

## Installation d'un environnement de développement pour Windows:

Pour pouvoir suivre ce cours, il vous faudra un environnement nécessaire au développement de projets de programmation en C++ :

- un compilateur C++ (conseillé : GCC ou Clang) ;
- un environnement de développement intégré (EDI, conseillé: Qt Creator). Notez qu'un simple éditeur de texte peut suffire à la place d'un EDI (conseillé: Geany, dans ce cas), cependant cette dernière option peut s'avérer nettement moins confortable pour l'utilisation des outils des «debugging» ;
- un outil de construction de projets (l'outil cmake <https://cmake.org/>, sera employé dans ce cours).
- une librairie graphique (SFML, <https://www.sfml-dev.org/>, sera utilisée dans ce cours).

Cette page a pour but de vous guider dans le processus d'installation d'un tel environnement.

Merci de suivre scrupuleusement les instructions ci-dessous. Vous pouvez aussi visionner la vidéo qui montre cette installation.

**Important: pour éviter tout problème, il est fortement recommandé d'installer tous les composants décrits dans des répertoires ne comportant ni espace ni caractères spéciaux ou accentués dans leurs noms.**

### Détermination de l'architecture

Avant de commencer l'installation de l'environnement, il est crucial de déterminer l'architecture de votre processeur. Pour ce faire, ouvrez une invite de commande (bouton **Windows+R**, puis entrez `cmd`) puis entrez-y la commande **SET Processor**. La variable `PROCESSOR_ARCHITECTURE` devrait contenir **64** dans sa valeur si c'est une architecture 64bits, ou **86** s'il s'agit d'une architecture 32bits. *Souvenez-vous de cette information et restez consistants dans vos choix de versions d'installation* (TOUJOURS 64bit ou TOUJOURS 32bit)

### Créer un dossier d'installation

Les installations qui suivent génèrent un grand nombre de fichiers. En cas de soucis, les retrouver sur votre ordinateur peut s'avérer pénible si les chemins sont définis par défaut. Pour améliorer cela, créez un dossier **Installation**, dans lequel vous pouvez créer un sous dossier par logiciel installé (`Cmake,sfml,MinGW,QT`) qui vont contenir les fichiers d'installation de chaque logiciel. Vous pourrez ainsi plus facilement supprimer de mauvaises installations.

## Installation de MinGW (g++, compilateur)

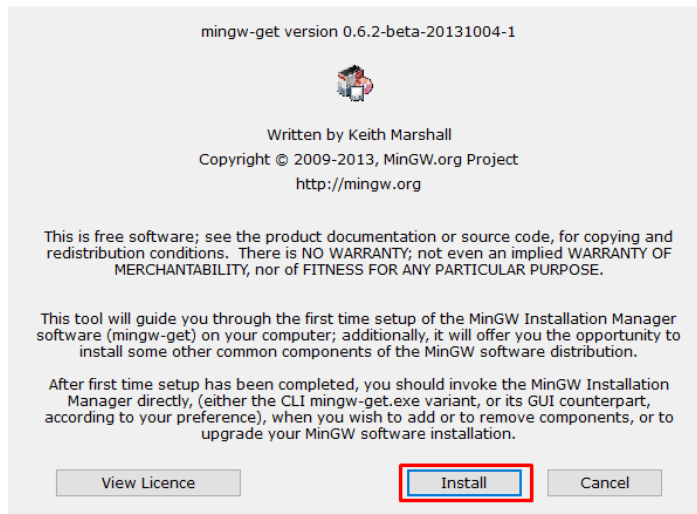
Pour installer le compilateur g++, nous allons installer MinGW :

- Télécharger l'installateur Windows à l'adresse

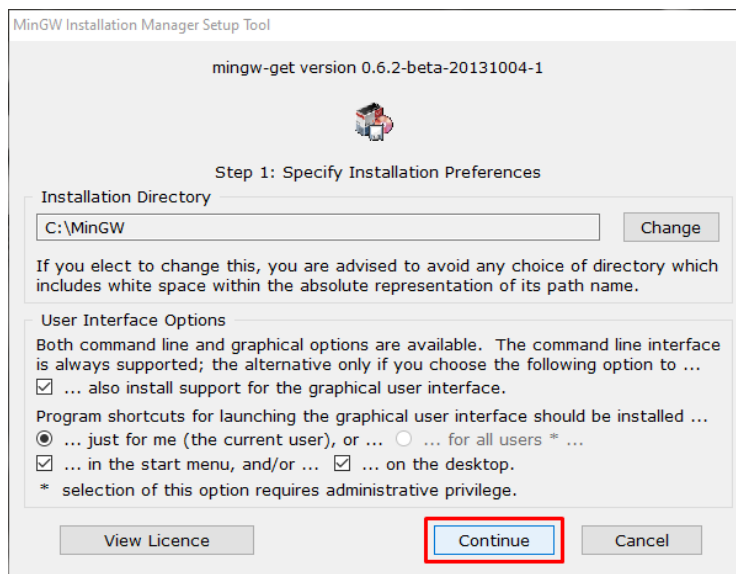
<https://sourceforge.net/projects/mingw/files/latest/download?source=files>

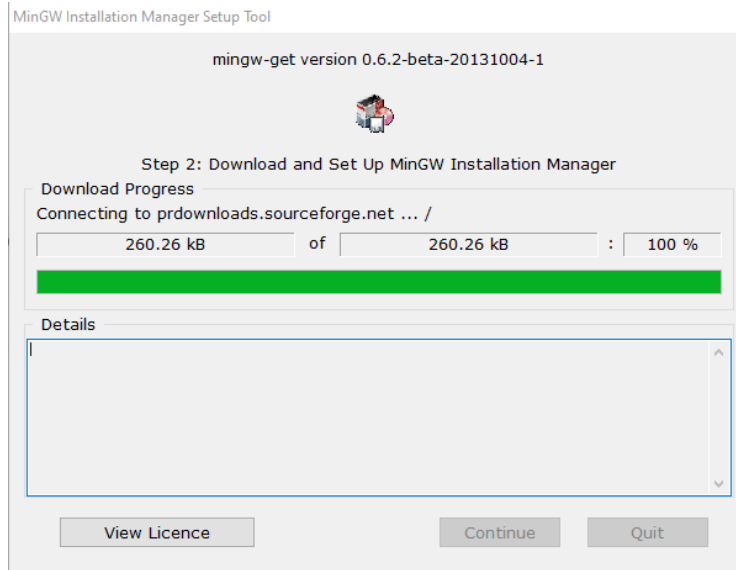
- Lancer l'installation

MinGW Installation Manager Setup Tool

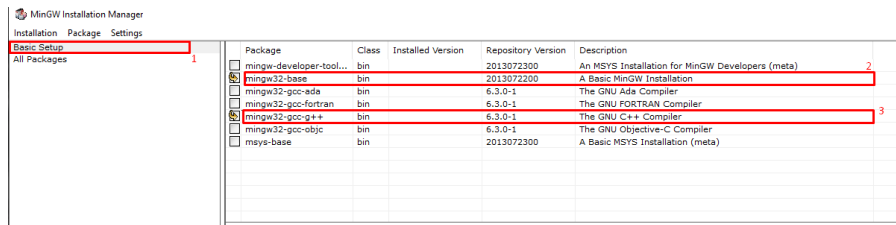


- Remplacer le chemin par défaut C:\MinGW par votre dossier d'installation [...] \Installation\MinGW puis appuyez sur Continue:

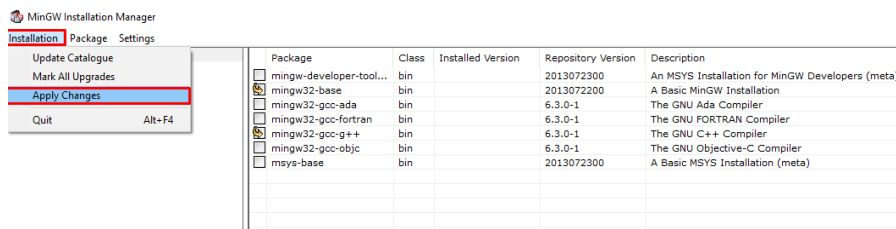




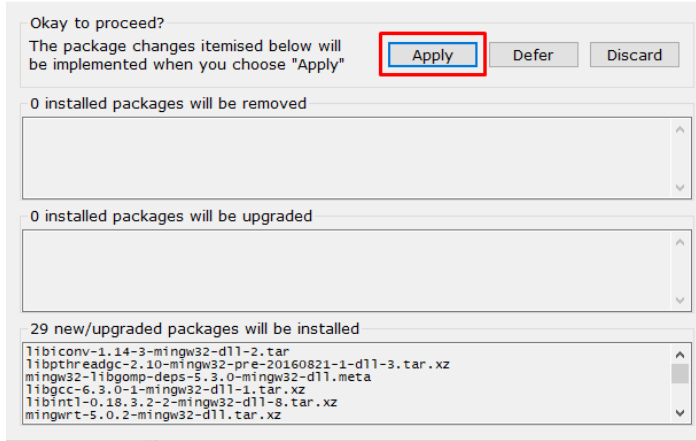
- Dans la fenêtre qui s'ouvre, marquer les packages mingw32-base et mingw32-gcc-g++ (dans la catégorie Basic setup) pour l'installation en cochant la case se situant à gauche puis en cliquant sur **Mark for installation**. Vous pouvez selectionner ces packages quelque soit votre architecture de processeur (64/32 bits), ceci est la seule exception à la règle.



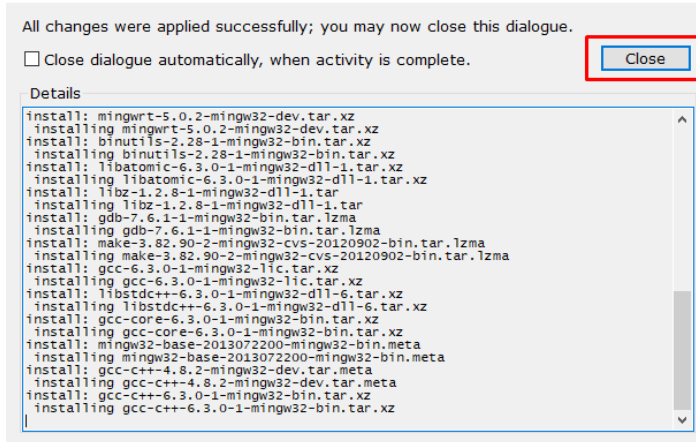
- Rendez-vous dans le menu Installation > Apply changes, puis cliquez sur Apply pour lancer le téléchargement puis l'installation de g++.



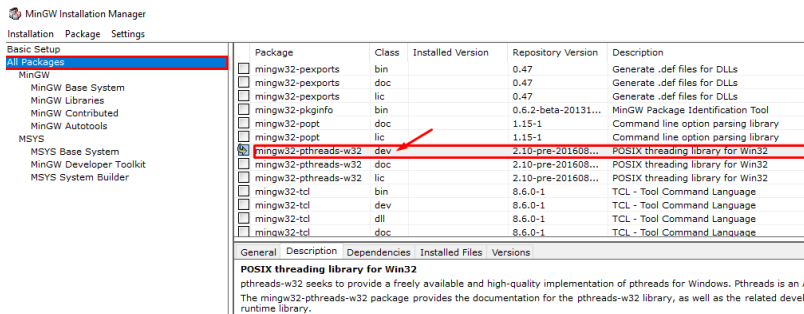
Schedule of Pending Actions



Applying Scheduled Changes

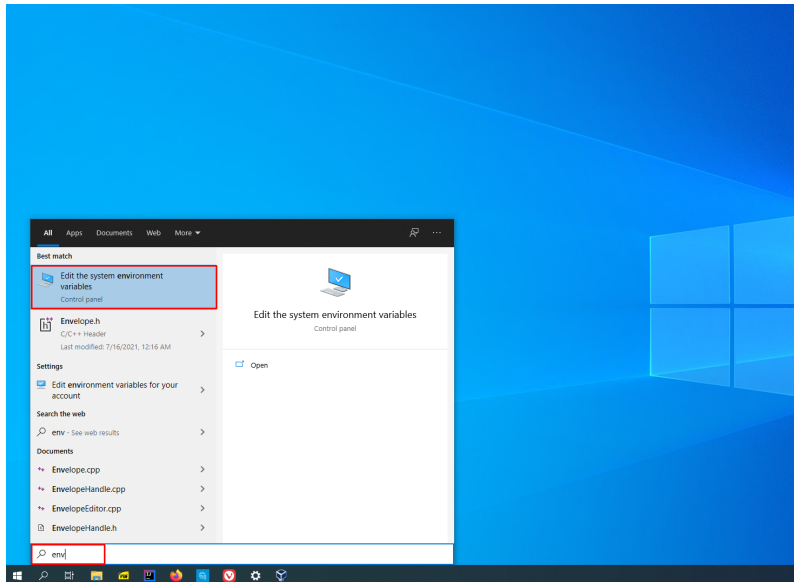


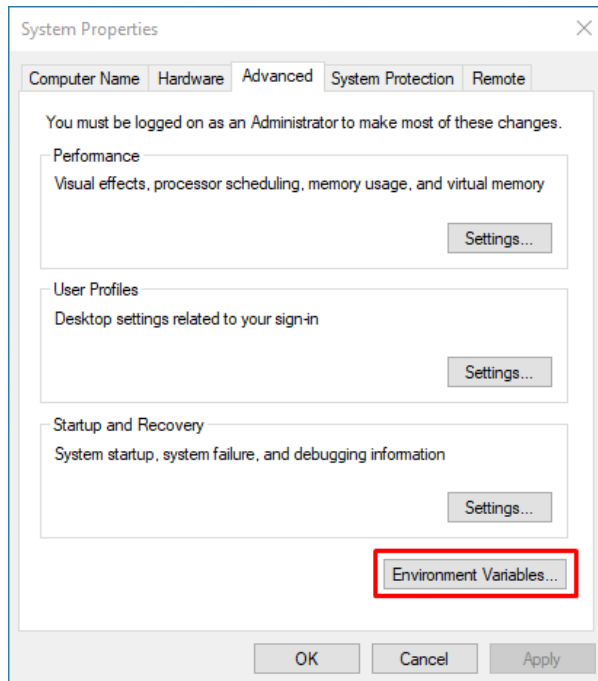
- Sur Windows 10, il faut également installer pthreads pour compiler. Pour ceci il suffit de marquer le paquet mingw32-pthreads-w32, classdev, (dans la catégorie All Packages > MinGW) pour installation, en cochant la case se situant à gauche puis en cliquant sur Mark for installation.



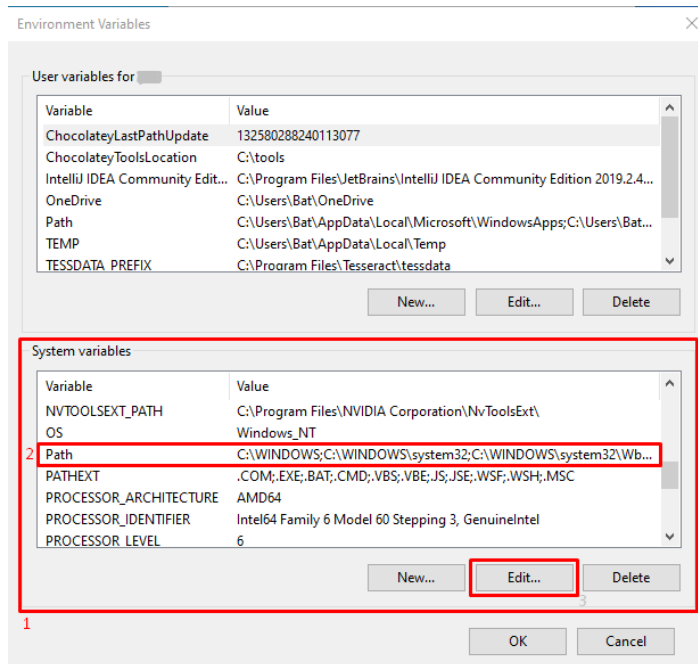
- Rendez-vous ensuite dans le menu Installation > Apply changes, puis cliquez sur Apply.
- Ajouter le répertoire `\bin` du dossier MinGW à la variable d'environnement PATH (voir la vidéo pour les détails).

Sur Windows 10, cela se fait en recherchant les variables d'environnement :

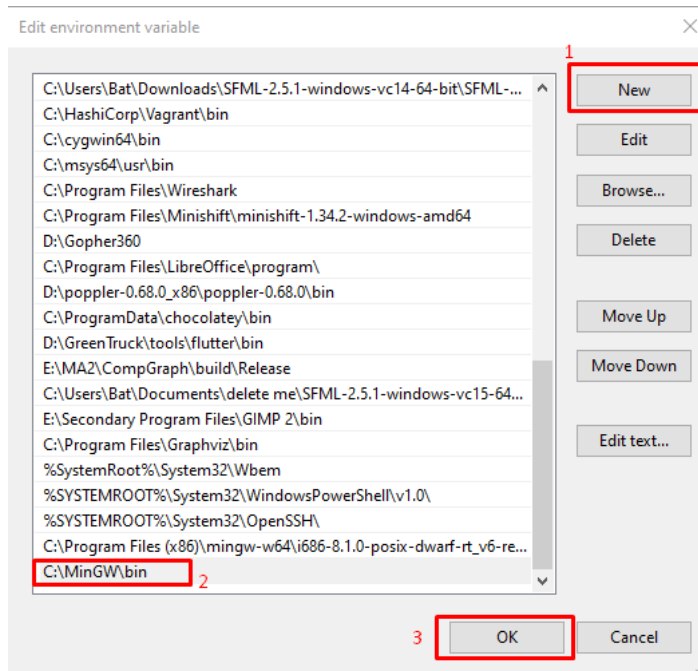




Editer la variable « Path » :



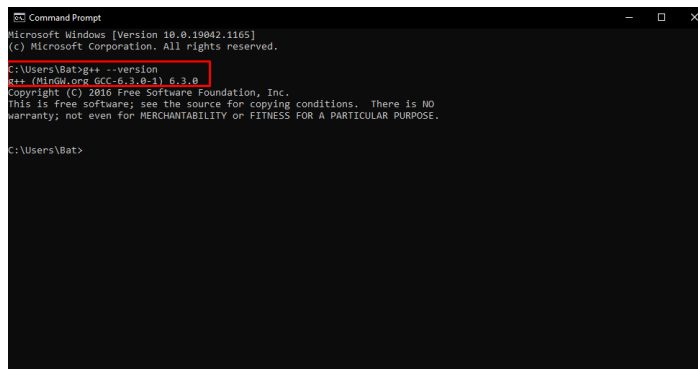
puis ajoutez une nouvelle ligne en bas de la fenêtre :



Vérifiez la version du compilateur en exécutant la commande suivante dans un invite de commandes :

```
g++ --version
```

Votre version doit être équivalente ou supérieure à 5.4.0.

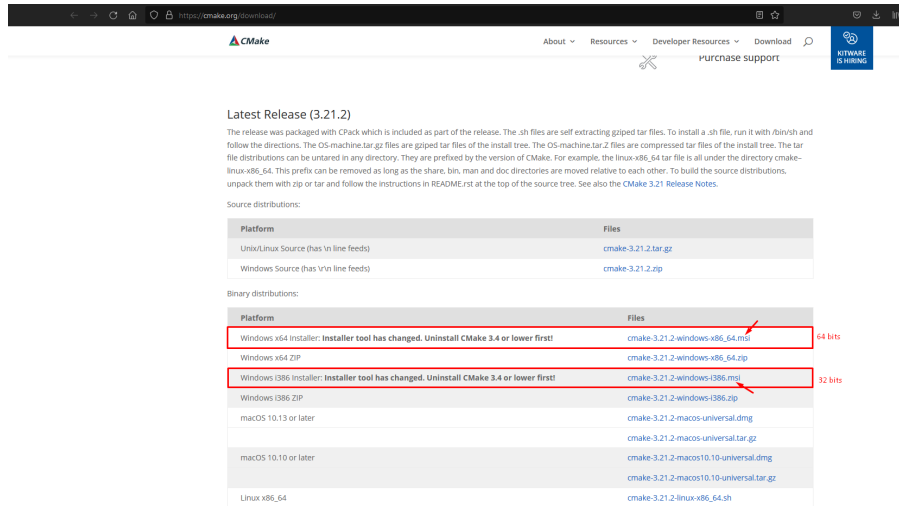


## Installation de CMake

Pour coder le projet de ce cours nous vous fournissons une coquille de base contenant un ensemble de fichiers à utiliser. Cette coquille contiendra en particulier des fichiers de configuration permettant de compiler le projet au moyen

d'un utilitaire appelé CMake qu'il vous faut installer.

- Télécharger l'installateur Windows à l'adresse <https://cmake.org/download>.



Latest Release (3.21.2)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gnuipcd tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gnuipcd tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be unarchived in any directory. They are prefixed by the version of CMake. For example, the linux-x86\_64 tar file is all under the directory cmake-linux-x86\_64. This prefix can be removed as long as the share, bin, main and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in README.txt at the top of the source tree. See also the CMake 3.21 Release Notes.

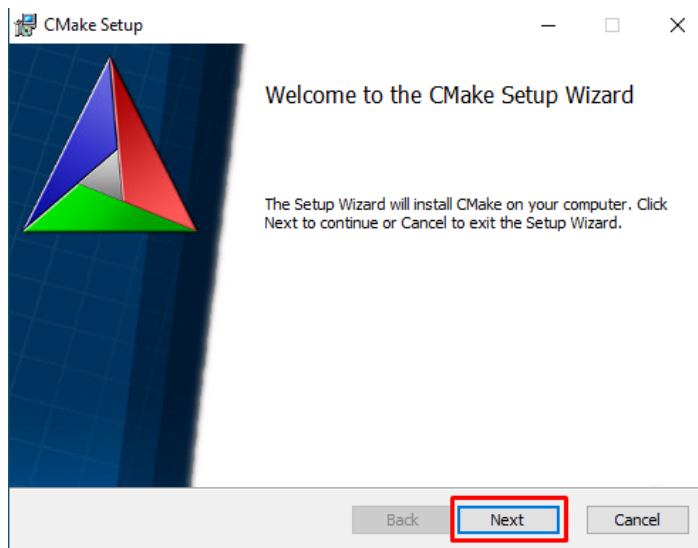
Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.21.2.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.21.2.zip

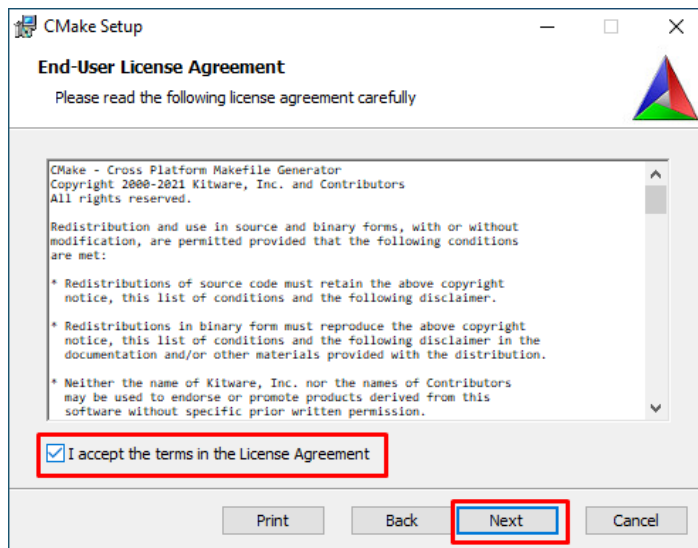
Binary distributions:

Platform	Files
Windows x64 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	cmake-3.21.2.windows-x86_64.msi <span style="color: red;">64 bits</span>
Windows x64 ZIP	cmake-3.21.2.windows-x86_64.zip
Windows (386) Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	cmake-3.21.2.windows-386.msi <span style="color: red;">32 bits</span>
Windows (386) ZIP	cmake-3.21.2.windows-386.zip
macOS 10.13 or later	cmake-3.21.2.macos-universal.dmg cmake-3.21.2.macos-universal.tar.gz
macOS 10.10 or later	cmake-3.21.2.macos10.10-universal.dmg cmake-3.21.2.macos10.10-universal.tar.gz
Linux x86_64	cmake-3.21.2.linux-x86_64.sh

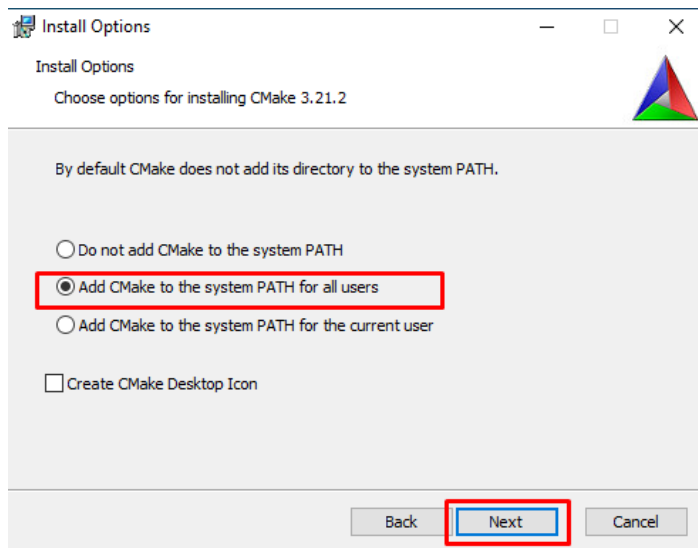
- Lancer l'installation



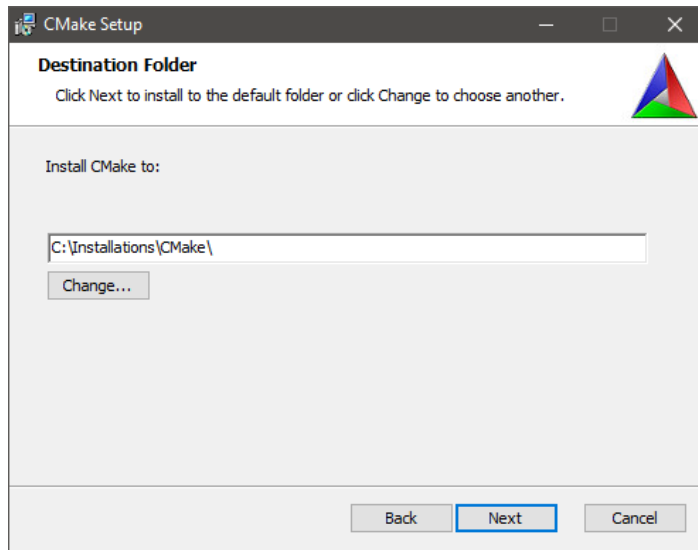




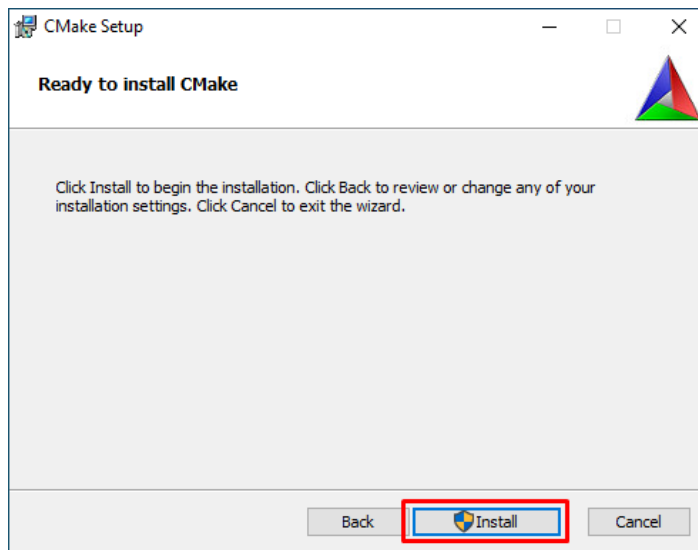
- Ajouter CMake à la variable PATH de Windows

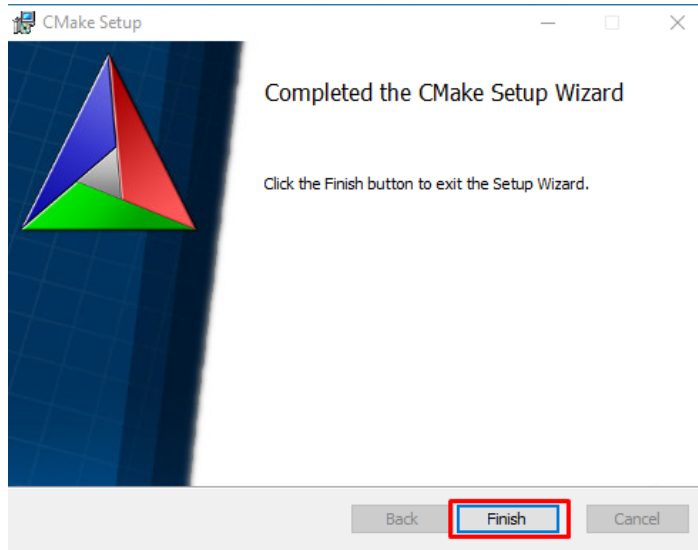


- Modifier le chemin par défaut C:\ProgramFile\CMake par votre dossier d'installation [...] \Installations\CMake, par exemple:



- Installer CMake



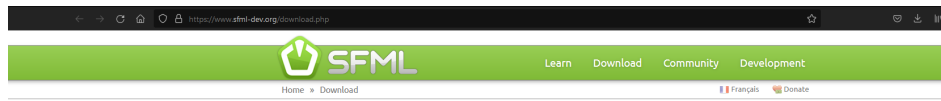


*Attention: Notez le chemin d'installation de Cmake. Lors de la création de projets sur Qt, vous pourriez avoir besoin de spécifier le chemin du répertoire `\bin` de Cmake si les modules Qt ne sont pas sélectionnables (si cela est le cas, sous "Kits" lors de la création de projet, passez votre souris sur le kit grisé, cliquez sur "Manage", puis sous CMake Tool, spécifiez le chemin susmentionné).*

## Installation de SFML(librairie graphique)

Dans la coquille de base qui vous sera initialement fournie, vous trouverez également une interface graphique vous permettant de visualiser vos progrès au fur et à mesure que vous progresserez. Cette interface graphique utilise la librairie SFML, qu'il vous faut également installer.

- Télécharger l'installateur Windows à l'adresse <https://www.sfml-dev.org/download.php>. *Attention: il faut choisir la version GCC MinGW (et non Visual ou autre par exemple). Il faut aussi choisir la version conforme à votre architecture: si elle est en 64 bits (x64), il faut sélectionner la version 64 bits de SFML, si architecture est en 32bits (x86), il faut sélectionner la version 32bits de SFML*



## Download

[SFML 2.5.1](#)  
Latest stable version

[Snapshots](#)  
In development versions

[Bindings](#)  
SFML in other languages

[Git repository](#)  
GitHub.com

[Goodies](#)  
Logos

[Older versions](#)  
SFML 1.6 and 2.x



## Download SFML 2.5.1

On Windows, choosing 32 or 64-bit libraries should be based on which platform you want to compile for, not which OS you have. Indeed, you can perfectly compile and run a 32-bit program on a 64-bit Windows. So you'll most likely want to target 32-bit platforms, to have the largest possible audience. Choose 64-bit packages only if you have good reasons.

**The compiler versions have to match 100%!  
Here are links to the specific MinGW compiler versions used to build the provided packages:  
TDM 5.1.0 (32-bit), MinGW Builds 7.3.0 (32-bit), MinGW Builds 7.3.0 (64-bit)**

Visual C++ 15 (2017) - 32-bit	<a href="#">Download</a>   16.3 MB	Visual C++ 15 (2017) - 64-bit	<a href="#">Download</a>   18.0 MB
Visual C++ 14 (2015) - 32-bit	<a href="#">Download</a>   18.0 MB	Visual C++ 14 (2015) - 64-bit	<a href="#">Download</a>   19.9 MB
Visual C++ 12 (2013) - 32-bit	<a href="#">Download</a>   18.3 MB	Visual C++ 12 (2013) - 64-bit	<a href="#">Download</a>   20.3 MB
GCC 5.1.0 TDM (SJLJ) - Code:Blocks - 32-bit	<a href="#">Download</a>   14.1 MB		
<b>GCC 7.3.0 MinGW (DW2) - 32-bit</b>	<a href="#">Download</a>   15.5 MB	<b>GCC 7.3.0 MinGW (SEH) - 64-bit</b>	<a href="#">Download</a>   16.5 MB



## Download SFML 2.5.1

On Windows, choosing 32 or 64-bit libraries should be based on which platform you want to compile for, not which OS you have. Indeed, you can perfectly compile and run a 32-bit program on a 64-bit Windows. So you'll most likely want to target 32-bit platforms, to have the largest possible audience. Choose 64-bit packages only if you have good reasons.

**The compiler versions have to match 100%!  
Here are links to the specific MinGW compiler versions used to build the provided packages:  
TDM 5.1.0 (32-bit), MinGW Builds 7.3.0 (32-bit), MinGW Builds 7.3.0 (64-bit)**

Visual C++ 15 (2017) - 32-bit	<a href="#">Download</a>   16.3 MB	Visual C++ 15 (2017) - 64-bit	<a href="#">Download</a>   18.0 MB
Visual C++ 14 (2015) - 32-bit	<a href="#">Download</a>   18.0 MB	Visual C++ 14 (2015) - 64-bit	<a href="#">Download</a>   19.9 MB
Visual C++ 12 (2013) - 32-bit	<a href="#">Download</a>   18.3 MB	Visual C++ 12 (2013) - 64-bit	<a href="#">Download</a>   20.3 MB
GCC 5.1.0 TDM (SJLJ) - Code:Blocks - 32-bit	<a href="#">Download</a>   14.1 MB		
GCC 7.3.0 MinGW (DW2) - 32-bit	<a href="#">Download</a>   15.5 MB	GCC 7.3.0 MinGW (SEH) - 64-bit	<a href="#">Download</a>   16.5 MB

Vous avez choisi d'ouvrir :

SFML-2.5.1-windows-gcc-7.3.0-mingw-64-bit.zip  
qui est un fichier de type : Compressed (zipped) Folder (16.5 Mo)  
à partir de : https://www.sfml-dev.org

Que doit faire Firefox avec ce fichier ?

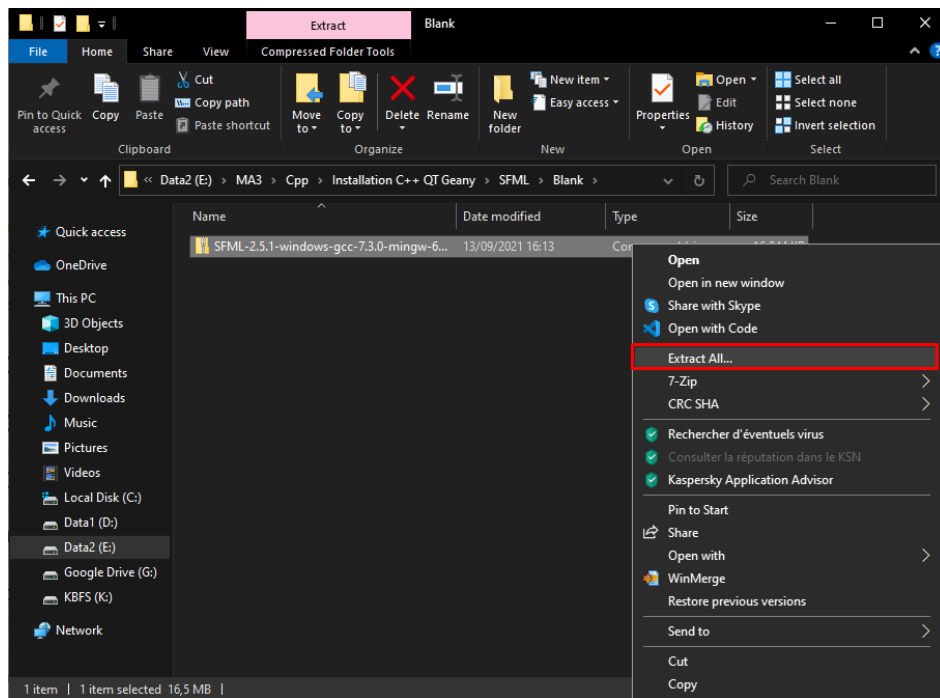
Ouvrir avec Windows Explorer (par défaut)

Enregistrer le fichier

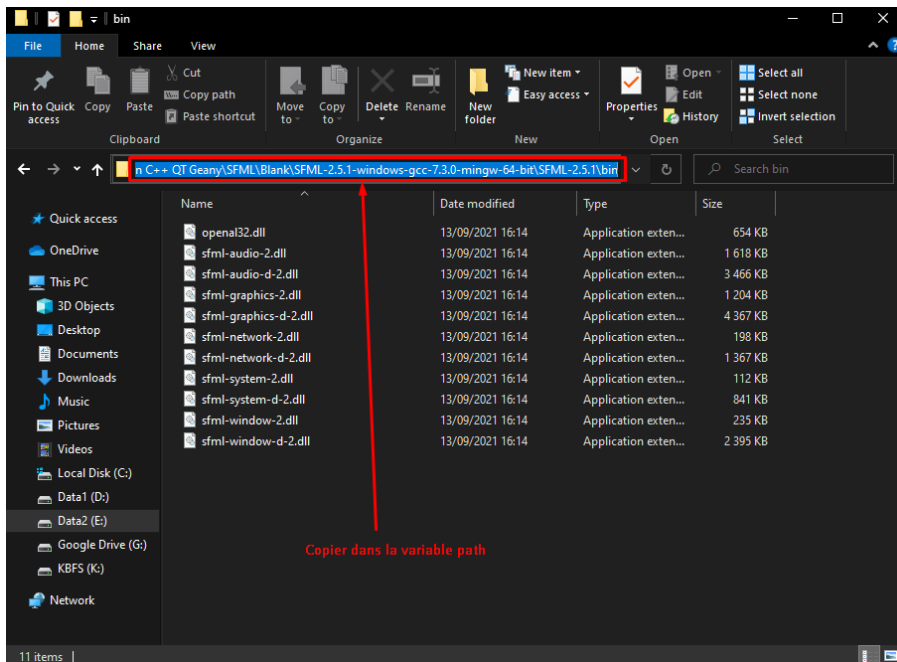
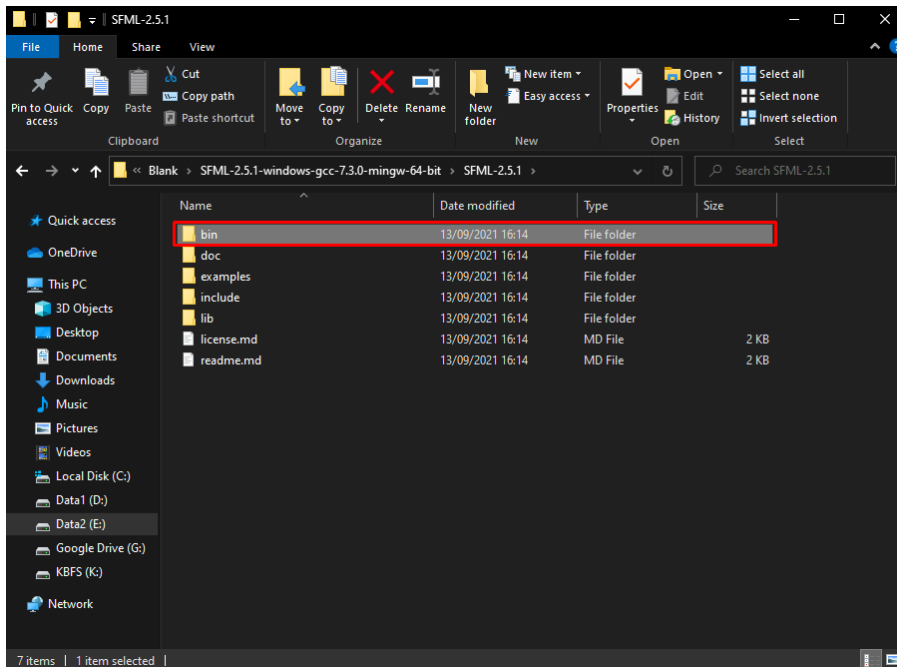
Toujours effectuer cette action pour ce type de fichier.

OK Annuler

- Dézipper le dossier vers votre dossier d'installation [...] \Installations\sFML

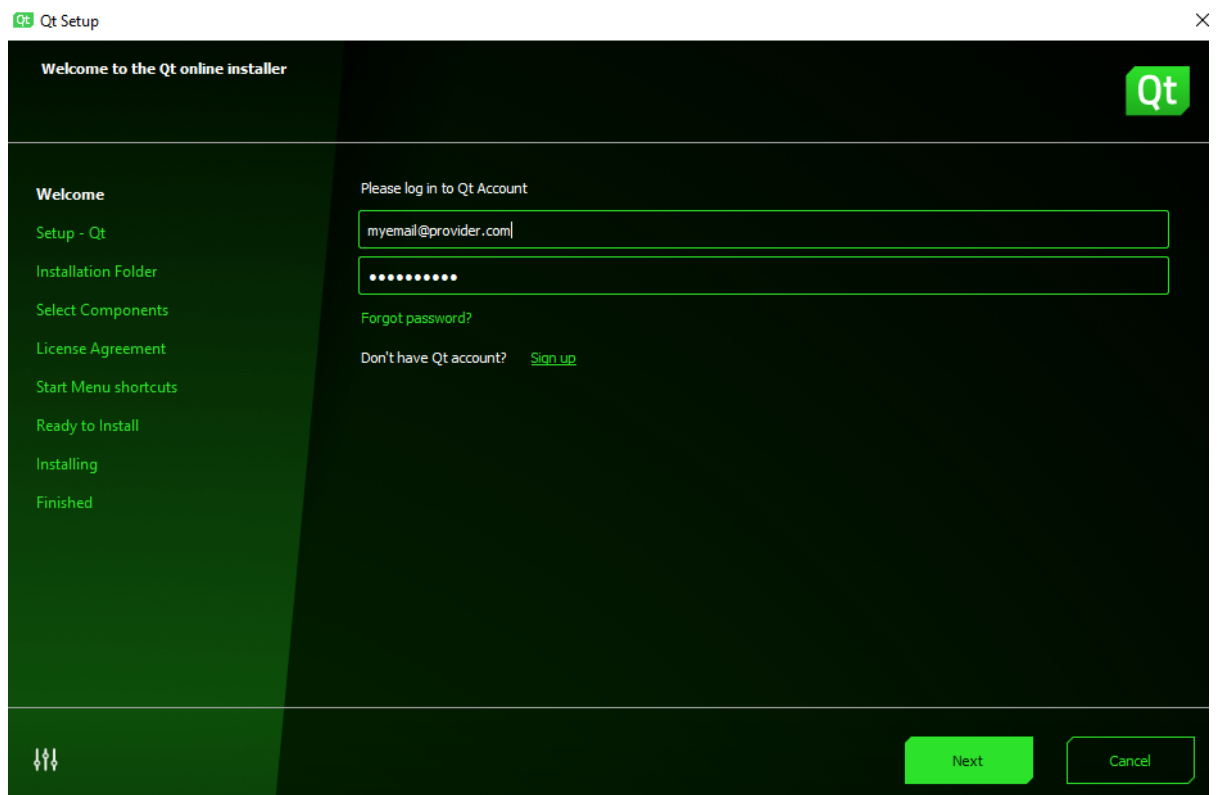


- Ajouter le répertoire \bin du dossier SFML à la variable d'environnement PATH (revoir la manipulation pour l'ajout du compilateur au PATH plus haut si nécessaire).




## Installation de Qt Creator

- Télécharger l'installateur Windows à l'adresse <https://www.qt.io/download> La version open source suffit amplement pour ce projet: une fois sur le site, cliquez sur "Go open source" sous "Downloads for open source users", puis au bas de la page qui s'affiche, cliquer sur "Download the Qt Online Installer".
- Lancer l'installation Vous aurez besoin de vous authentifier ou de créer un compte pour accéder à l'installateur, utilisez une adresse mail valide puis authentifiez vous via l'installateur.



### Qt Open Source Usage Obligations



Welcome

**Open Source Obligations**

Setup - Qt

Installation Folder

Select Components

License Agreement

Start Menu shortcuts

Ready to Install

Installing

Finished

Qt Open Source version is available under GNU General Public License v3 and Lesser GNU General Public License v3. A few components are available under GNU General Public License v2. Please read and accept the Open Source Usage Obligations below. Reading the link below helps you choosing the right license for your project.

[Choosing the right license for your projects](#)  
[Buy Qt](#)

**(Lesser) GNU General Public License v3 obligations**

- You must not combine code developed with a commercial Qt license with code developed with an open source license of Qt in one project or product
- Provide a license copy & explicitly acknowledge Qt use
- Make a Qt source code copy available for customers
- Accept that Qt source code modifications are non-proprietary
- Make consumer devices, which allow users to access to install and run modified versions of the SW inside them
- Accept Digital Rights Management terms, please see the [GPL FAQ](#)
- Take special consideration when attempting to enforce software patents [FAQ](#)
- For further information, please refer to [GPLv3](#) and [LGPLv3](#)

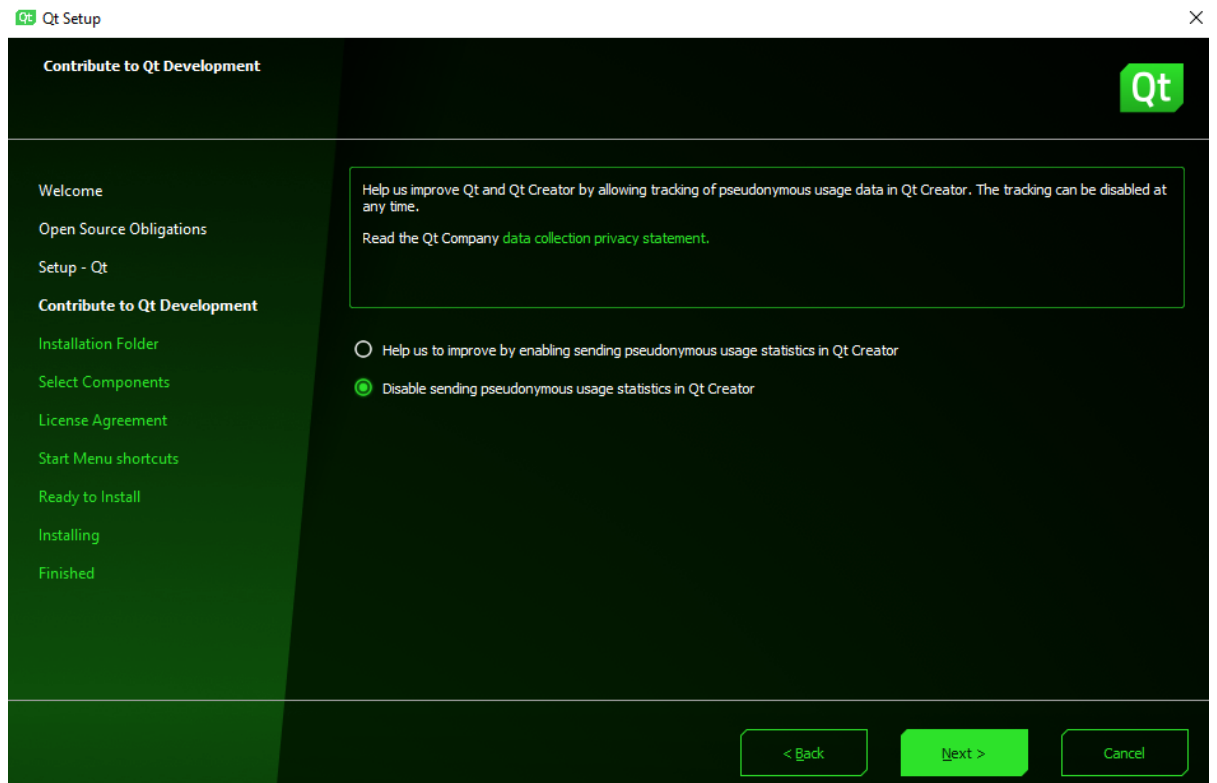
I have read and approve the obligations of using Open Source Qt

Please enter your company/business name

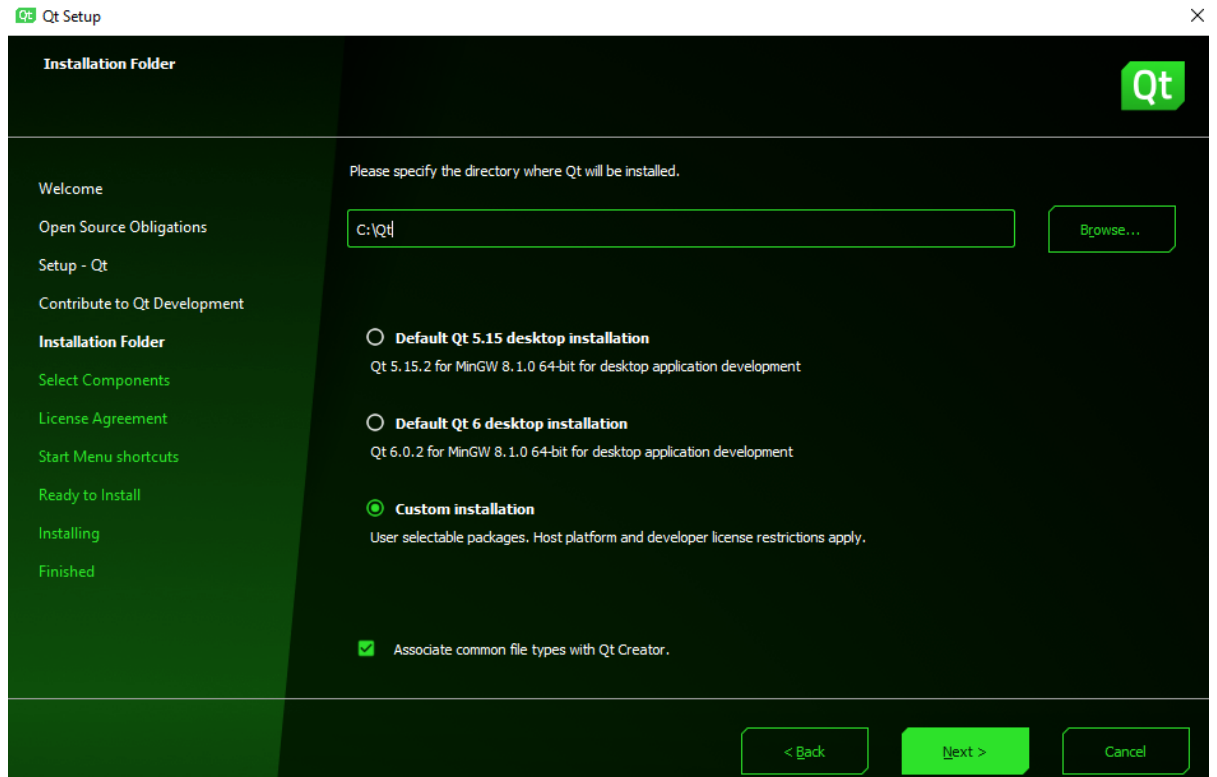
I am an individual person not using Qt for any company

⏪ Back   Next >   Cancel

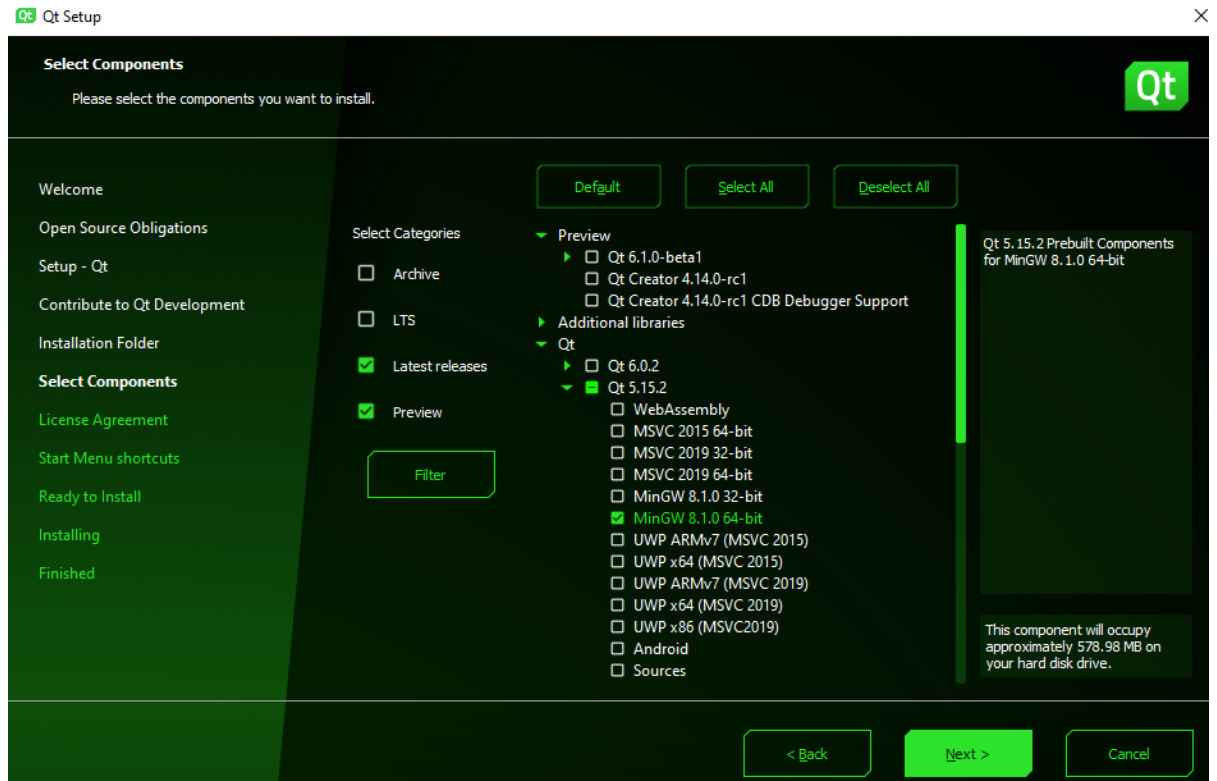




Dans la page “Installation Folder” où vous pouvez définir le chemin d’installation (choisissez un chemin simple d’utilisation, comme C:\Qt), sélectionnez “Custom installation”.

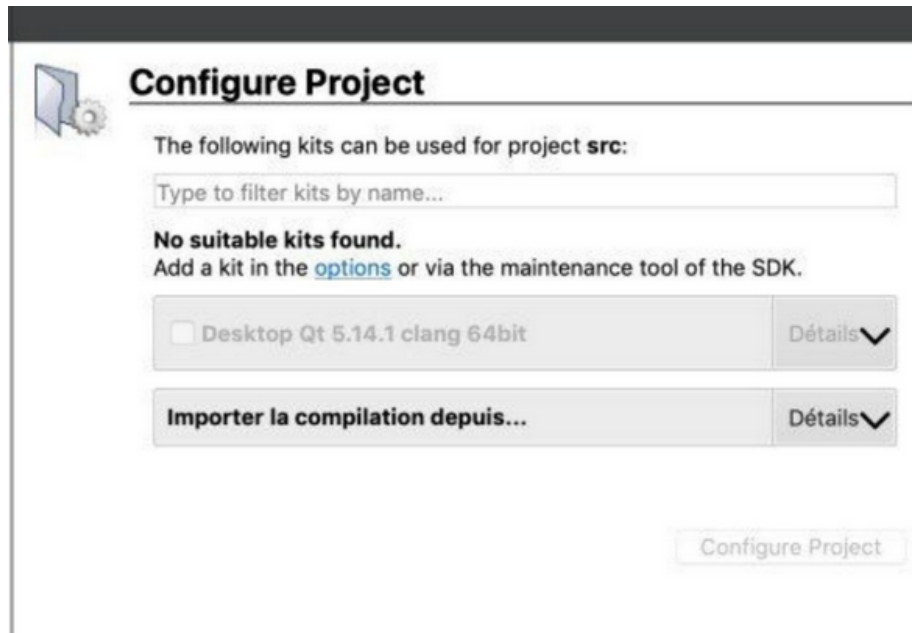


Lorsque lors de l'installation vous arrivez à un écran "Sélectionner des composants", sélectionner une version de Qt et cochez la case du compilateur que vous avez choisi (MinGW dans notre cas, en prenant soin de ne pas toucher aux cases sélectionnées par défaut)(voir les deux figures ci-dessous). Veillez à bien sélectionner une version de MinGW compatible avec votre installation SFML! (64/32 bits) *Attention: la sélection du package MinGW est à faire sous un menu lié à une version de QT (de la forme "Qt 5.xx.x") et non sous "Developer and Design Tools"*



## Problèmes potentiels

Il se peut que vous rencontriez parfois un souci suite à la configuration de votre premier projet. Lorsque vous tentez de sélectionner un kit pendant la création de projet, il peut arriver que cela soit tout simplement impossible et que vous ne parveniez pas à cocher la case du kit en question. QtCreator vous avertira donc : `No suitable kits found`, comme sur l'image ci-dessous.



Si vous êtes face à cette situation, il vous faut cliquer sur le lien bleu “options” et aller dans les paramètres de kit. Une fois dans cette fenêtre, sélectionner le kit en question et QtCreator devrait vous afficher lesdits paramètres juste en dessous de la zone de sélection de kit.

Dès lors vous devez vérifier 4 paramètres dans cette liste :

Les compilateurs C et C++ ("Compiler")

Le debugger ("Debugger")

Votre version de Qt ("Qt version")

Votre outil cmake ("CMake Tool")

Tous les 4 champs doivent posséder une valeur autre que “None” ! Pour ce qui est des valeurs précises que vous devez y trouver, cela dépendra beaucoup de votre OS ainsi que de l’endroit où vous avez installé Qt, cmake, ....

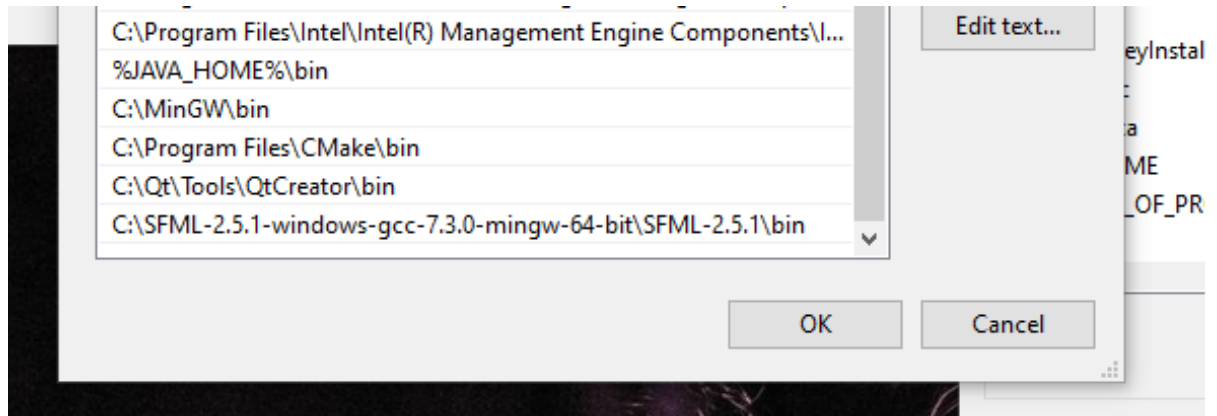
Pour régler votre problème, identifiez le/les champ/s contenant la valeur “None” (impliquant que QtCreator ne sait pas où trouver les outils dont il a besoin) et remplissez les avec les valeurs adéquates.

Dans quasiment tous les cas, les compilateurs seront automatiquement détectés et ne devraient pas poser de problèmes et il en va de même pour le debugger. Cependant, il se peut que vous n’ayez pas installé Qt ou cmake ou bien alors que QtCreator n’a pas réussi à les détecter tout seul sur votre système. Dans ce dernier cas assez pénible, il va vous falloir trouver l’emplacement des exécutables qmake et cmake respectivement. Encore une fois leur emplacement est bien trop dépendant de l’OS pour vous aiguiller ici mais vous pouvez tenter

de les chercher sur le Finder sous Mac et l'explorateur de fichier sous Windows (chercher `qmake.exe` et `cmake.exe` dans ce dernier cas).

Une fois les fichiers trouvés, il ne vous reste plus qu'à dire à QtCreator leur emplacement en remplaçant le "None" par le bon path. A partir de là le kit devrait être sélectionnable et utilisable pour la compilation !

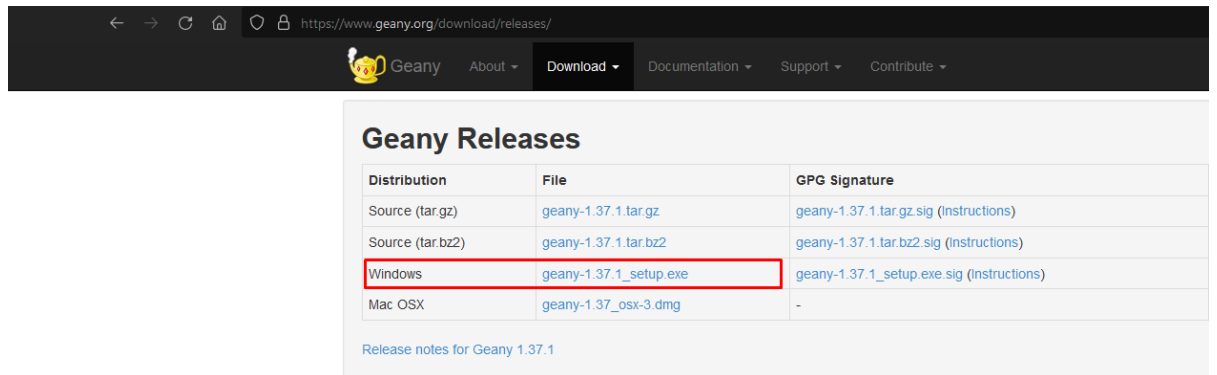
Pour Windows, assurez-vous que les *environment variable PATH* pour MinGW , CMake , QtCreator et SFML soient bien définis.



Le problème est quasiment toujours `Qt\Tools\QtCreator\bin` qui manque.

## Installation de l'éditeur de texte Geany (pas nécessaire si Qt Creator est choisi comme outil)

- télécharger l'installateur Windows à l'adresse <https://www.geany.org/Download/Releases> ;
- lancer l'installation.



Distribution	File	GPG Signature
Source (tar.gz)	<a href="#">geany-1.37.1.tar.gz</a>	<a href="#">geany-1.37.1.tar.gz.sig (Instructions)</a>
Source (tar.bz2)	<a href="#">geany-1.37.1.tar.bz2</a>	<a href="#">geany-1.37.1.tar.bz2.sig (Instructions)</a>
Windows	<a href="#">geany-1.37.1_setup.exe</a>	<a href="#">geany-1.37.1_setup.exe.sig (Instructions)</a>
Mac OSX	<a href="#">geany-1.37_osx-3.dmg</a>	-

[Release notes for Geany 1.37.1](#)



## Welcome to Geany 1.37.1 Setup

Setup will guide you through the installation of Geany 1.37.1.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Next >

Cancel

### License Agreement

Please review the license terms before installing Geany 1.37.1.



Press Page Down to see the rest of the agreement.

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

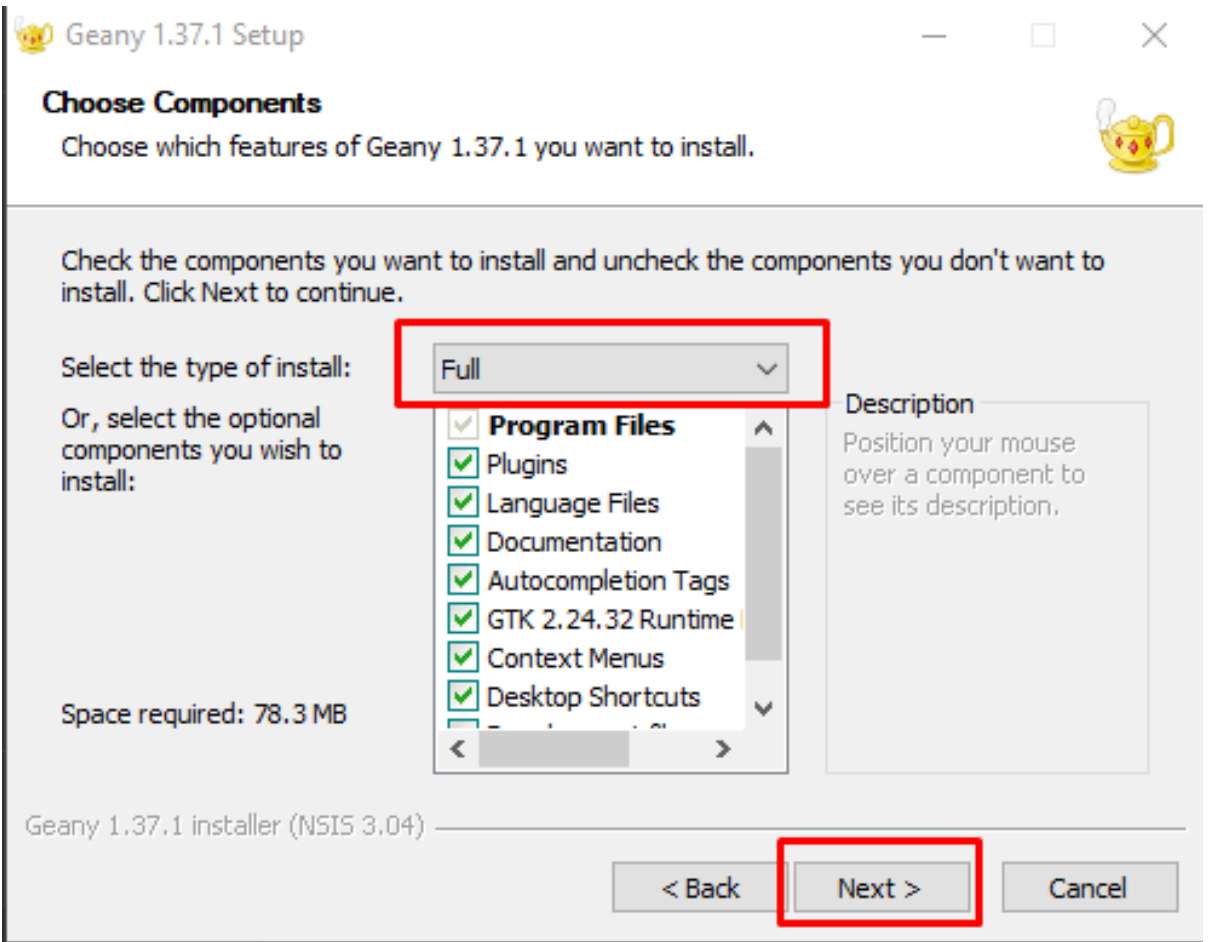
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Geany 1.37.1.

Geany 1.37.1 installer (NSIS 3.04)

< Back

I Agree

Cancel





Geany 1.37.1 Setup



### Choose Install Location

Choose the folder in which to install Geany 1.37.1.



Setup will install Geany 1.37.1 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Program Files (x86)\Geany

Browse...

Space required: 78.3 MB  
Space available: 21.8 GB

Geany 1.37.1 installer (NSIS 3.04)

< Back

Next >

Cancel

