Guide to Qt Creator

Jesper Roland Kjærgaard Qwist

Created 2019.10.03
Version 2019.10.04

Contents

1 Information

This tutorial is prepared on a dual-boot machine with Ubuntu 18.04.

Qt Creator can be downloaded and installed from the terminal with the command (REMEMBER TO CHECK THAT YOU HAVE ENOUGH SPACE!):

```
$: sudo apt-get install qtcreator
```

Qt Creator is opened through **activities** and not from the terminal. It can be quite annoying to accidently close Qt Creator when yo close a terminal.

Currently it does not work for me to use environment variables.

It is possible to compile applications and run cases from Qt Creator, but it is not covered in this document, since I prefer to do these operations in a terminal window. Furthermore I find the setup of this time consuming, and it did not make my life easier.

I have no or very limited experience with version control and debugging with gdb in Qt Creator.

2 Set up Qt Creator Example

Open a new terminal window (CTRL+ALT+T). Source OpenFOAM and go to user solvers:

```
OFv1906
sol
```

Now we will make a copy of the existing solver icoFoam and rename it qtIcoFoam:

```
cp -r $FOAM_SOLVERS/incompressible/icoFoam $WM_PROJECT_USER_DIR/applications/solvers/i
cd $WM_PROJECT_USER_DIR/applications/solvers/incompressible
mv icoFoam qtIcoFoam
cd qtIcoFoam
mv icoFoam.C qtIcoFoam.C
sed -i s/icoFoam/qtIcoFoam/g Make/files
sed -i s/FOAM_APPBIN/FOAM_USER_APPBIN/g Make/files
```

Compile the application:

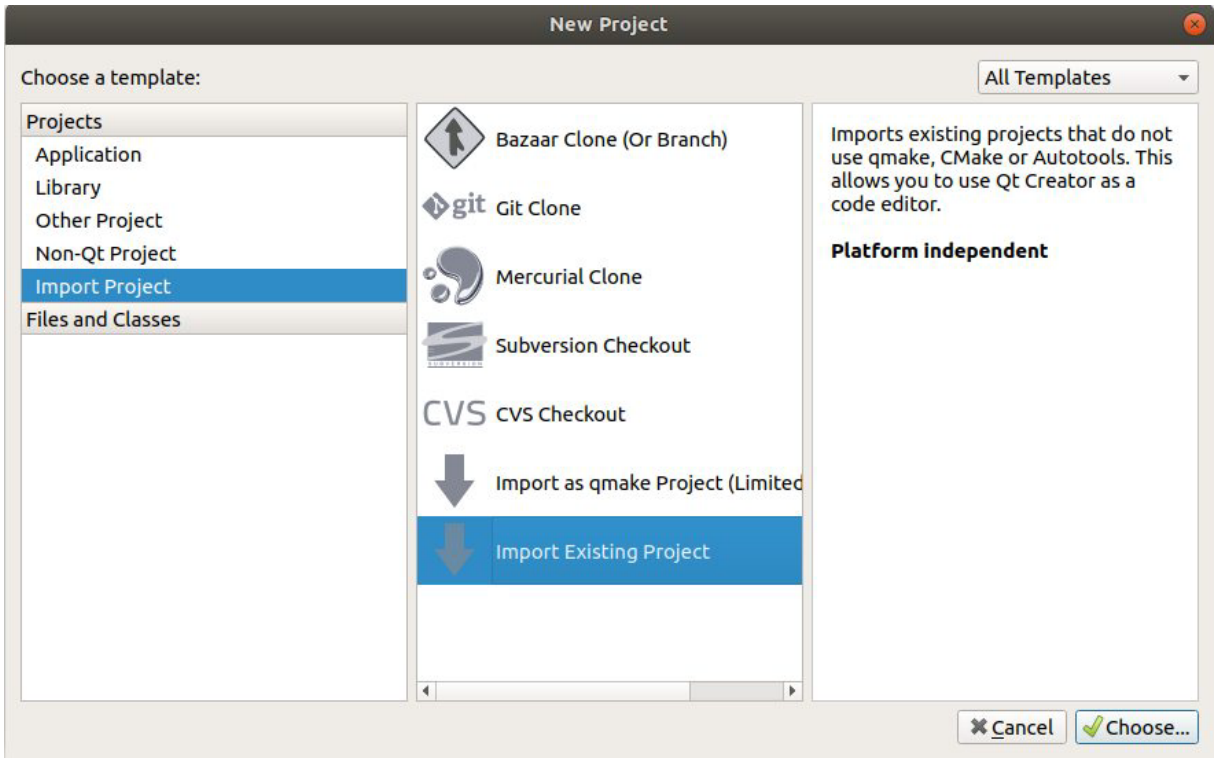
```
wmake
```

Now open Qt Creator running in the background:

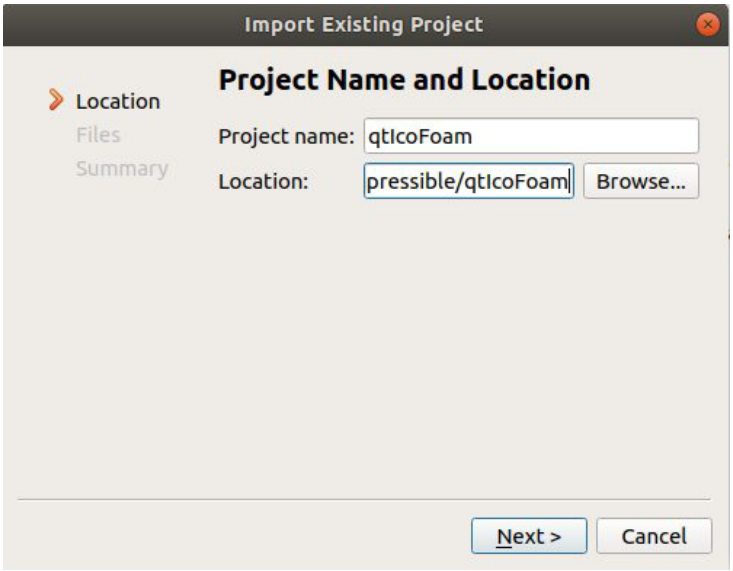
```
qtcreater &
```

Then choose "New Project":

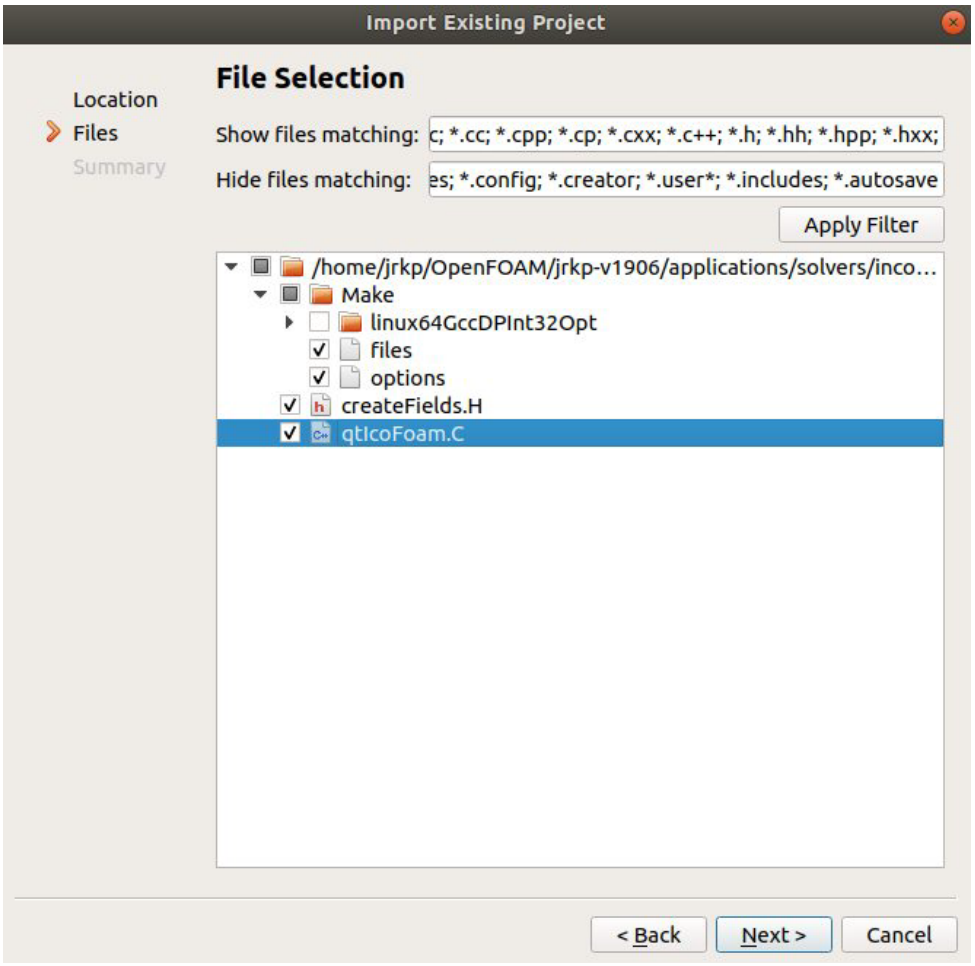
Mark "Import Project" and "Import Existing Project". Then press the button "Choose" to continue.



Now the window "Import Existing Project" opens, and you should supply the project name, which should be the same as the application. The location is the full path to the application folder. When you are finished press "Next".

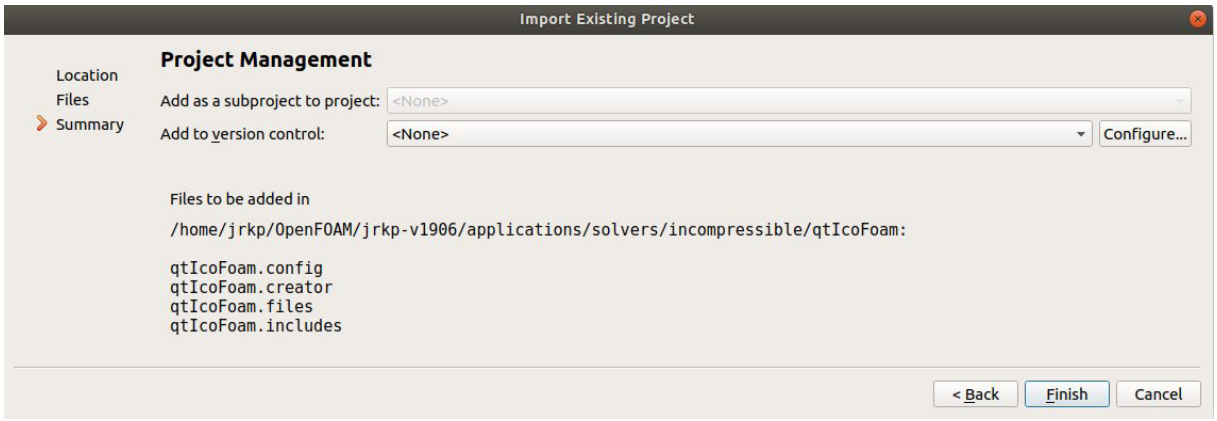


The window "File Selection" opens:



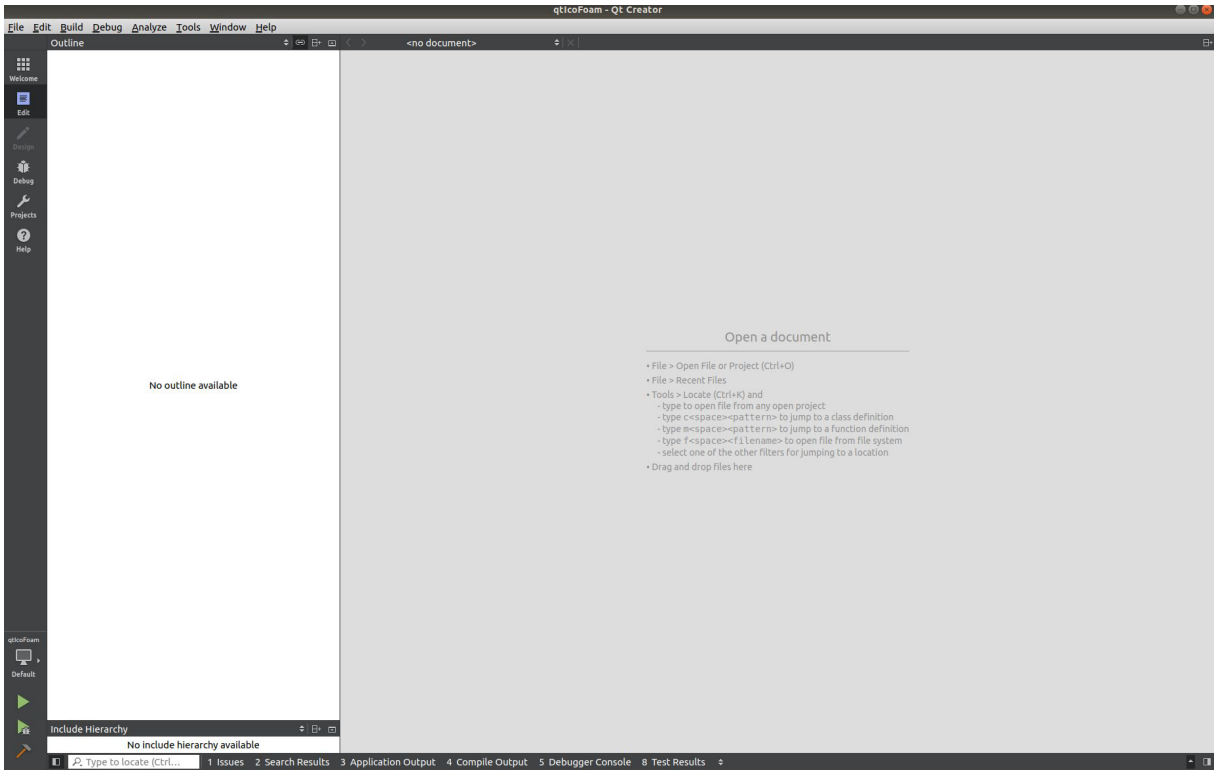
Here you can expand on the arrows to get a view of the chosen folder. Choose the files that you want to include in the project. If you experience that you cannot find any files in the folder, try to add *.C;*.H on the line which says "Show files matching". When you are finished press "Next".

Now the "Project Management Window" opens:

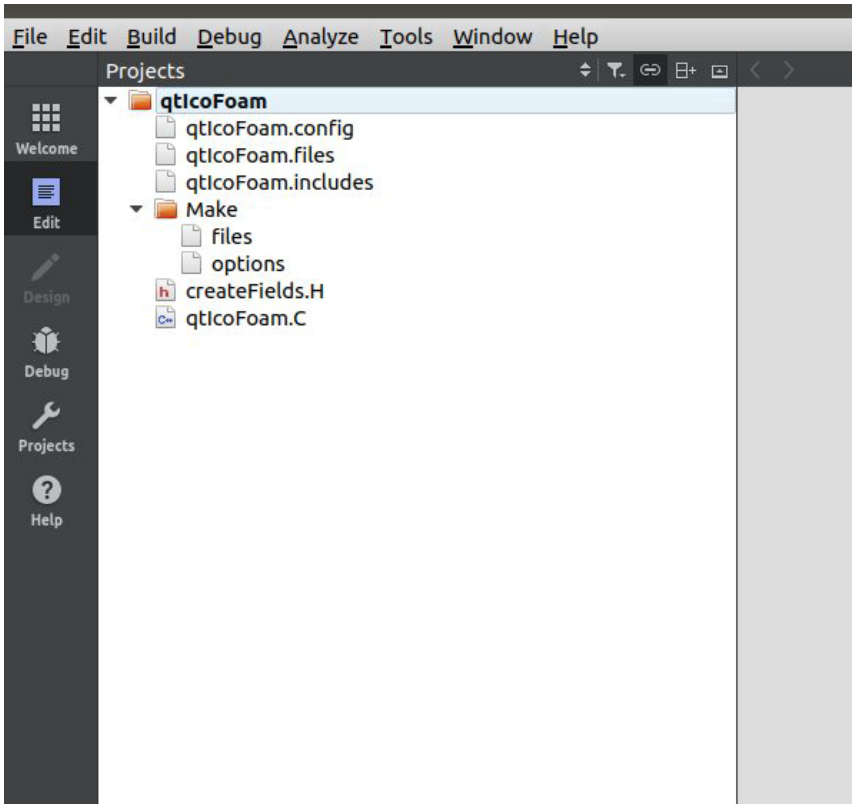


Here you can verify that you have chosen the correct path to your application, and the program tells you which files it is going to create. You also have the option to use a Git version control system, if you are using that for your files. I have not tried this myself, but please feel free to try it out.

Now Qt Creator will open the following window:



Go to the upper left corner and open the frop-down menu which says "Outline". Change to "Project", and expand the project on the small arrows.



Now we need to tell Qt Creator where to look for header files and shared object libraries (.so). The required includes and shared libraries for the solver is specified in the options file under the Make folder:

```
// Open file with solver options
gedit Make/options &

// Copy from options file:
    -I$(LIB_SRC)/finiteVolume/lnInclude \
    -I$(LIB_SRC)/meshTools/lnInclude

// Insert in qtIcoFoam.includes

// Substitute:
"    -I$(LIB_SRC)"

// with the result from the command:
"echo $FOAM_SRC"

// Remove backslashes at the end of each line.

// Furthermore add to qtIcoFoam.includes above the other two
// lines:
$FOAM_SRC/OpenFOAM/lnInclude

// The end result in qtIcoFoam.includes is:
.
$FOAM_SRC/OpenFOAM/lnInclude
$FOAM_SRC/finiteVolume/lnInclude
$FOAM_SRC/meshTools/lnInclude

// Remember that you should substitute the environment variable with
// the actual path.

// Now copy from Make/options to qtIcoFoam.files
    -lfiniteVolume \
    -lfvOptions \
    -lmeshTools

// Remove backslash at end of line and replace "l" with "lib" as follows:
-libfiniteVolume
-libfvOptions
-libmeshTools

// The shared object libraries are found in
echo $FOAM_LIBBIN

// The path is in my case:
/home/jrkp/OpenFOAM/OpenFOAM-v1906/platforms/linux64GccDPInt32Opt/lib
```

```
// Now I replace "-" with the path saved in environment variable
// FOAM_LIBBIN and add the (.so) extension in the file qtIcoFoam.files:
$FOAM_LIBBIN/libfiniteVolume.so
$FOAM_LIBBIN/libfvOptions.so
$FOAM_LIBBIN/libmeshTools.so
```

Now we can open "qtIcoFoam.C" and verify that there are no longer any curly lines below any of the code. If the curly lines are still there, then try to close qt creator, and open it through activities in ubuntu. I have experienced problems when I open it directly from the terminal.

3 Commands and drop-down menu

The most common commands are:

```
Auto-completion/show options: CTRL-SPACE
Follow symbol under cursor
(Open declaration file/find object construction): F2
Switch from header/source and vice versa: F4
```

You can get a menu with more commands by right-clicking in any open file in Qt Creator.

In the drop-down menu where it should currently say "Projects", we can also choose:

Open Documents:
Here you can see which documents, that you have opened.

File system:
Here you can see the location of the file open in your browser in the file system. Can be used to find similar files quickly. For example find one boundary condition and you can easlily get an overview of the other options.

Outline:
This will give you an outline of the class. You can double-click on each entry and Qt Creator will place the cursor at this location in the file. Very convienient to navigate very large class definition source files.

Include Hierarchy:
Here you can get a full overview of the included headers of other classes by the class, and which other classes that includes the current class.