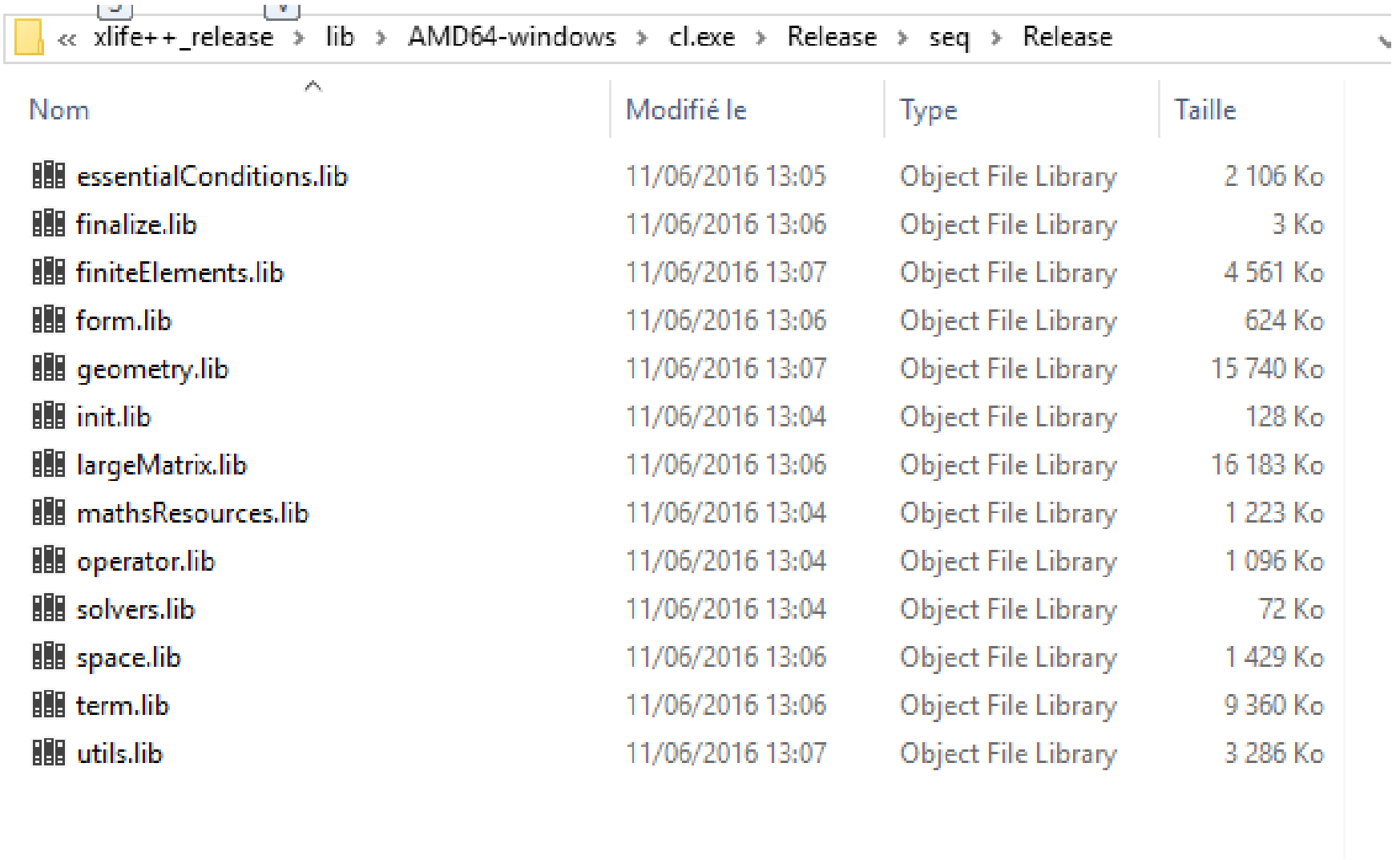


# Compile librairies

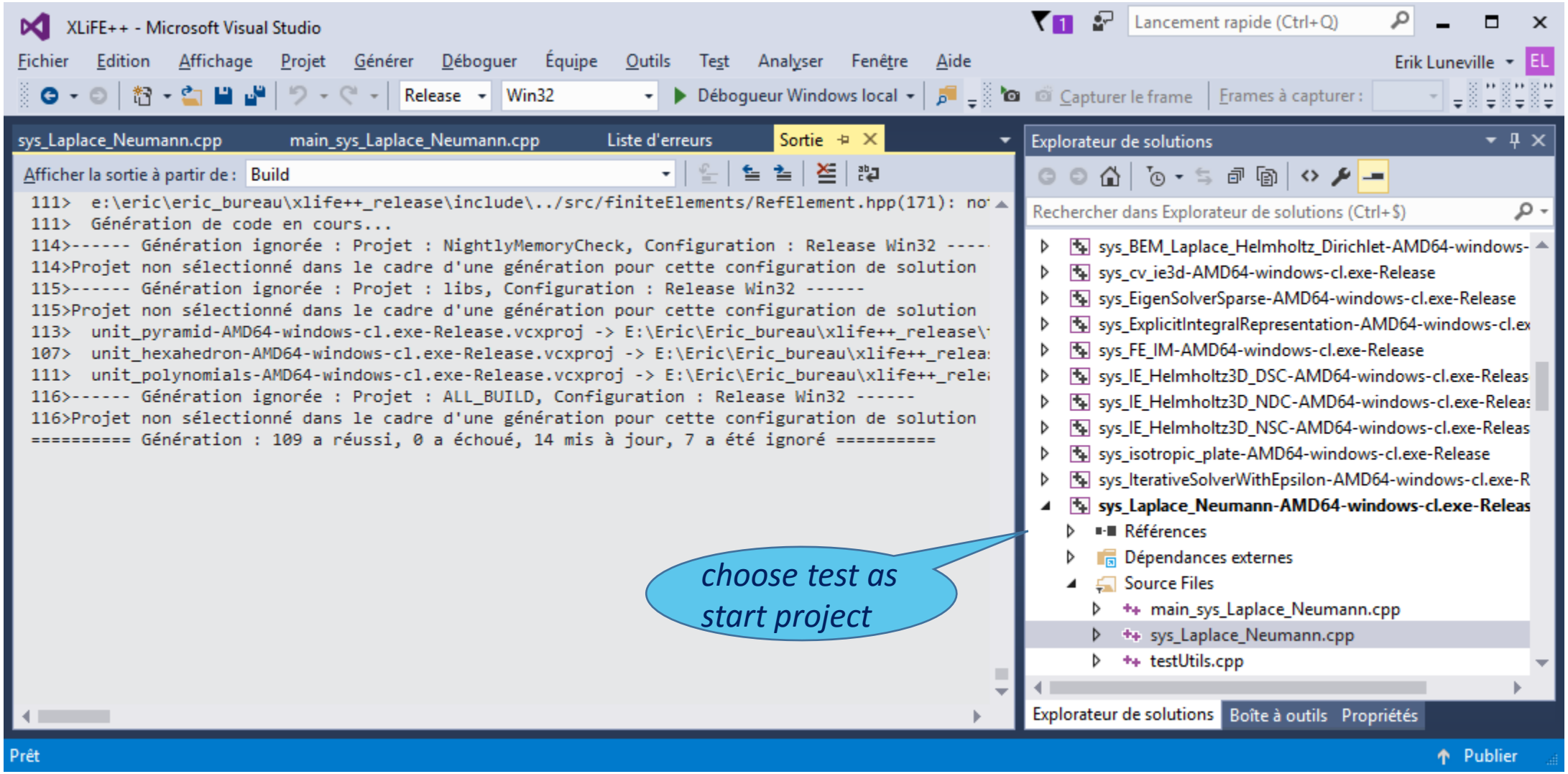



***don't worry about warnings (VS is very rigorous)***

# Libraries are created

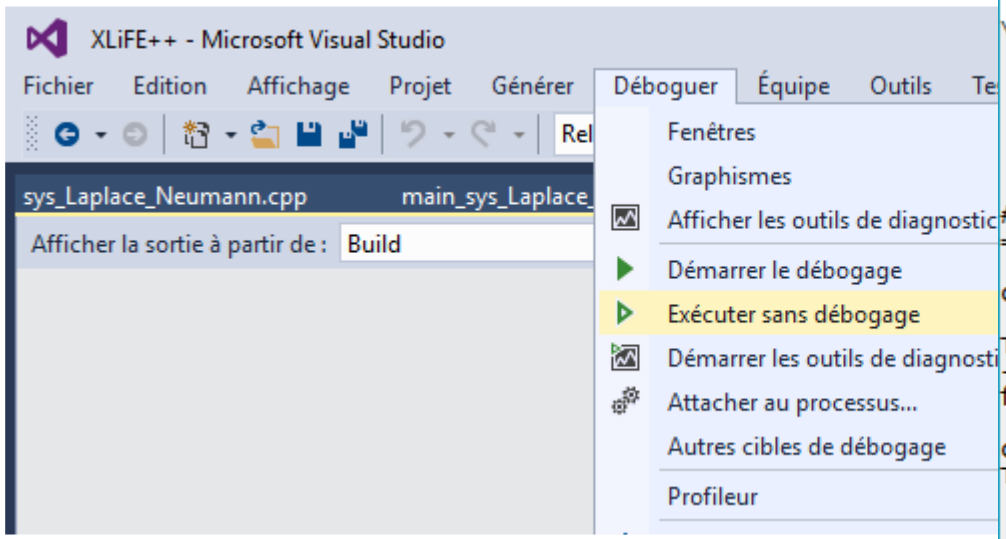


# Compile a test




***may take few minutes because template instantiations ...***

# Run test




**Ctrl+F5**

```
C:\WINDOWS\system32\cmd.exe
 XLiFE++ v1.4-r26 2016-06-10 running on june 11, 2016 at 13h32 on WIN_NT-6.1-i686 (NEWTON)

# TRIANGULAR mesh:  10 sudivisions, h = 0.02
========================================
 -> Triangles P2-------------------
computing FE term intg_Omega grad(u) | grad(v) + intg_Omega u * v, using 1 threads :  done

TermMatrix a(u,v) computed, size 9801 X 9801 :   SuTermMatrix a(u,v)_sub :   block (v, u)
->  matrix 9801 X 9801 of real scalar in symmetric_compressed sparse (csr,csc) (60516 coef
ficients)

computing FE term intg_Omega grad(u) | grad(v), using 1 threads :  done
TermMatrix R computed, size 9801 X 9801 :   SuTermMatrix R_sub :   block (v, u) ->  matrix
 9801 X 9801 of real scalar in symmetric_compressed sparse (csr,csc) (60516 coefficients)

computing FE term intg_Omega u * v, using 1 threads :  done
TermMatrix M computed, size 9801 X 9801 :   SuTermMatrix M_sub :   block (v, u) ->  matrix
 9801 X 9801 of real scalar in symmetric_compressed sparse (csr,csc) (60516 coefficients)

cpu time -> compute matrices : 0s.
 factorise matrix 9801 x 9801 using LDLt (skyline)
solving linear system a(u,v)_copy * X = f(v) (size 9801) cpu time -> solve system using LD
LT : 0.437s.
--------------- P2, h=0.02, nb dl=9801 ---------------
L2 Error = 8.44461e-09, Rel. L2 error = 3.50269e-07
H1 error = 3.24028e-06, Rel. H1 error = 2.95127e-05
C0 error = 1.55733e-07, Rel. C0 error = 3.22979e-06
 -> cpu time = 0
sys_Laplace_Neumann : results saved to E:/Eric/Eric_bureau/xlife++_release\tests\res\sys_L
aplace_Neumann.res
total cputime -> Total CPU time : 0.953s.
total chrono -> Total elapsed time : 1.078s.
Appuyez sur une touche pour continuer...
```

# Other Visual Studio builds

- **OpenMp** : **DO NOT ENABLE OMP**

  ***because Visual Studio is only compliant with OMP2 and XLiFE++ uses some OMP3 stuff***

- **arpack** : **if you want to load arpack (assuming you have somewhere blas, lapack, arpack lib)**

  **set the variables XLIFEPP_ARPACK_LIB_DIR, XLIFEPP_BLAS_LIB_DIR, XLIFEPP_LAPACK_LIB_DIR**

  **before the second configure step**

  ***it may fails if libraries are not compliant***

- **suitesparse : if you want to load suitesparse**

  **assuming you have install the suitesparse distribution and compile libraries**

  **set the variable XLIFEPP_SUITESPARSE_HOME_DIR**

  ***it may fails if libraries are not compliant***

- **Visual Studio 64 bits : previous install concerns Visual Studio in 32 bits**

  **It is possible to choose VS 64 bits when choosing the generator**

  ***it may happen some trouble with the mspd140.dll***

# Compile libraries

- **OpenMp** : **DO NOT ENABLE OMP**

  ***because Visual Studio is only compliant with OMP2 and XLiFE++ uses some OMP3 stuff***

- **arpack** : **if you want to load arpack (assuming you have somewhere blas, lapack, arpack lib)**

  **set the variables XLIFEPP_ARPACK_LIB_DIR, XLIFEPP_BLAS_LIB_DIR, XLIFEPP_LAPACK_LIB_DIR**

  **before the second configure step**

  ***it may fails if libraries are not compliant***

- **suitesparse : if you want to load suitesparse**

  **assuming you have install the suitesparse distribution and compile libraries**

  **set the variable XLIFEPP_SUITESPARSE_HOME_DIR**

  ***it may fails if libraries are not compliant***

- **Visual Studio 64 bits : previous install concerns Visual Studio in 32 bits**

  **It is possible to choose VS 64 bits when choosing the generator**

  ***it may happen some trouble with the mspd140.dll***

# Create new XLiFE++ application using Visual Studio on Windows,

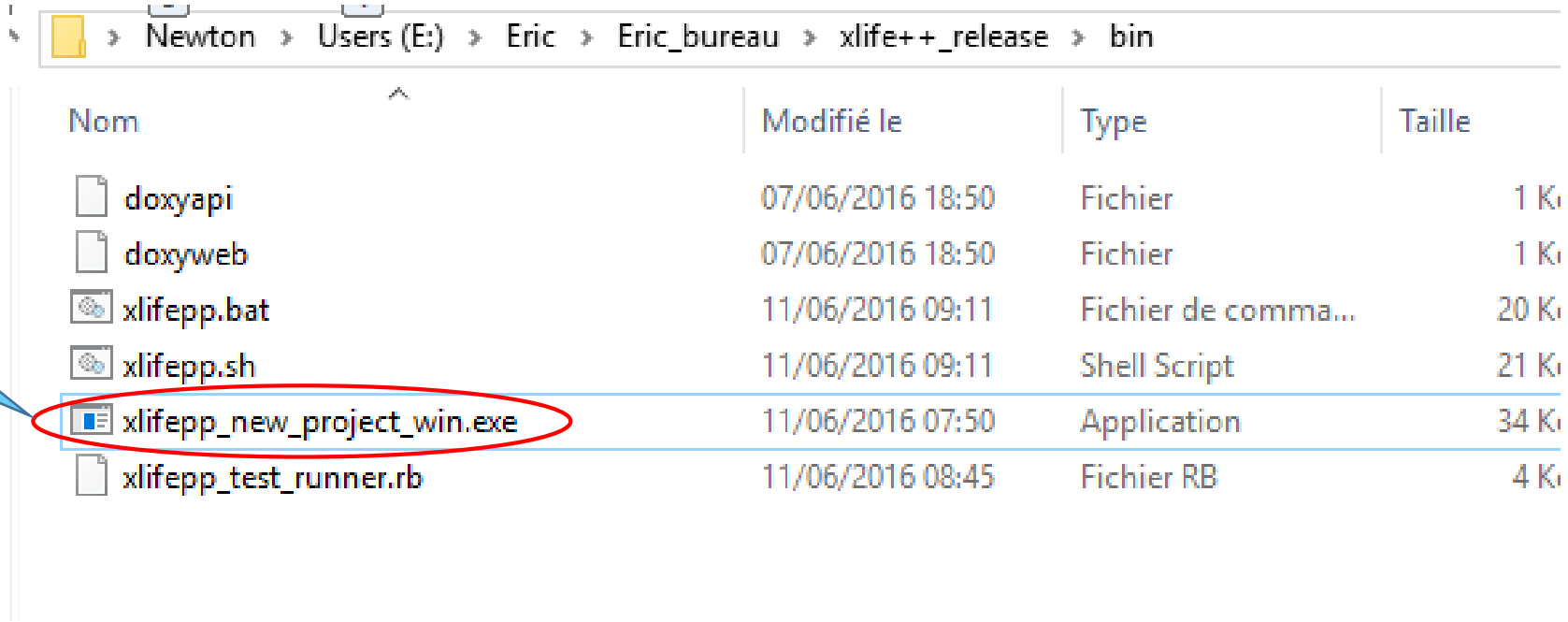
**Eric Lunéville**

# Requirements

- XLiFE++ source installed
- XLifE++ libraries built from Visual Studio generator
- XLifE++ libraries compiled with Visual Studio

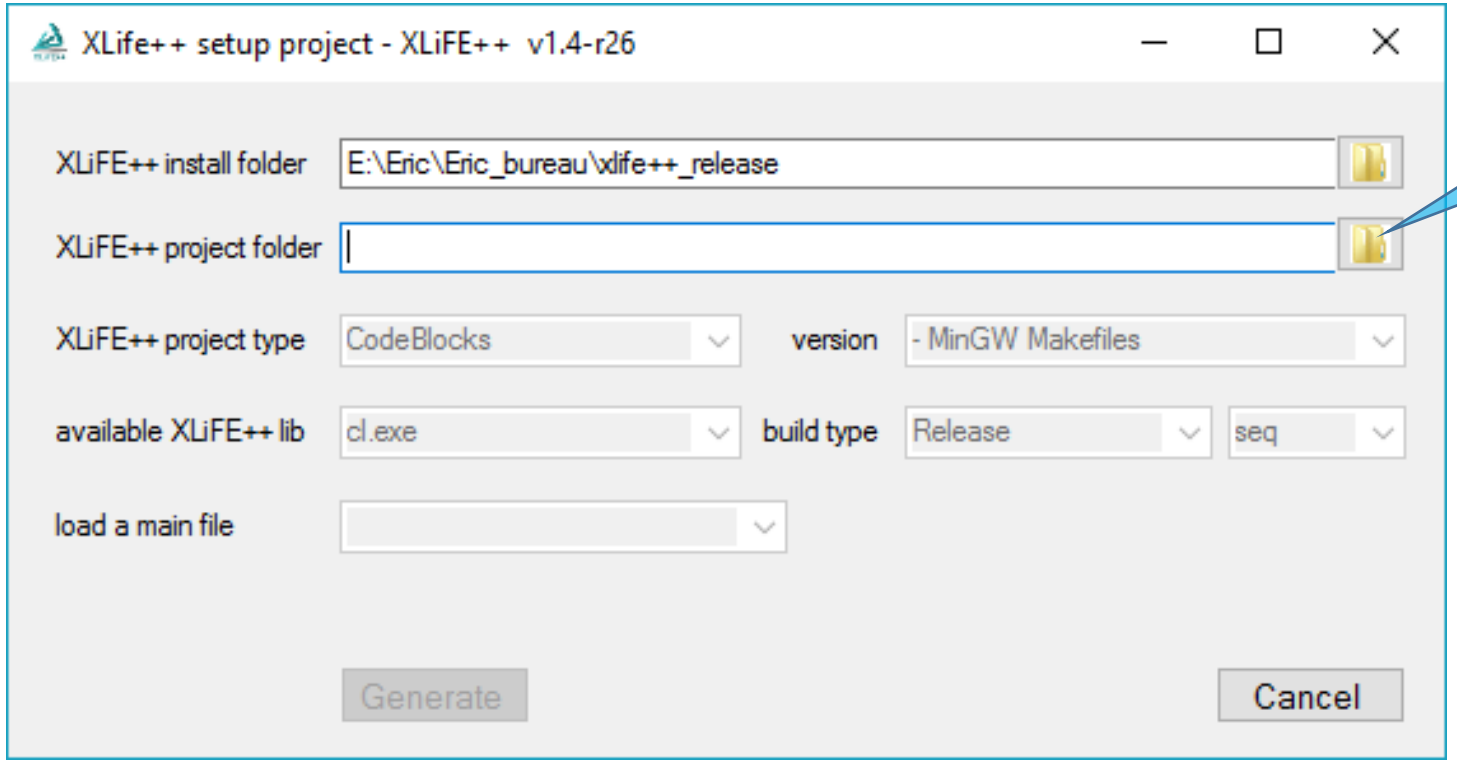
# Run xlifepp_setup_win_project

*double click on*

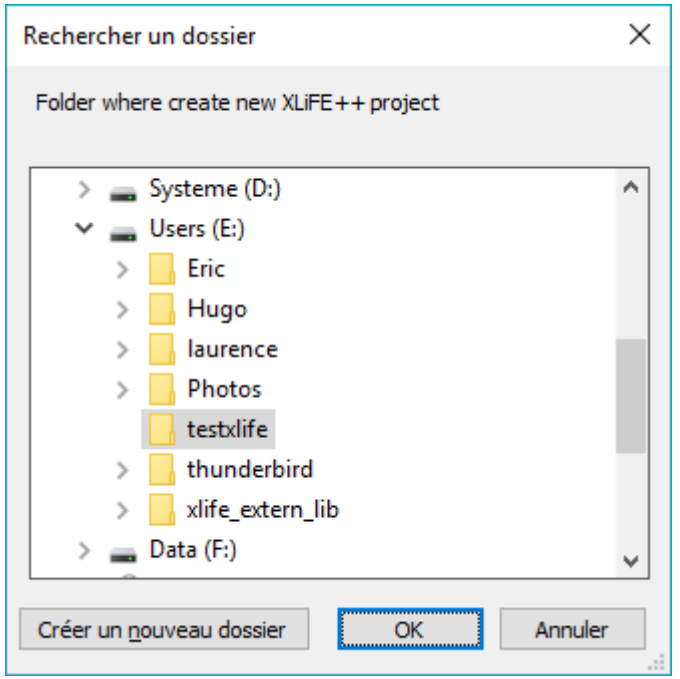


# Run xlifepp_setup_win_project

***XLifE++ install folder name should be up to date, if not set it yourself***



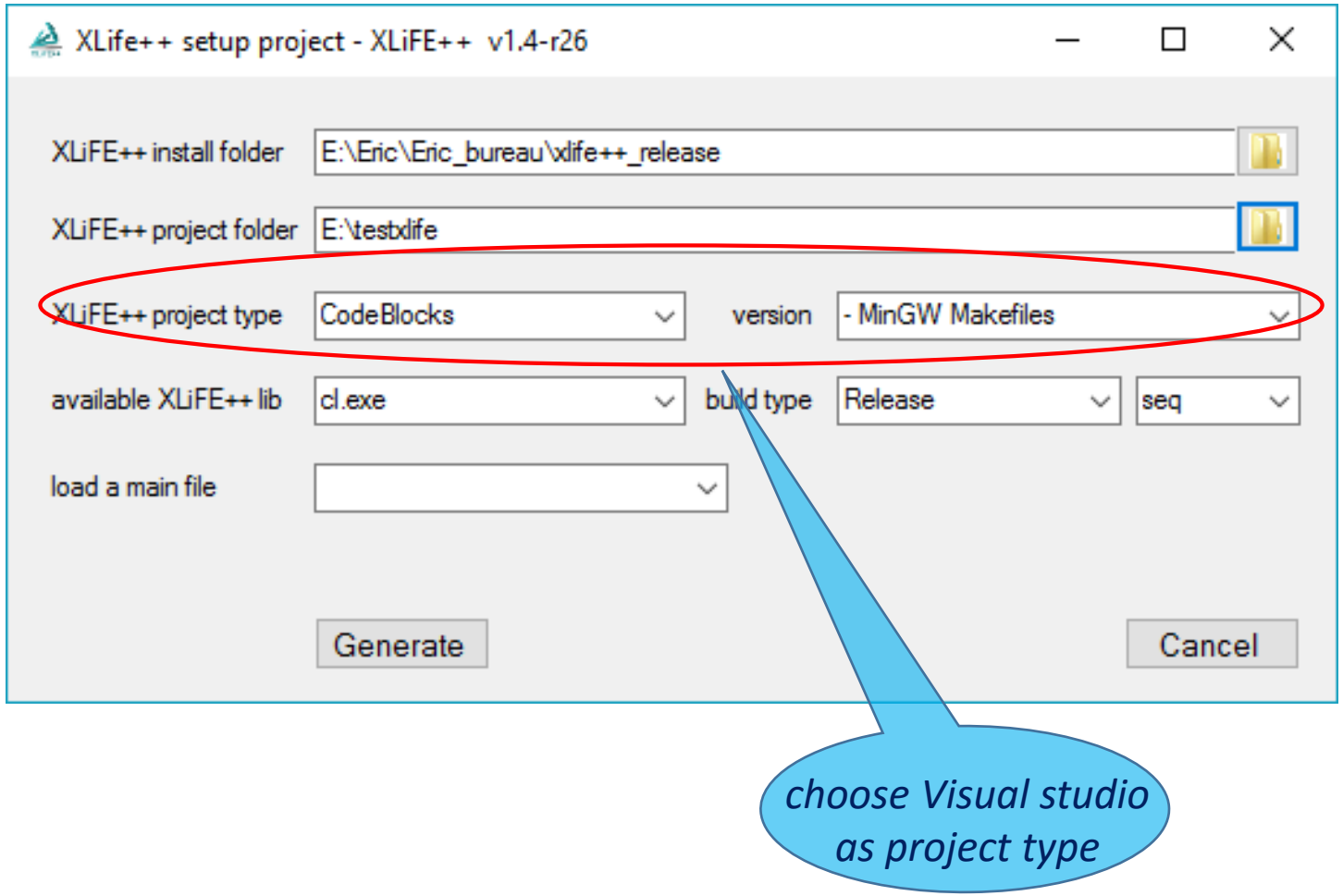

*choose folder where create new application*



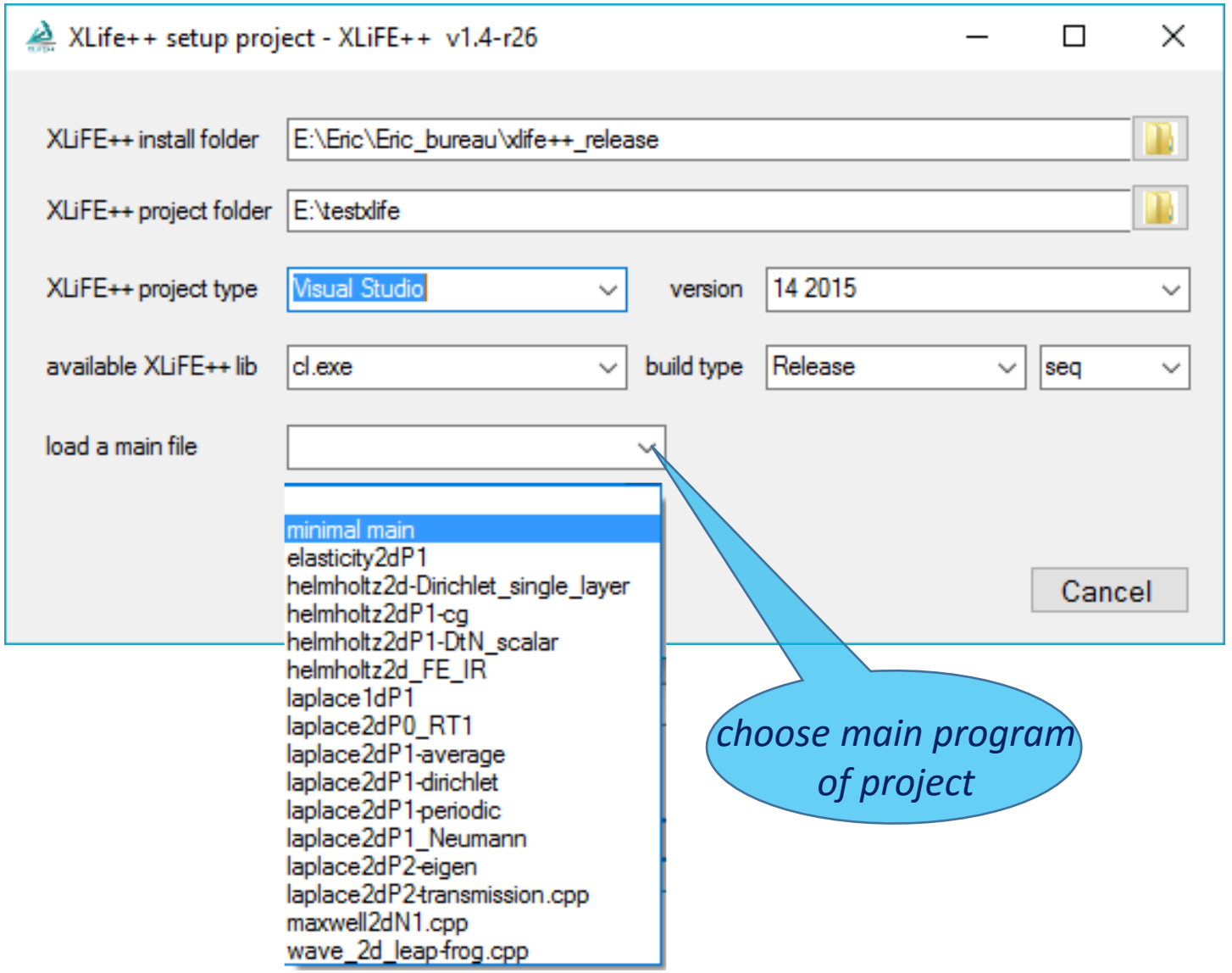

***XLiFE++ project folder does not exist, you can create it***

***if XLiFE++ project folder exists and contains some old files, it is recommanded to clean it***

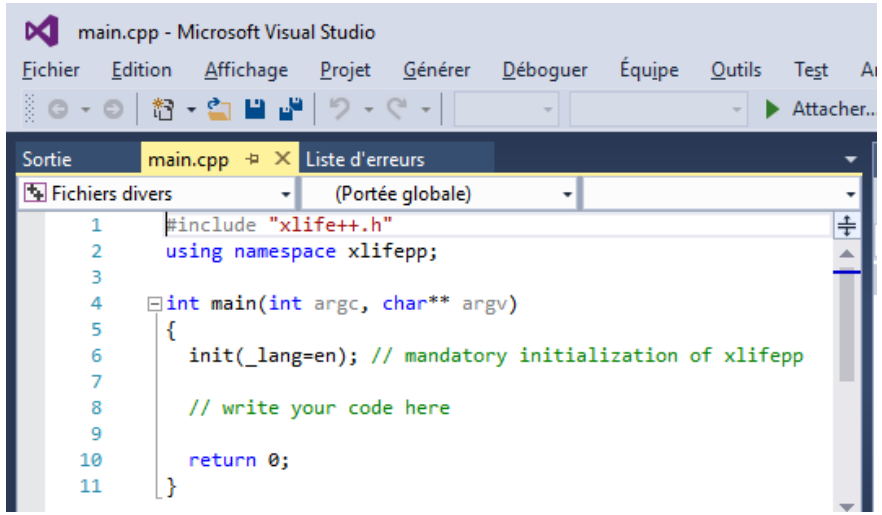
# Run xlifepp_setup_win_project



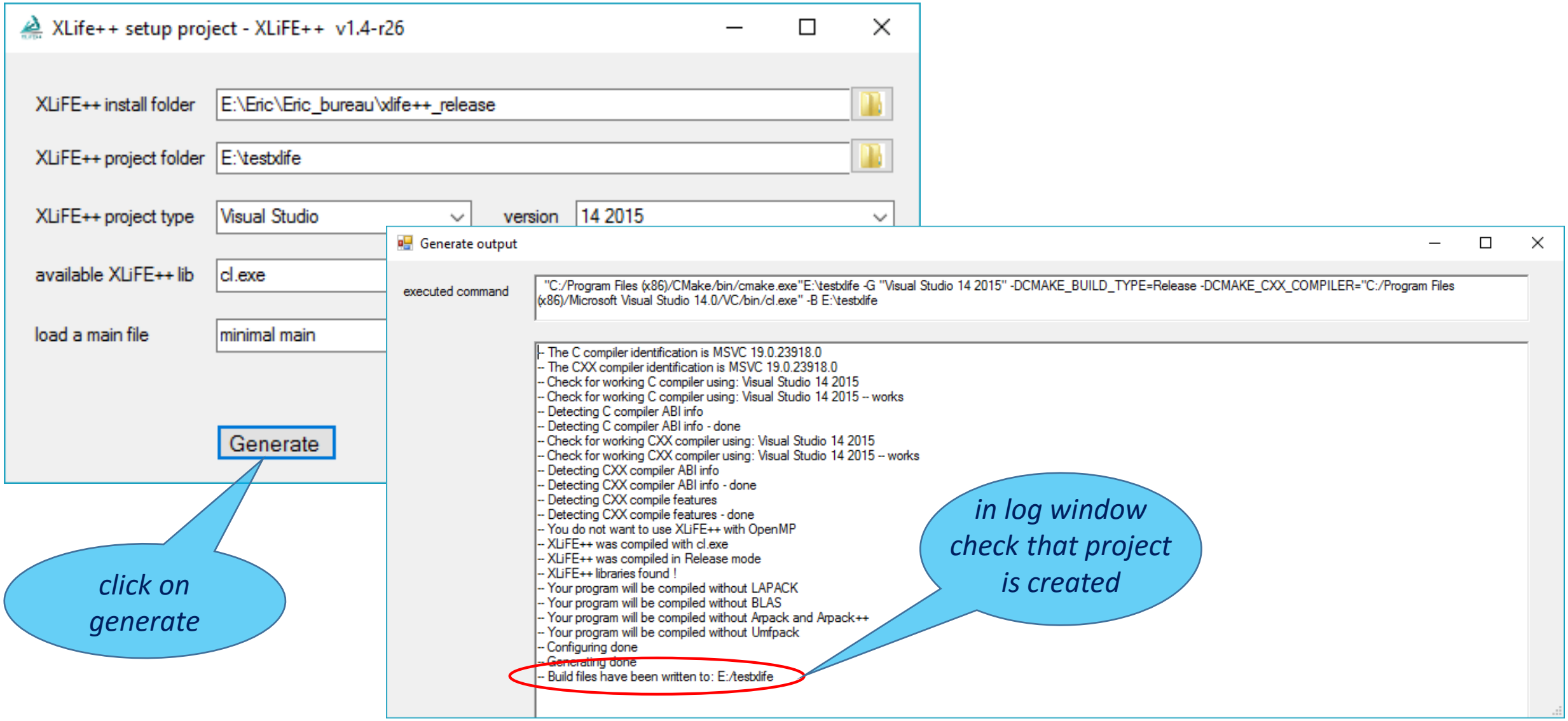
# Run xlifepp_setup_win_project




**minimal main**



# Run xlifepp_setup_win_project



# Run Visual Studio

**close all windows of setup project and goto project folder**