



# TUTO INSTALLATION de SFML dans Qt Creator



- 1) Télécharger la SFML [ici](http://www.sfm1-dev.org/). Choisissez la dernière version disponible en prenant bien la version « Tous OS » : « Code source ».

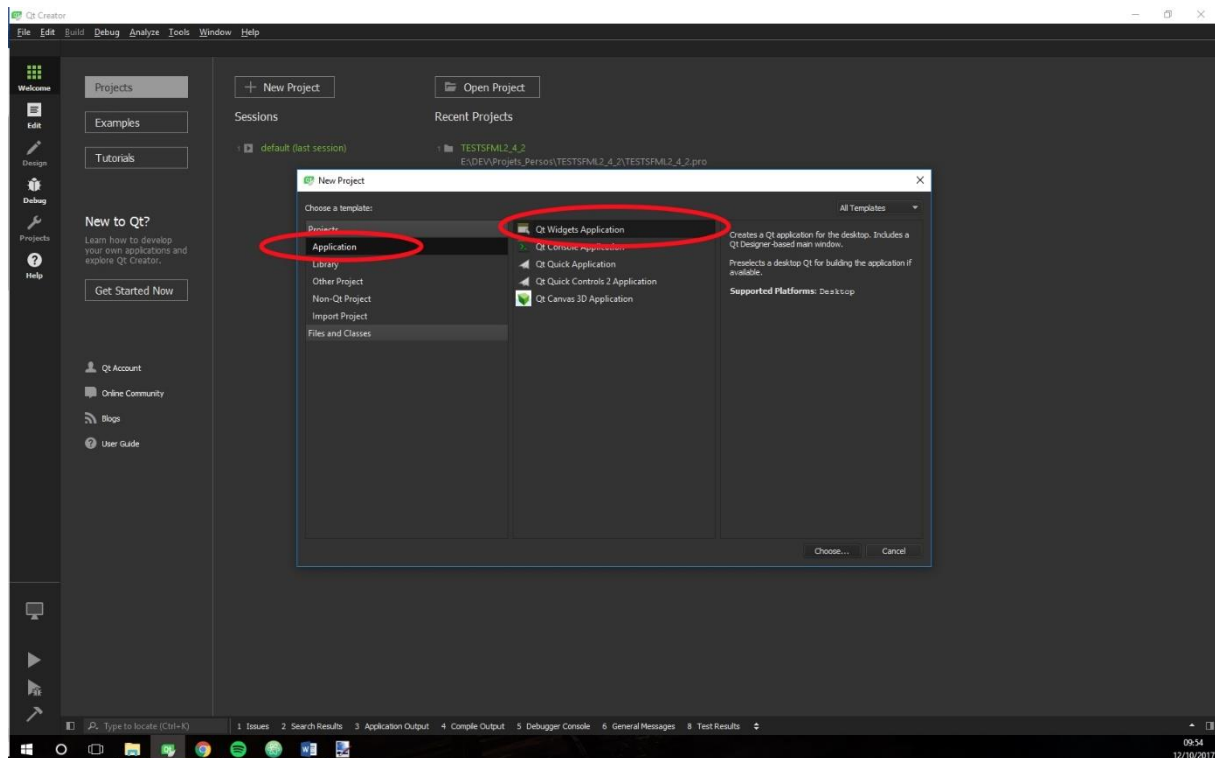
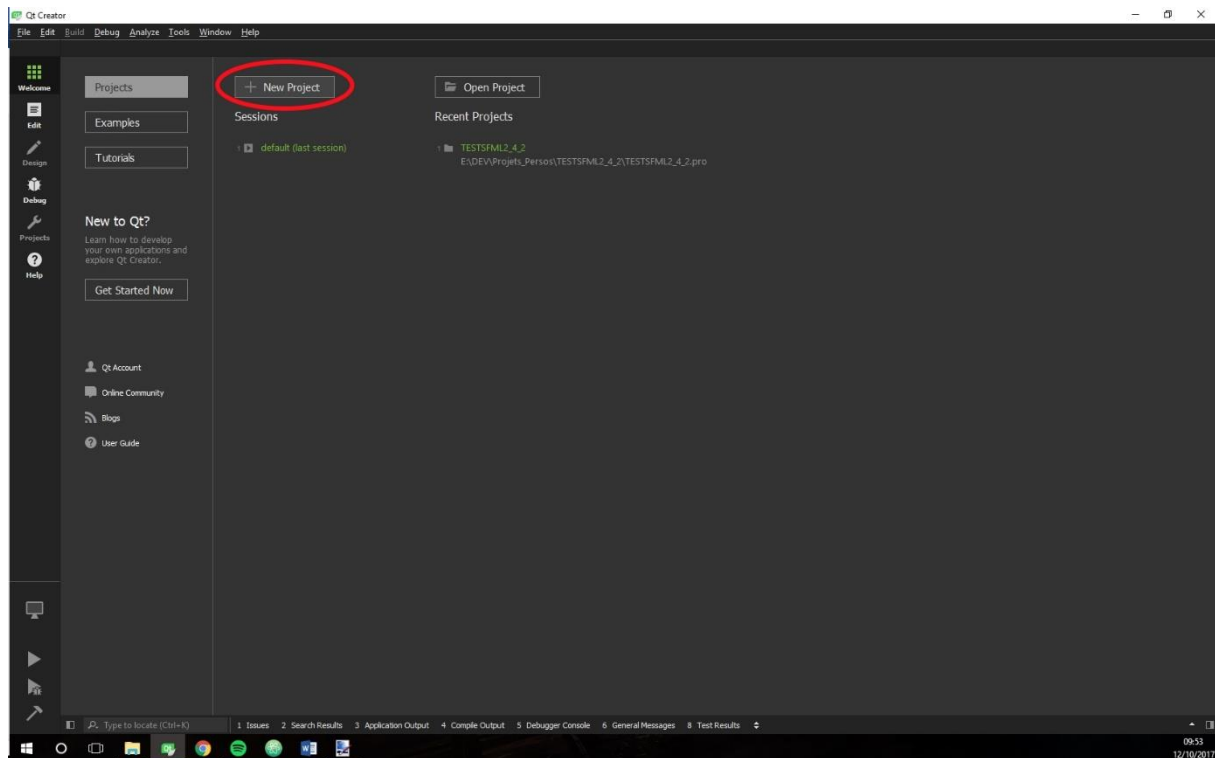
The screenshot shows the SFML 2.4.2 download page. The page is organized into sections for different operating systems: Windows, Linux, Mac OS X, and Tous OS. Each section lists available download packages with their respective sizes. The 'Tous OS' section is highlighted with a red circle, showing two download links: 'Code source' (12.8 Mo) and 'Documentation HTML' (1.48 Mo).

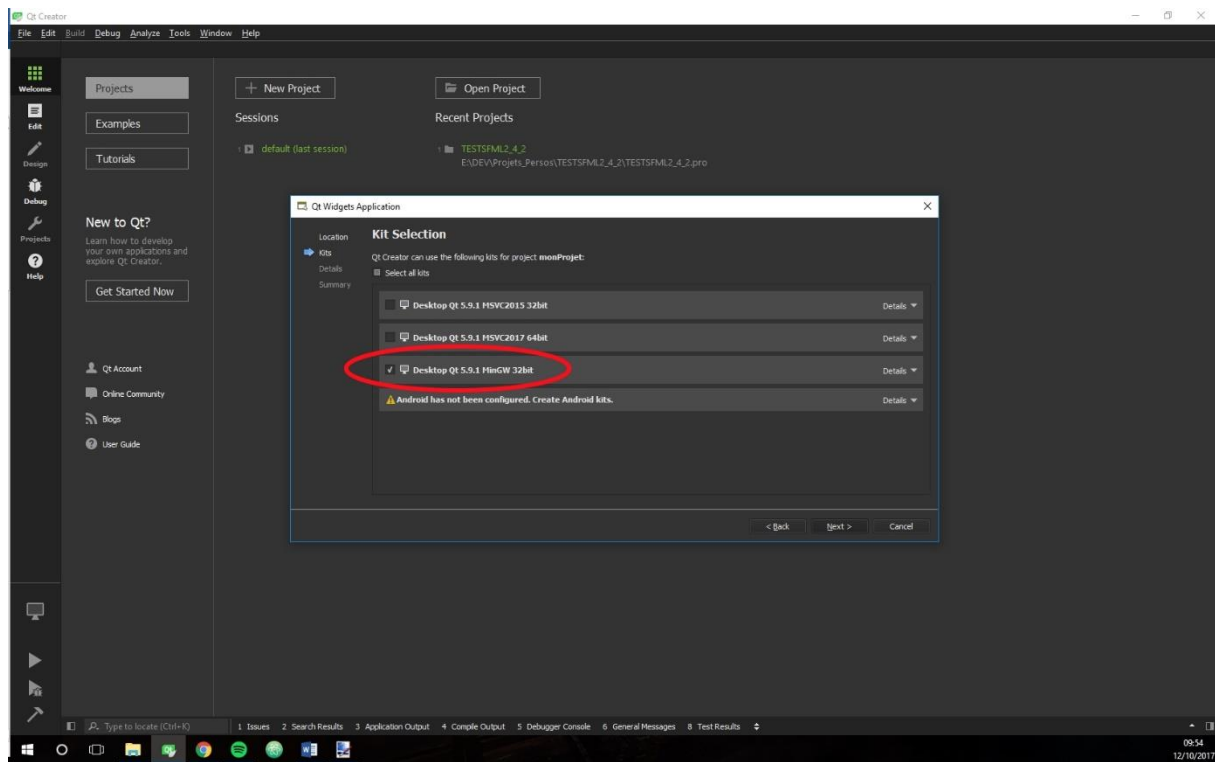
OS	Compiler/Package	Download Link	Size
Windows	Visual C++ 12 (2013) - 32-bit	Télécharger	16.1 Mo
	Visual C++ 12 (2013) - 64-bit	Télécharger	17.8 Mo
	Visual C++ 14 (2015) - 32-bit	Télécharger	16.0 Mo
	Visual C++ 14 (2015) - 64-bit	Télécharger	17.6 Mo
	GCC 4.9.2 TDM (SJLJ) - 32-bit	Télécharger	13.8 Mo
	GCC 4.9.2 TDM (SJLJ) - 64-bit	Télécharger	15.8 Mo
Linux	GCC 6.1.0 MinGW (DW2) - 32-bit	Télécharger	15.3 Mo
	GCC 6.1.0 MinGW (SEH) - 64-bit	Télécharger	16.2 Mo
Mac OS X	Clang - 64-bit (OS X 10.7+, compatible C++11 et libc++)	Télécharger	5.34 Mo
	Documentation HTML	Télécharger	1.48 Mo
Tous OS	Code source	Télécharger	12.8 Mo
	Documentation HTML	Télécharger	1.48 Mo

- 2) Dézippez le fichier téléchargé dans un dossier de votre convenance mais de préférence sur votre disque dur local.

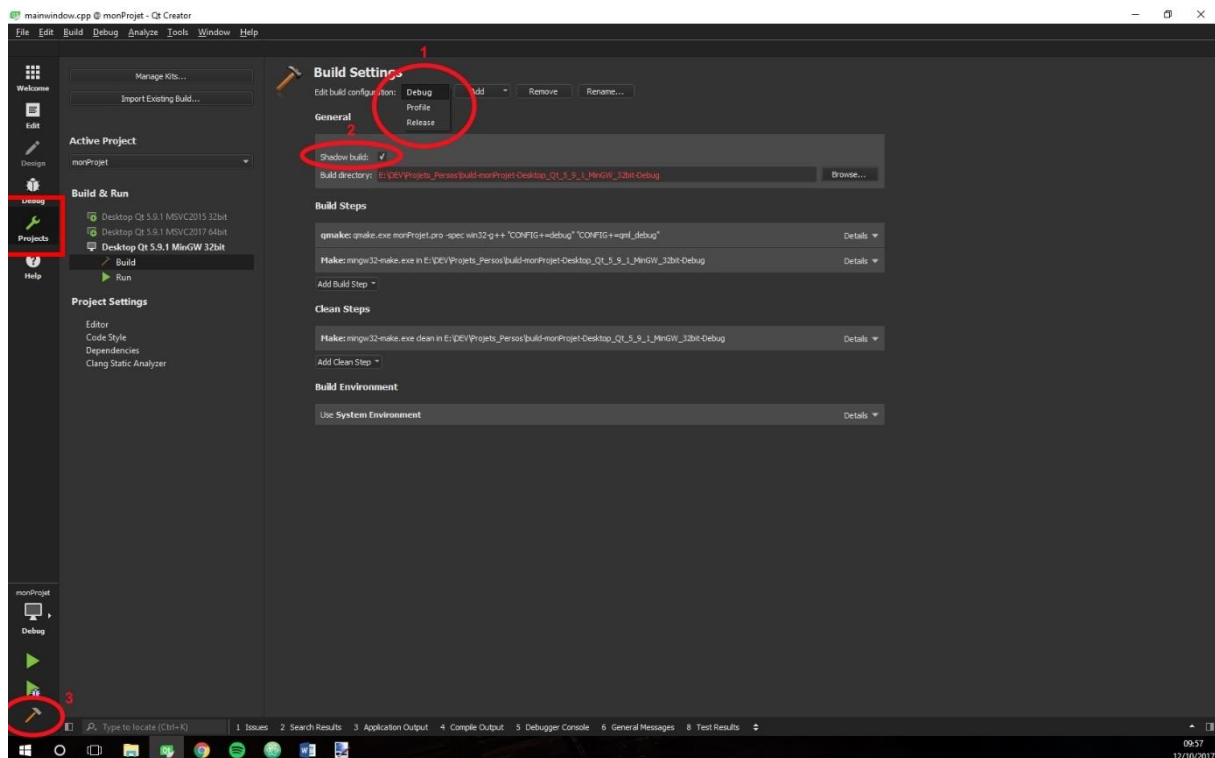
The screenshot shows the Qt Creator IDE interface. The 'File Explorer' panel on the left shows the 'Qt\_Libraries' folder selected. The 'Contents' pane on the right shows the files and folders within 'Qt\_Libraries'. The 'SFML-2.4.2' folder is highlighted with a red circle, indicating it is the target of the installation process.

3) Ouvrez Qt Creator et créez un nouveau projet, pour ce tuto je le nommerai « monProjet », en cliquant sur « New project ». monProjet pour ce tuto est de type Qt Widgets Application. Réglez les paramètres suivants selon vos besoins mais veillez surtout à bien choisir le compilateur MinGW (ici en version 32 bits) car MSVC génère des .lib au lieu des .a et cela pose problème plus tard dans l'installation de la bibliothèque. Finissez la création de votre projet.

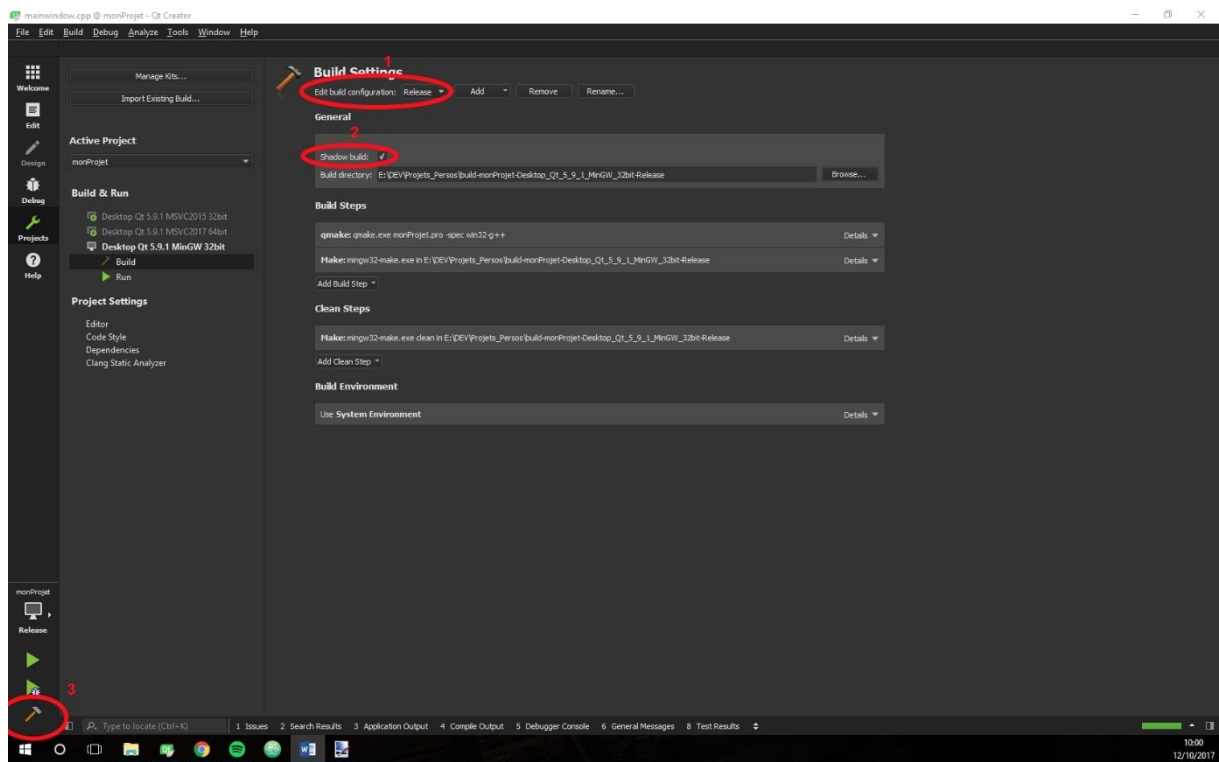




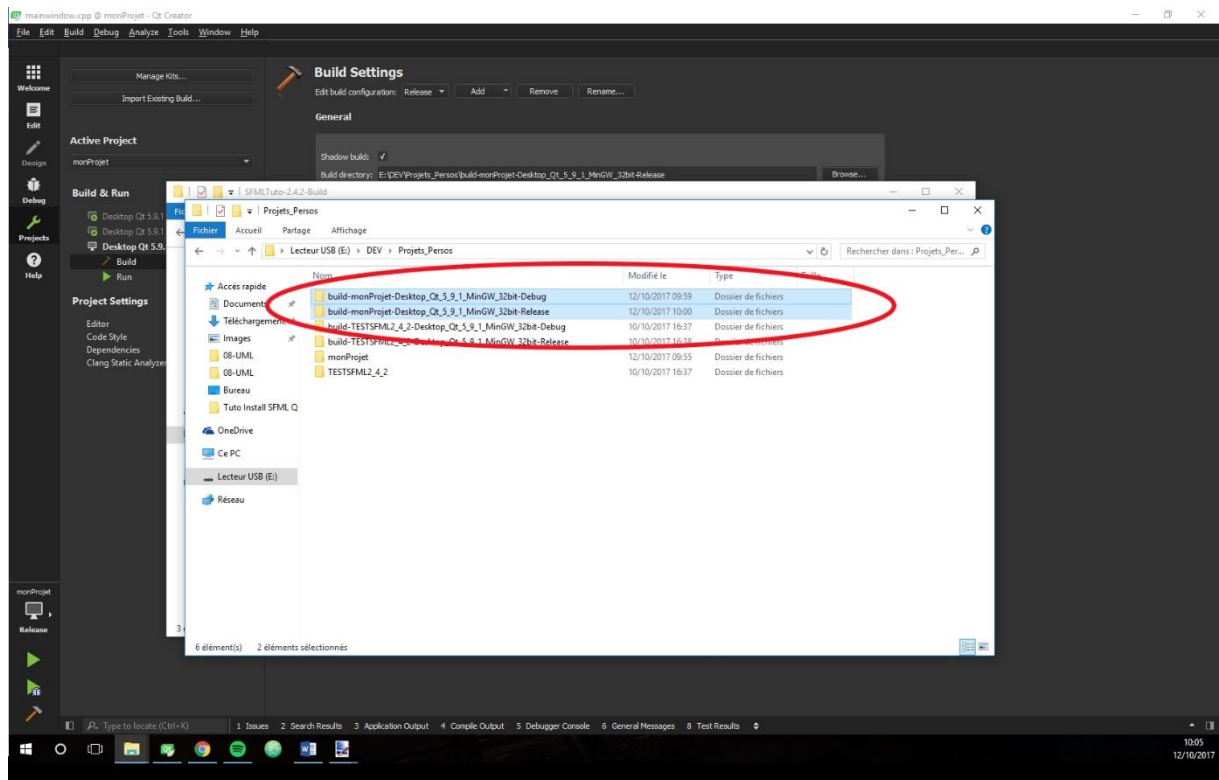
4) Dirigez-vous dans l'onglet « Projects » de Qt, sélectionnez « Debug » en haut en type de build et vérifiez que le Shadow Build est bien coché. Très important. Lancez un build. Refaire la même manipulation mais en choisissant un type de build « Release ». Lancez un build.



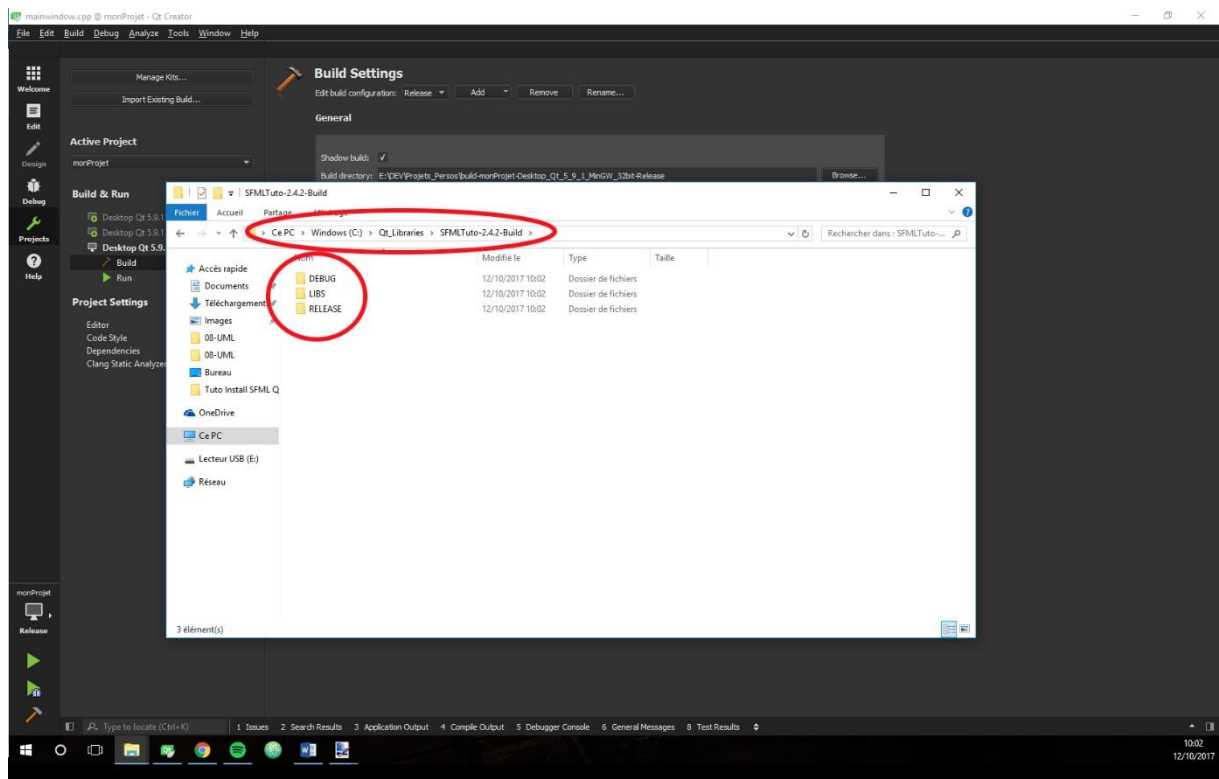
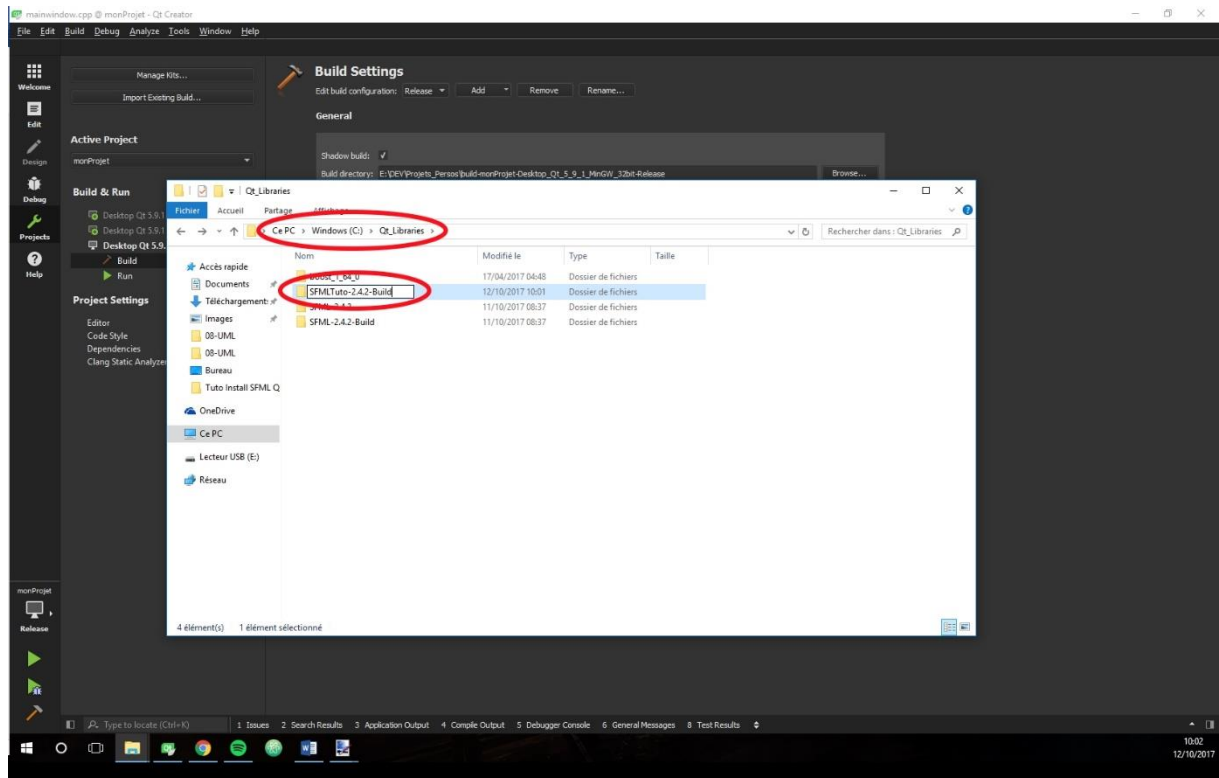
ATTENTION ! Ici vous n'êtes pas obligé de rester en « Shadow Build ». Vous pouvez tout à fait décocher les deux « Shadow Build ». Cependant, si vous faites ce choix pensez bien à préciser les chemins de destination et de source vers les dossiers « debug » et « release » qui seront créés dans votre dossier de projet. Lors de la copie des dlls (étape 9) pensez bien à les placer dans les dossiers « debug » et « release » de votre dossier de projet.



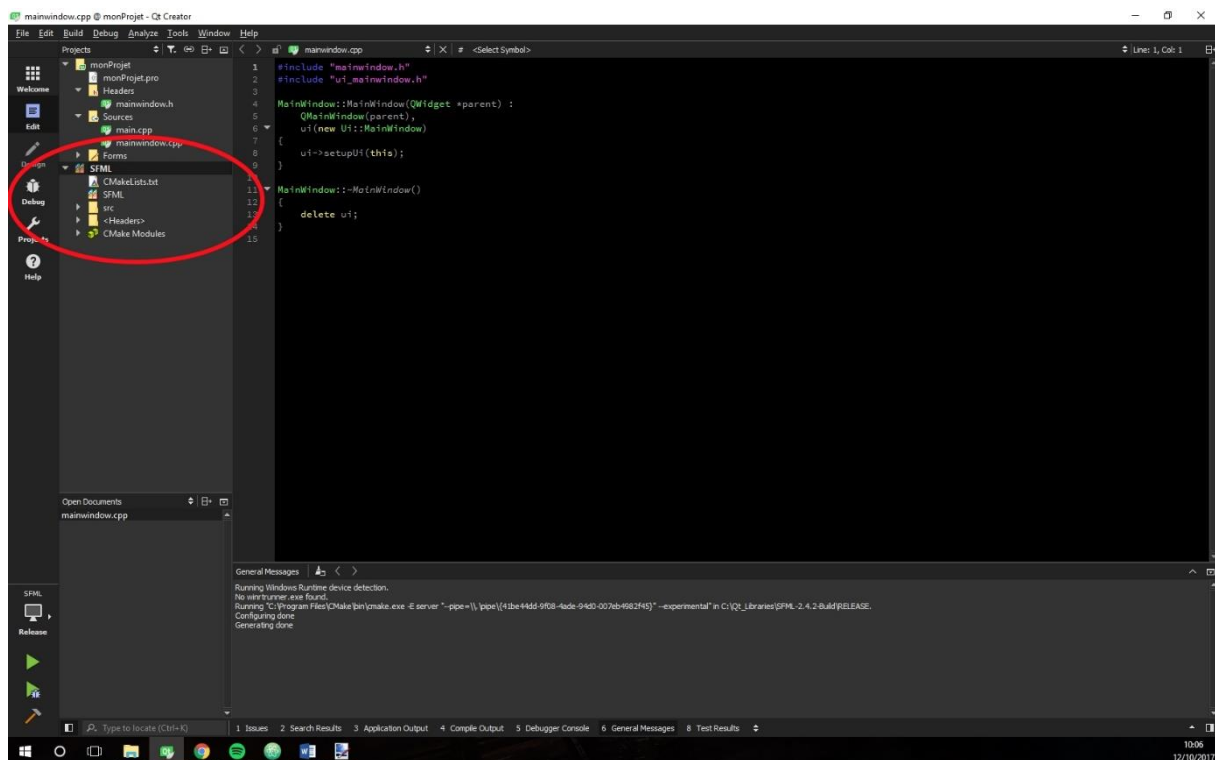
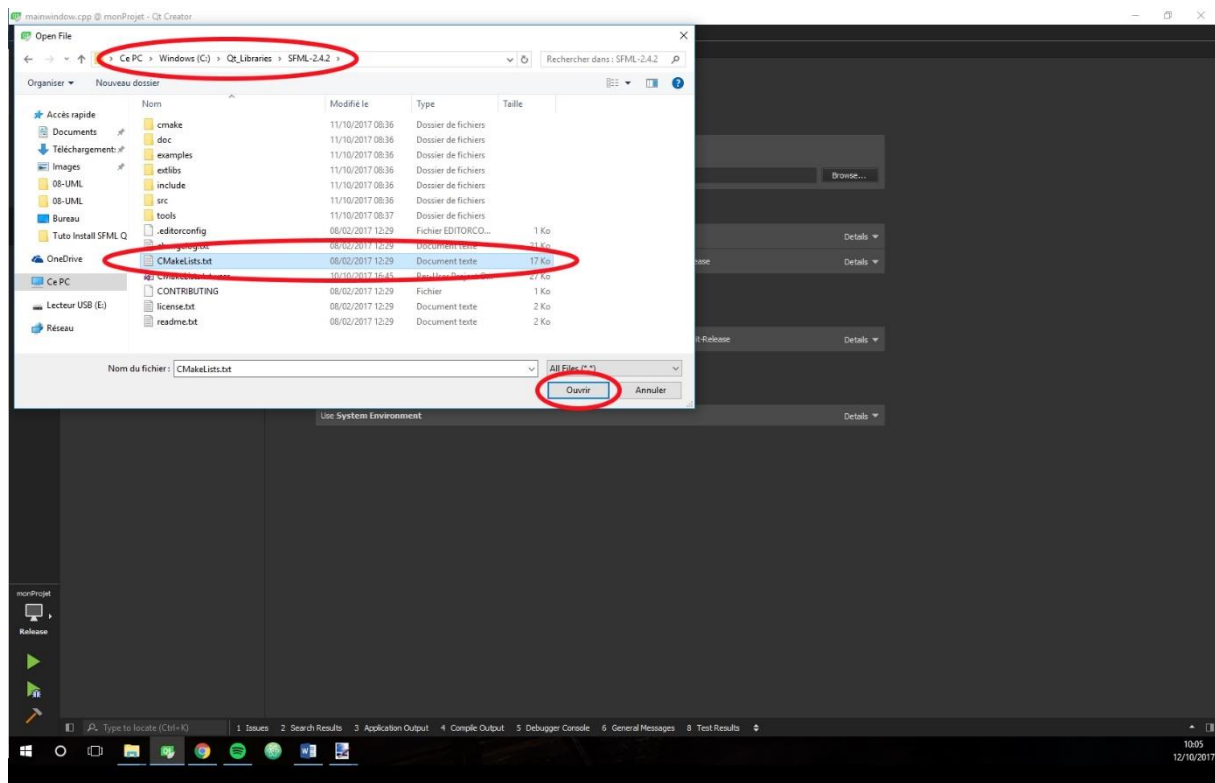
Cela vous a créé 2 nouveaux dossiers.



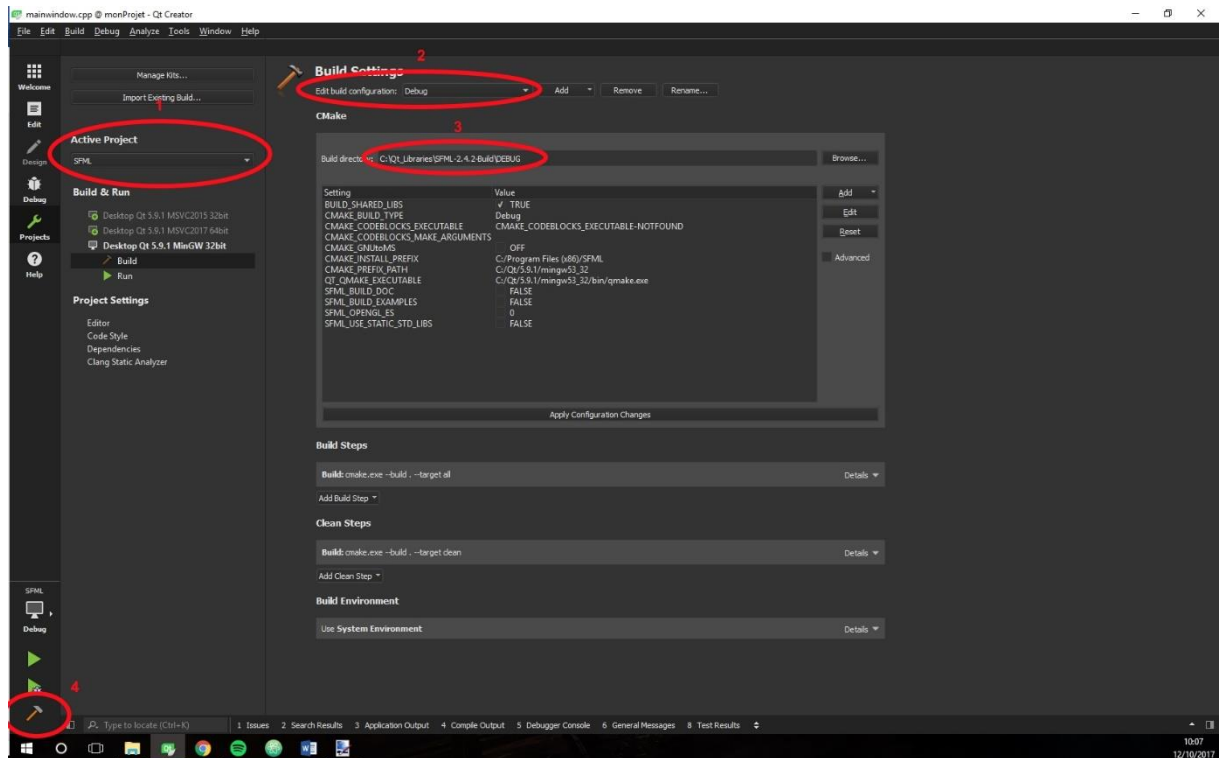
5) Dans le dossier contenant votre fichier dézippé de la SFML, créez un dossier dont le nom ressemble à « SFML-votreVersion-Build » (vous pouvez mettre ce que vous voulez) et créez également dans ce dernier 3 sous-dossiers : « DEBUG », « RELEASE » et « LIBS ».



6) Toujours dans Qt, allez dans « File » puis « Open File or Project... » et cherchez le fichier CMakeLists.txt qui se trouve dans le dossier de la SFML que vous avez dézippé. Surtout ici, si Qt vous demande, faites bien attention à encore choisir « MinGW » comme compilateur. Il faut que le compilateur de « monProjet » et celui de « CMakeLists.txt » soient identiques. Laissez le temps à Qt pour tout créer et lier de lui-même.

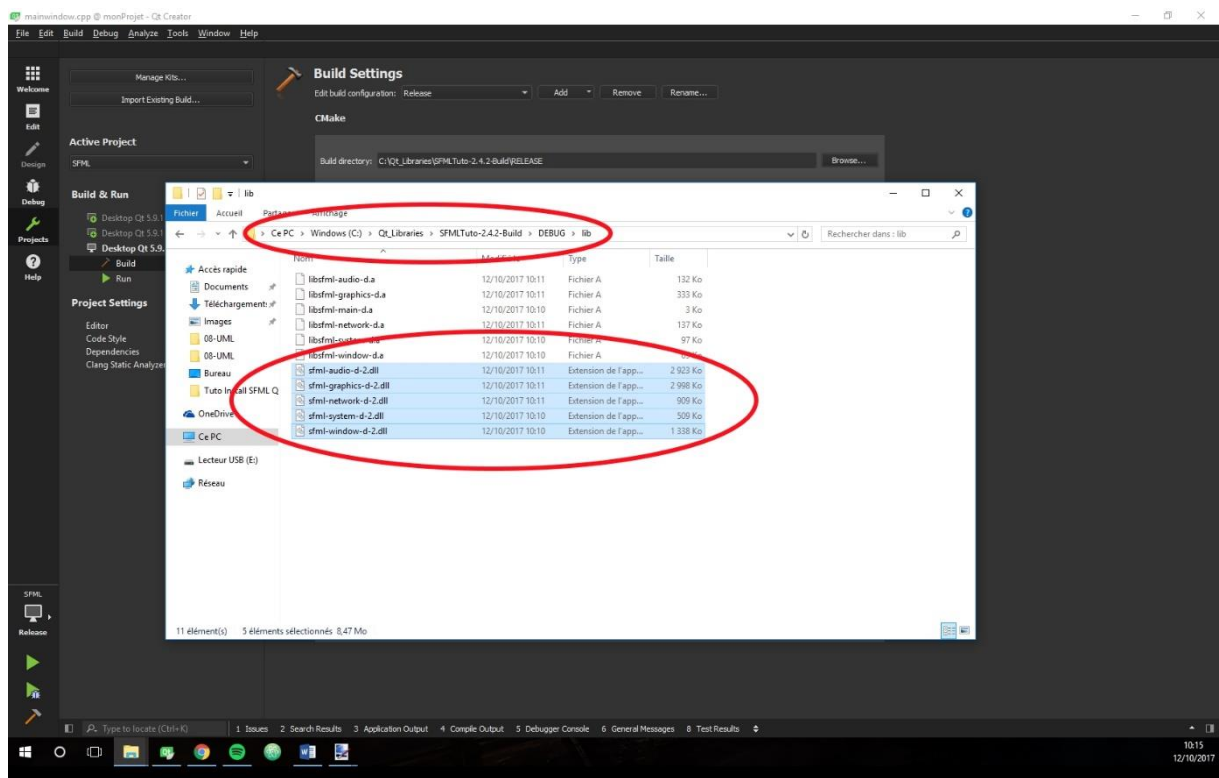


7) Retournez ensuite dans l'onglet « Projects » et regardez bien que dans « Active Project » « SFML » soit sélectionné. Encore une fois, laissez bien « Shadow build » coché et changez le « Build directory » pour qu'il corresponde au dossier « DEBUG » dans « SFML-votreVersion-Build ». Vérifiez que vous êtes bien en « Debug » en haut et lancez un build. Cela vous crée des nouveaux fichiers indispensables à la compilation de la SFML par CMake. Vous pouvez vérifier dans « DEBUG ».

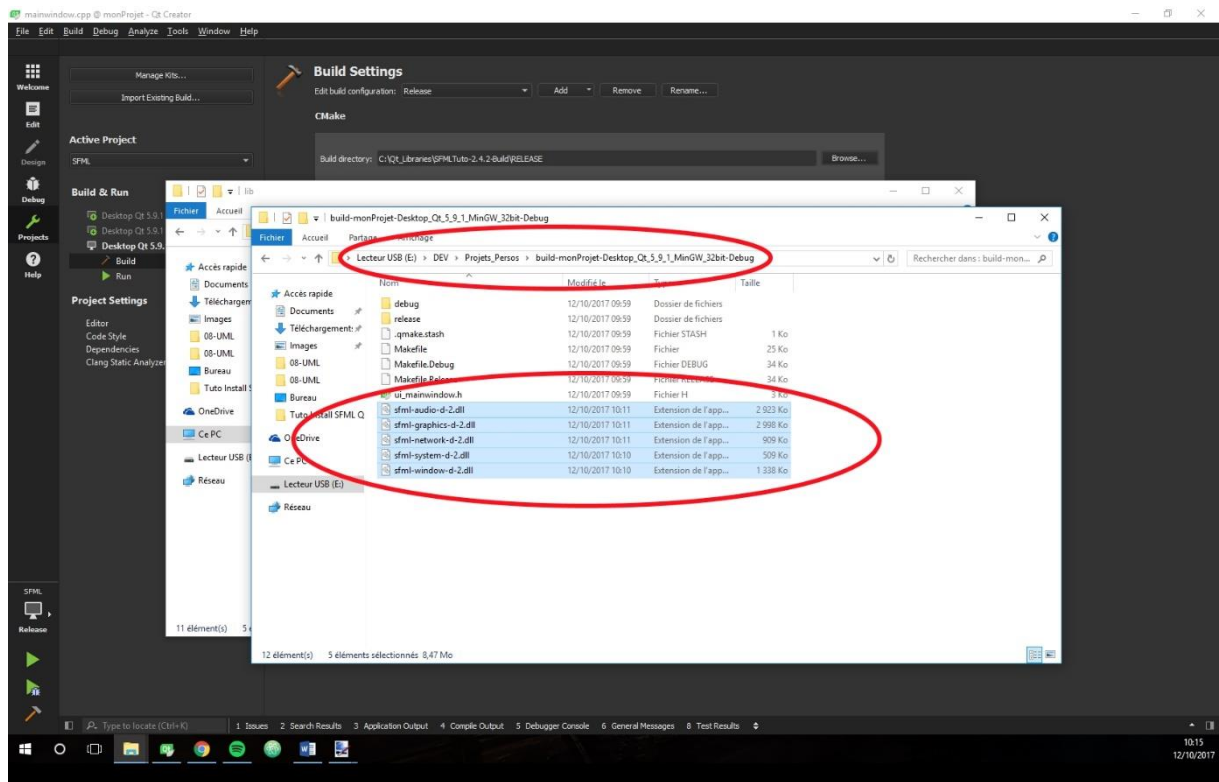


8) Faire la même chose en « Release » dans « SFML-votreVersion-build/RELEASE ». Lancez un build.

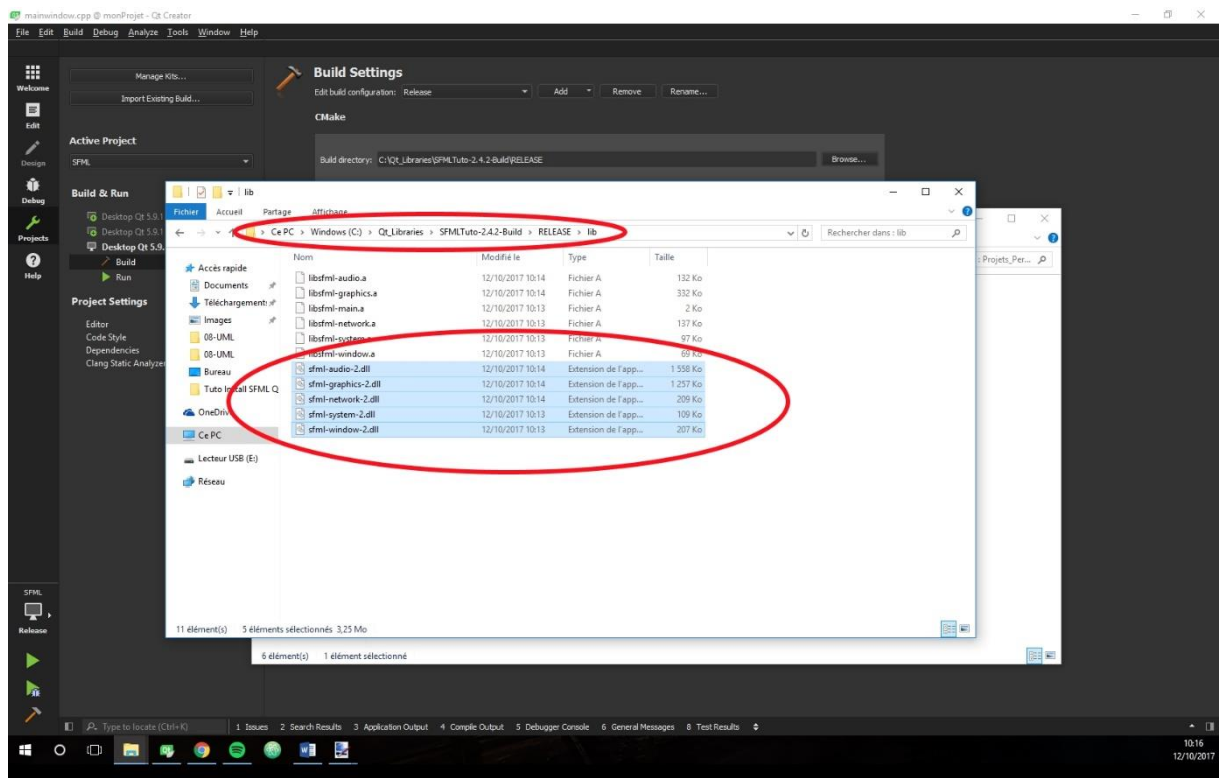
9) Allez dans le dossier « SFML-votreVersion-Build/DEBUG/lib » et copiez TOUS les fichiers .dll. Collez les dans le dossier « build-nomProjet-Desktop\_Qt\_5\_9\_1\_MinGW\_32bit-Debug ».



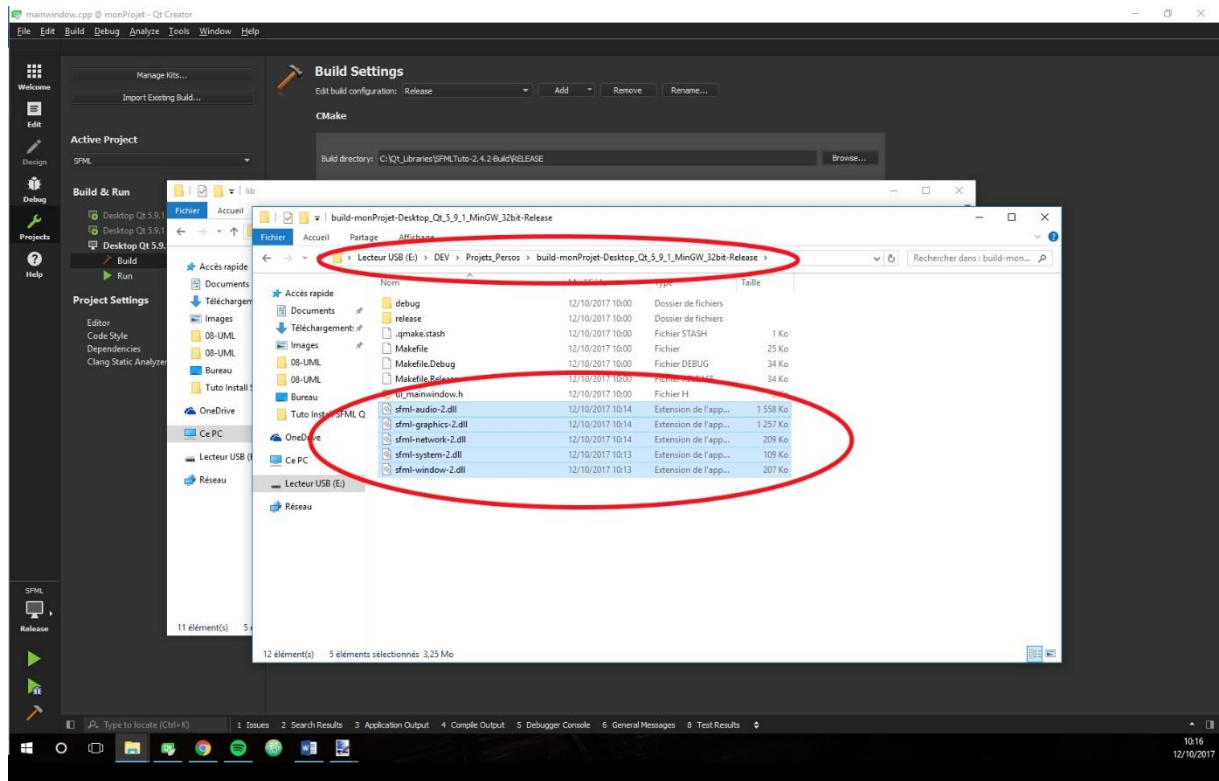




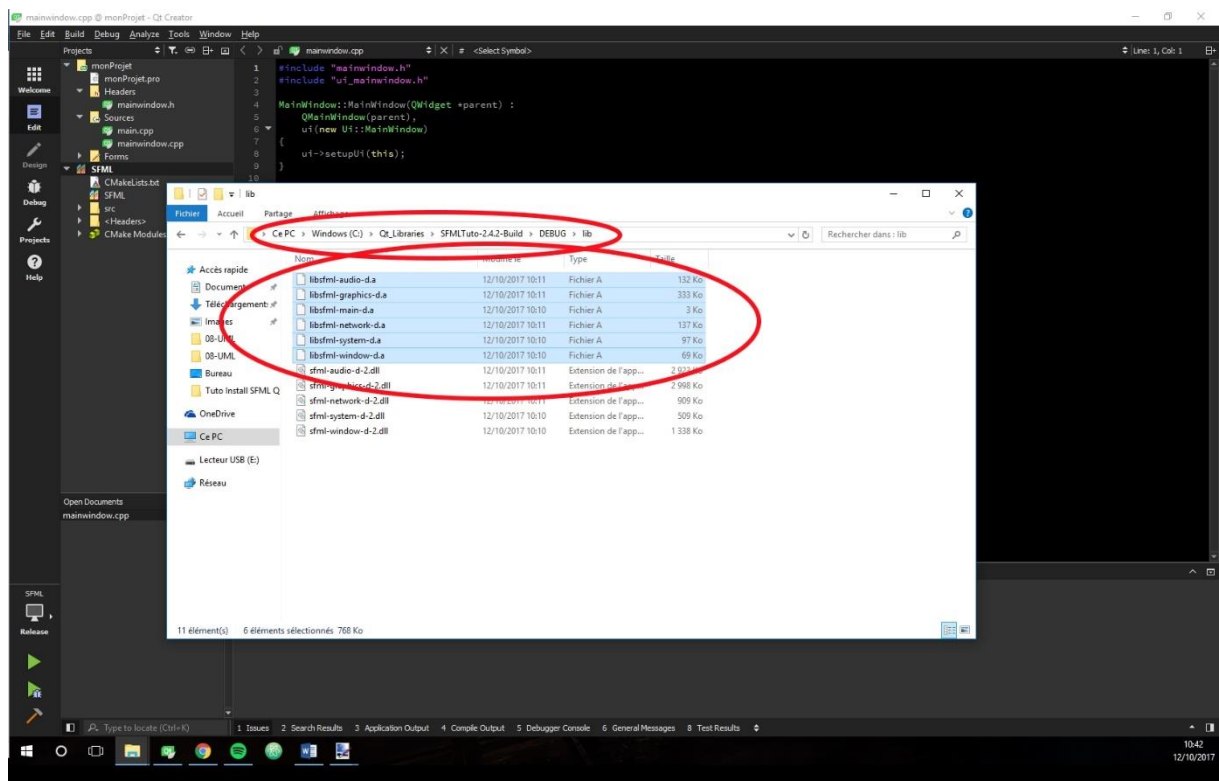
10) Allez dans le dossier « SFML-votreVersion-Build/RELEASE/lib » et copiez TOUS les fichiers .dll. Collez les dans le dossier « build-nomProjet-Desktop\_Qt\_5\_9\_1\_MinGW\_32bit-Release ».

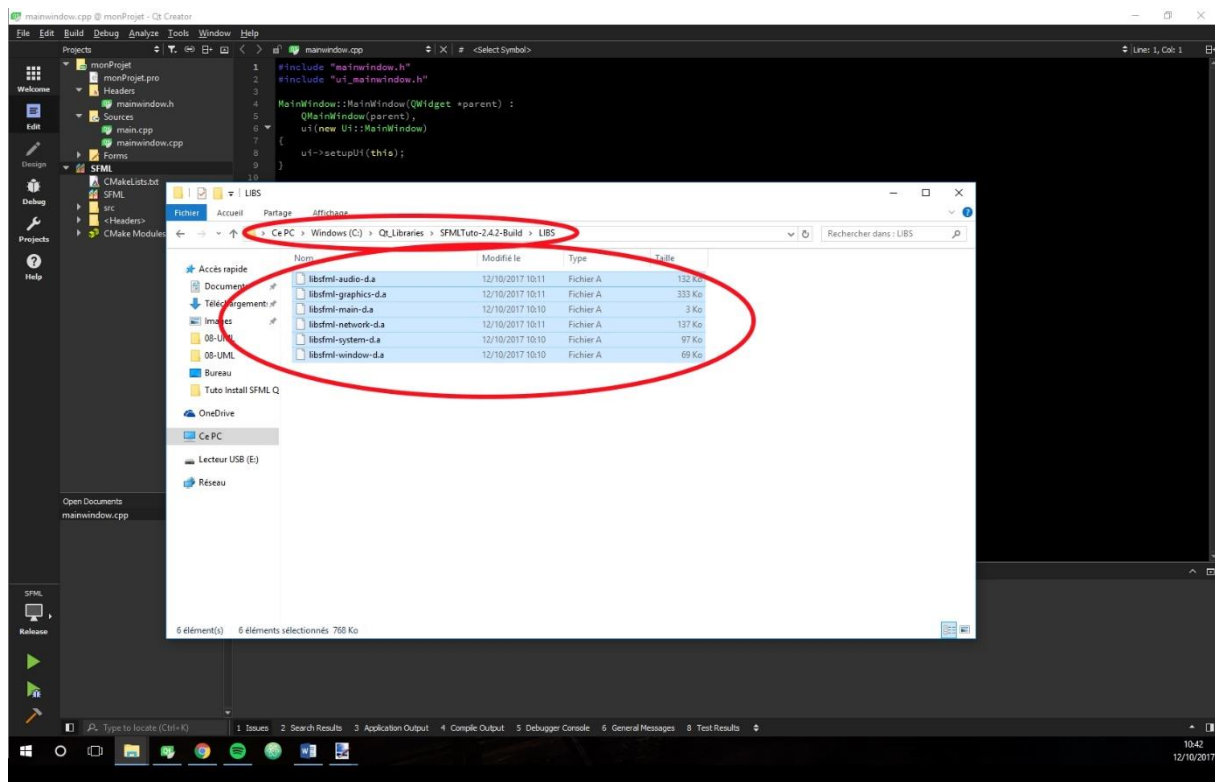




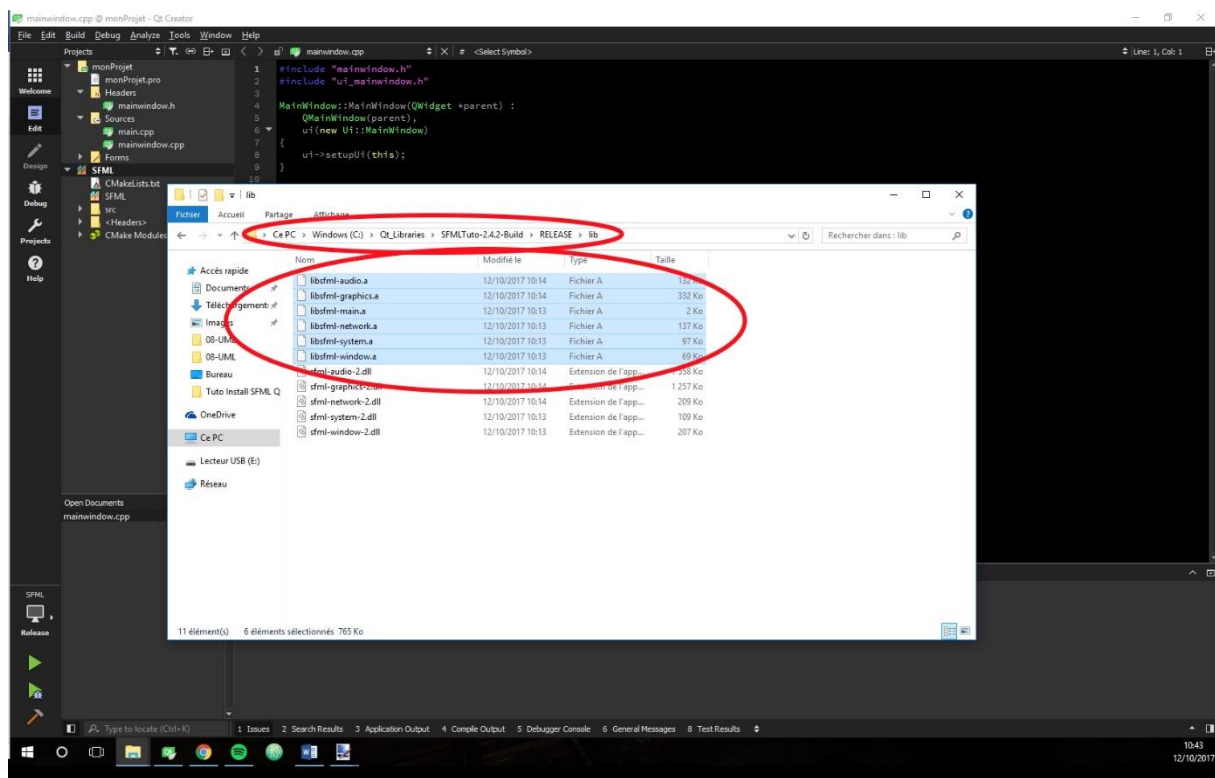


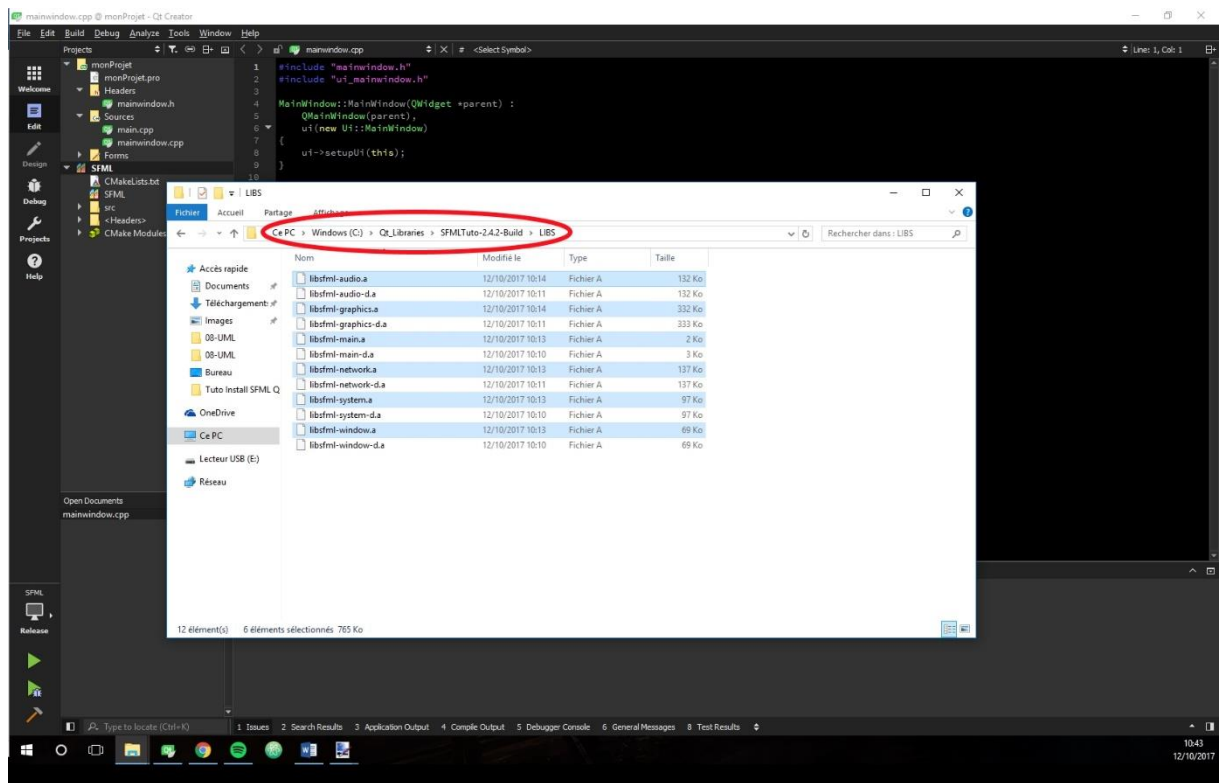
11) Copier TOUS les .a de « SFML-2.4.2-build/DEBUG/lib » dans « SFML-2.4.2-build/LIBS ».





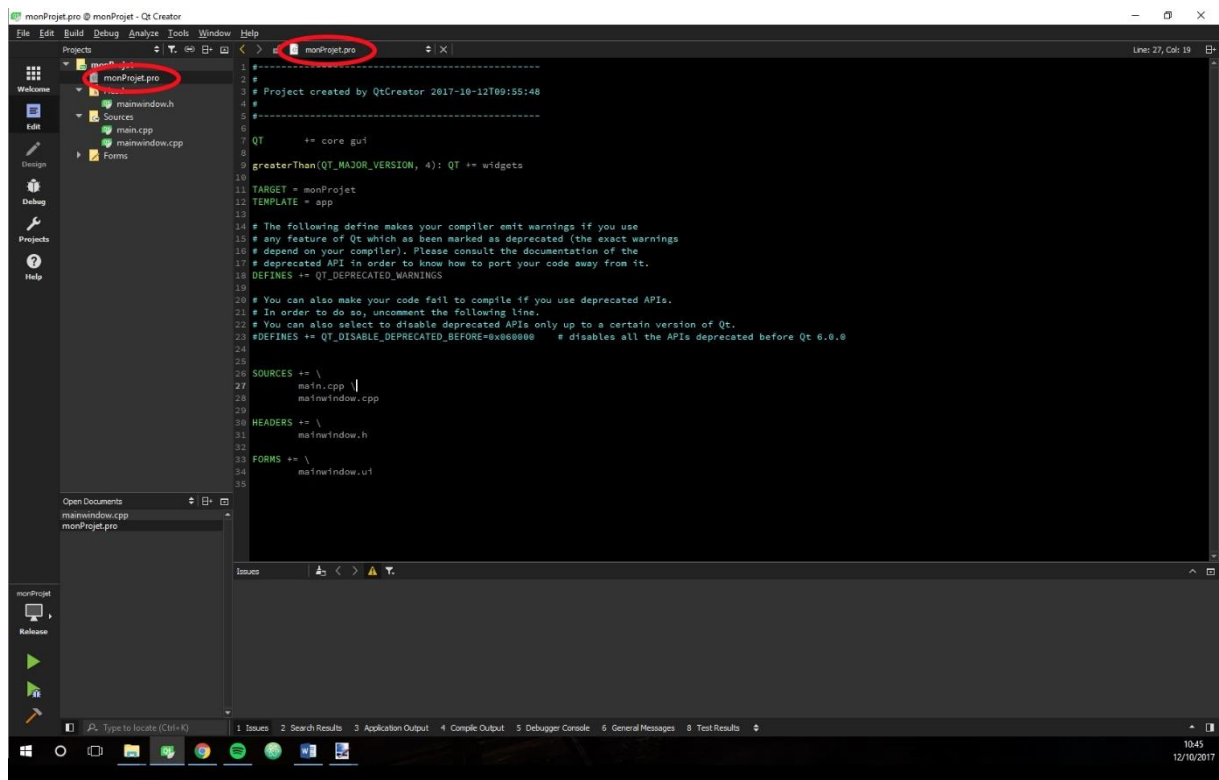
12) Copier TOUS les .a de « SFML-2.4.2-build/RELEASE/lib » dans « SFML-2.4.2-build/LIBS ».





13) Fermer le projet « SFML » dans Qt. Plus besoin.

14) Dans Qt, repérez le .pro de votre projet et ouvrez le pour l'éditer.



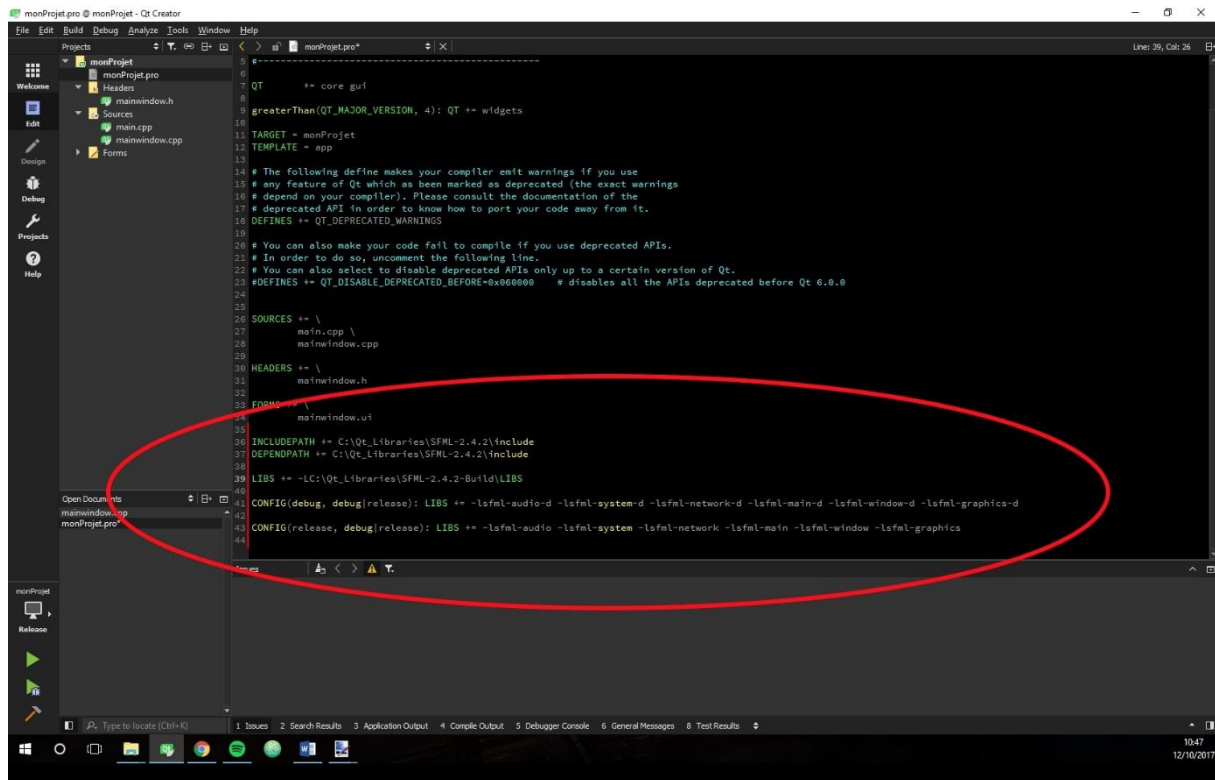
15) Rajouter à la fin du .pro :

```
INCLUDEPATH += C:\cheminDestination\SFML-2.4.2\include
DEPENDPATH += C:\cheminDestination\SFML-2.4.2\include

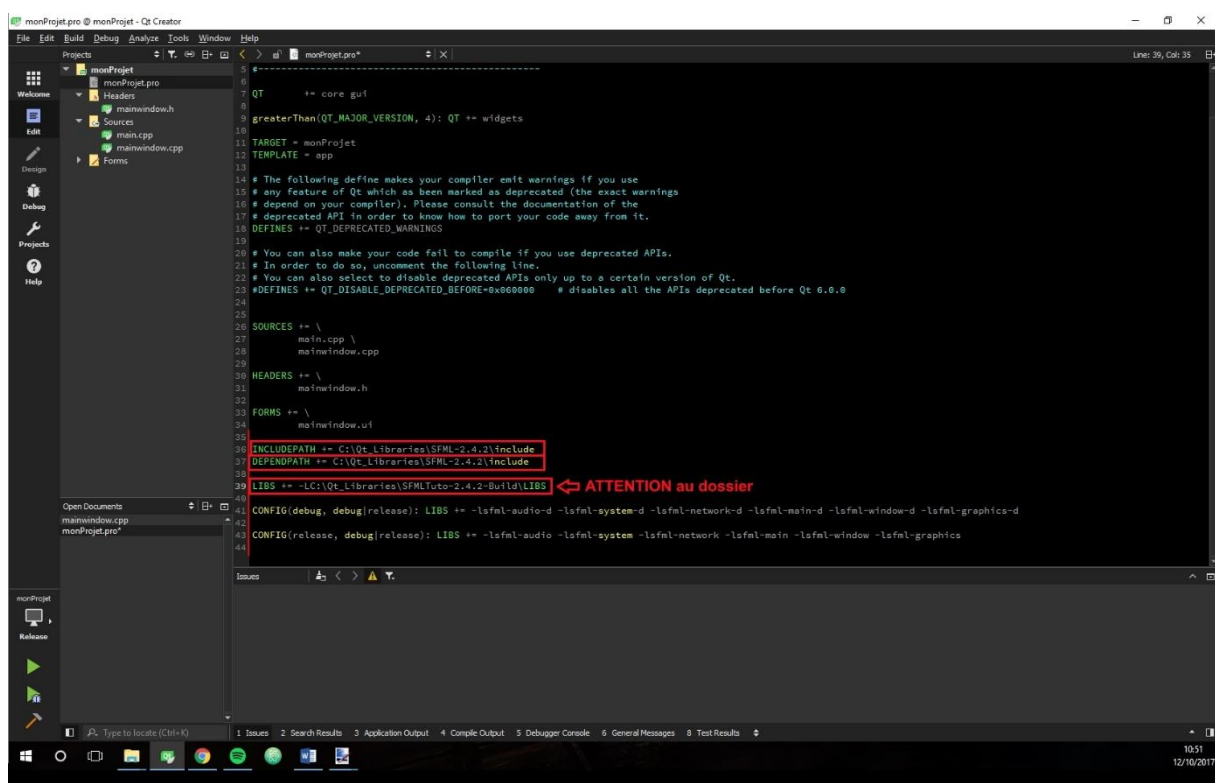
LIBS += -LC:\cheminDestination\SFML-votreVersion-Build\LIBS

CONFIG(debug, debug|release): LIBS += -lsfml-audio-d -lsfml-system-d -lsfml-network-d -lsfml-main-d -lsfml-window-d -lsfml-graphics-d

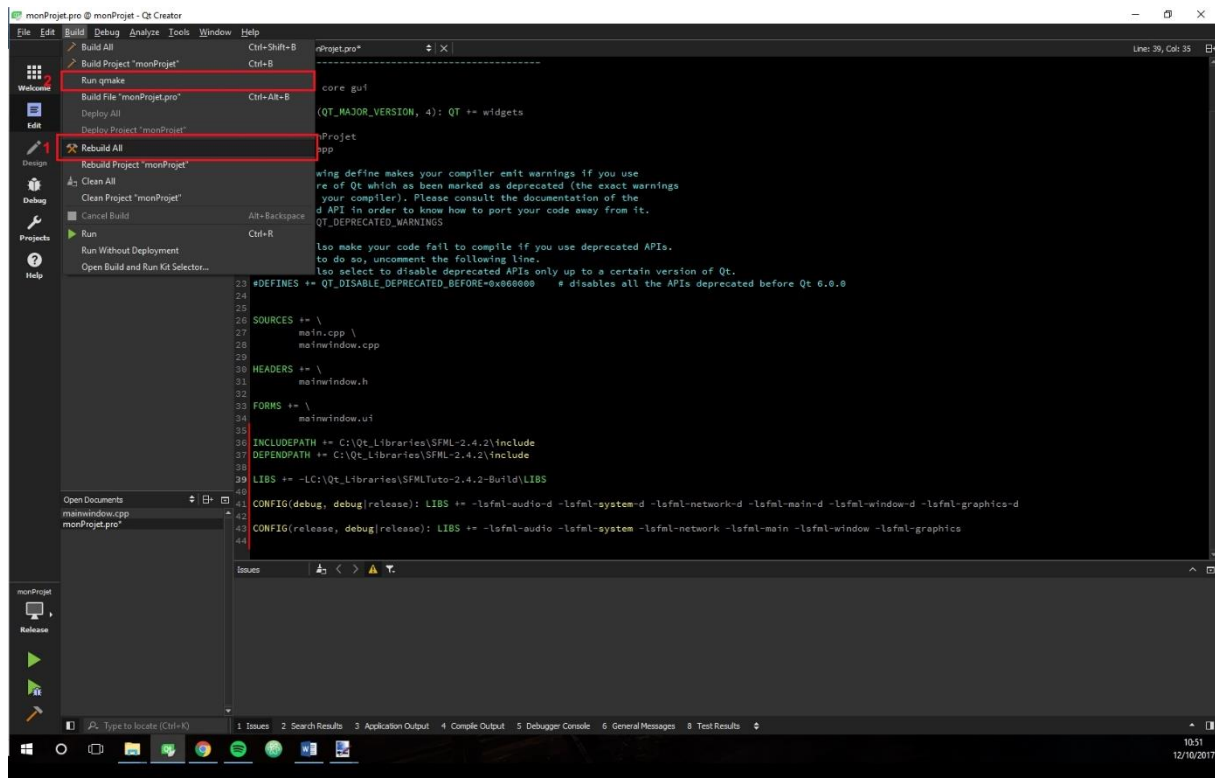
CONFIG(release, debug|release): LIBS += -lsfml-audio -lsfml-system -lsfml-network -lsfml-main -lsfml-window -lsfml-graphics
```



16) Remplacez « cheminDestination » par le chemin correspondant à votre dossier contenant la SFML. ATTENTION : pour la ligne « LIBS += ... » le dossier pointé doit être celui que vous avez créé au début, c'est-à-dire celui contenant « DEBUG », « LIBS » et « RELEASE ».



17) Dans Qt, « build » -> « Rebuild All » puis « Run qmake ».



18) Et voilà ! Pour tester si l'installation s'est bien déroulée entrez le code ci-dessous dans votre « main.cpp ».

```
#include "mainwindow.h"
#include <QApplication>
#include <SFML/Graphics.hpp>

using namespace sf;

int main(int argc, char *argv[])
//{
//    QApplication a(argc, argv);
//    MainWindow w;
//    w.show();

//    return a.exec();
//}
{
RenderWindow window(VideoMode(800, 600), "SFML DANS QT");
RectangleShape rect;
rect.setPosition(Vector2f(400, 300));
rect.setSize(Vector2f(30, 30));
rect.setFillColor(Color(33, 42, 2, 255));

while (window.isOpen())
{
Event e;

while (window.pollEvent(e))
{
if (e.type == Event::Closed)
{
window.close();
}
}
window.clear();
window.draw(rect);
window.display();
}
}
```

19) Voici 2 liens très utiles pour apprendre à maîtriser cette bibliothèque :

<https://www.sfm1-dev.org/learn-fr.php>

<https://www.developpez.net/forums/f1393/applications/developpement-2d-3d-jeux/api-graphiques/sfm1/>