

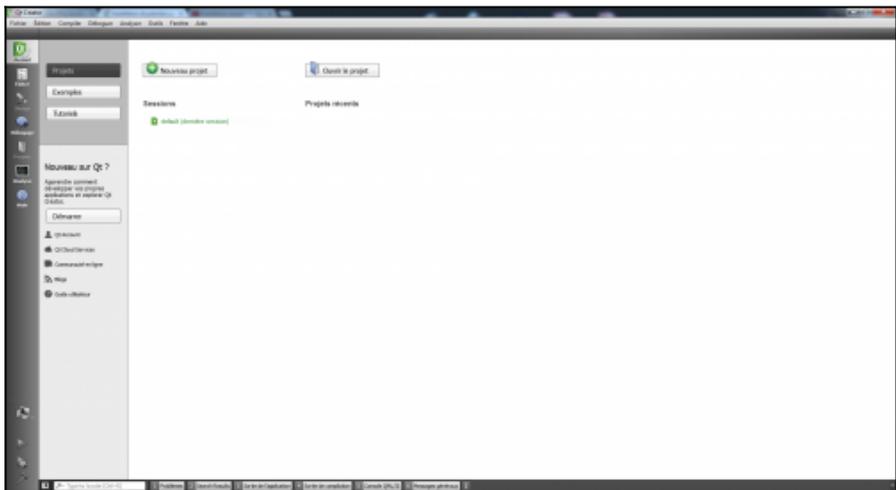
[Revenir à la page principale](#)

Tester l'installation de Qt 5.5

Avec Qt Creator

La première chose à faire après avoir installé Qt est de tester si l'installation s'est correctement déroulée. Pour cela, nous allons simplement créer un programme par défaut et l'exécuter. Si tout se passe bien, le programme se lancera et affichera une fenêtre.

Lorsque vous lancez l'éditeur Qt Creator, vous arrivez sur la page d'accueil suivante :



Quelques éléments de vocabulaire relatifs à Qt Creator, pour bien comprendre les choses. Les icônes à gauche en haut permettent de choisir le mode :

- “Accueil”, la page actuelle ;
- “Éditer”, lorsque vous éditez un fichier ;

- “Design”, pour les éditeurs graphiques de Qt ;
- “Débogage”, pour corriger les programmes ;
- “Projet”, pour éditer les paramètres de compilation et d'exécution des projets ;
- “Analyse”, pour les outils d'analyse de performances ;
- “Aide”, pour les pages d'aide de Qt.

En dessous des icônes de mode (toujours dans la barre à gauche), une série d'icônes (actuellement grisée, puisqu'aucun projet n'est ouvert) permettent, de haut en bas :

- de choisir le kit à utiliser pour la compilation et l'exécution (voir la suite pour les explications sur les kits, comment les configurer et les utiliser) ;
- de lancer le programme en mode normal (le triangle) ;
- de lancer le programme en mode Debug (le triangle avec un insecte - *bug* en anglais) ;
- de simplement compiler le programme, sans le lancer (bien sûr, les boutons précédents pour lancer le programme le compilent dans un premier temps, avant de le lancer).

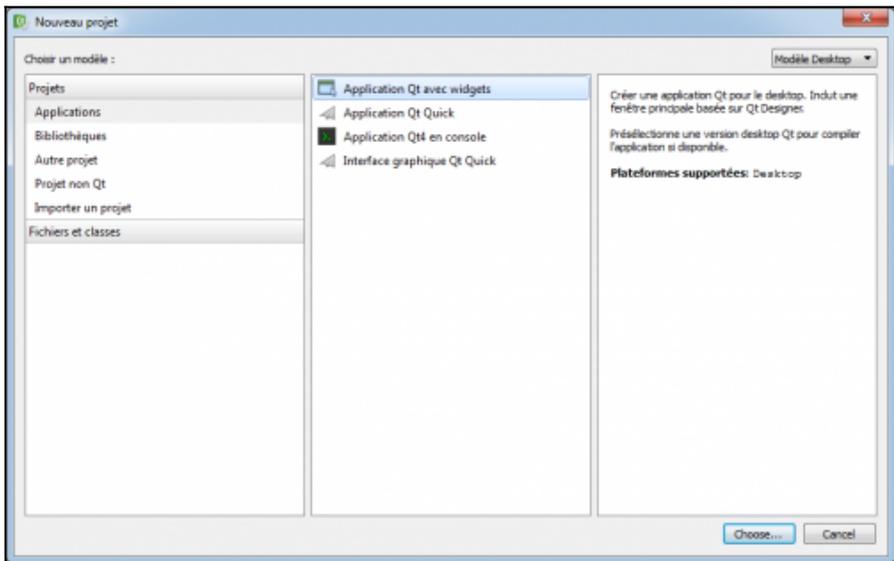
En bas, une série d'onglets permet d'ouvrir des fenêtres de messages. Lors de la compilation ou lorsqu'il y a un problème, les messages s'affichent dans ces fenêtres. De gauche à droite :

- le localisateur (Ctrl+K) pour rechercher des classes, variables ou fonctions dans le projet ;
- la fenêtre de problèmes, qui affiche les messages d'erreur ;
- la fenêtre de recherche, pour afficher le résultat des recherches (Ctrl+F pour une recherche dans le fichier courant, Ctrl+Shift+F pour rechercher dans plusieurs fichiers) ;
- la sortie d'application, qui affiche les messages de l'application (par exemple avec `std::cout` ou `qDebug()`) ;
- la sortie de compilation, qui affiche les commandes lancées lors de la compilation. Cette fenêtre vous sera particulièrement utile en cas de problème de configuration du projet (fichier non trouvé

par exemple) ;

- la console pour le QML et le JavaScript ;
- la fenêtre de messages généraux.

Pour créer un nouveau projet par défaut, vous pouvez aller dans le mode “Accueil” puis cliquer sur “Nouveau projet” ou aller dans le menu “Fichier” puis “Nouveau fichier ou projet...”. Un assistant vous permet de sélectionner le type de projet :



Il est possible de créer beaucoup de types de projet différents, il suffit de lire la description à droite pour savoir à quoi cela correspond.

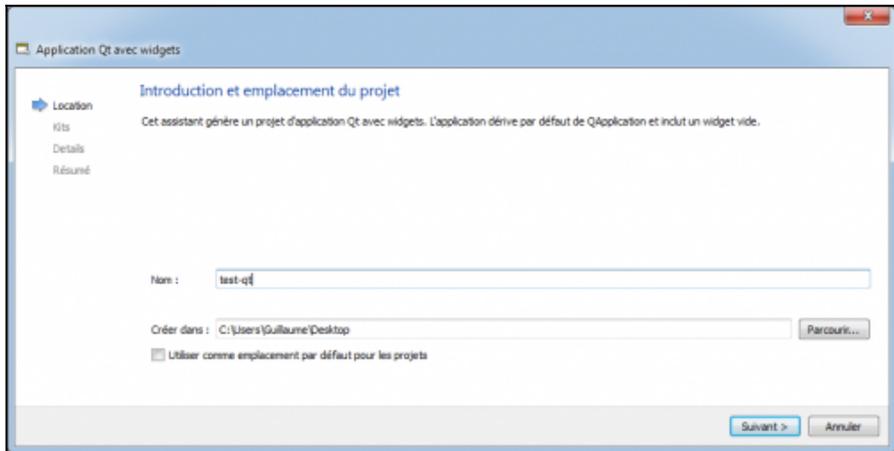
Pour les plus importants :

- “Application” puis “Application Qt avec widgets” pour les applications graphiques Qt classiques ;
- “Application” puis “Application Qt Quick” pour les applications graphiques Qt utilisant le nouveau langage de Qt : le QML ;
- “Application” puis “Application Qt console” pour les applications non graphiques Qt ;

- “Projet non Qt” puis “Projet C++” pour les applications C++ sans Qt ;
- “Importer un projet” pour créer un clone d'un projet existant dans un gestionnaire de versions (CVS, SVN, Git, etc.).

Choisissez “Application Qt avec widgets” pour ce premier test.

La page suivante permet de choisir le nom du projet que l'on souhaite créer et l'emplacement sur le disque. Remarque : ne mettez pas vos projets dans “C:\Qt”, créez un répertoire dédié pour cela, par exemple dans vos documents ou votre répertoire de travail.

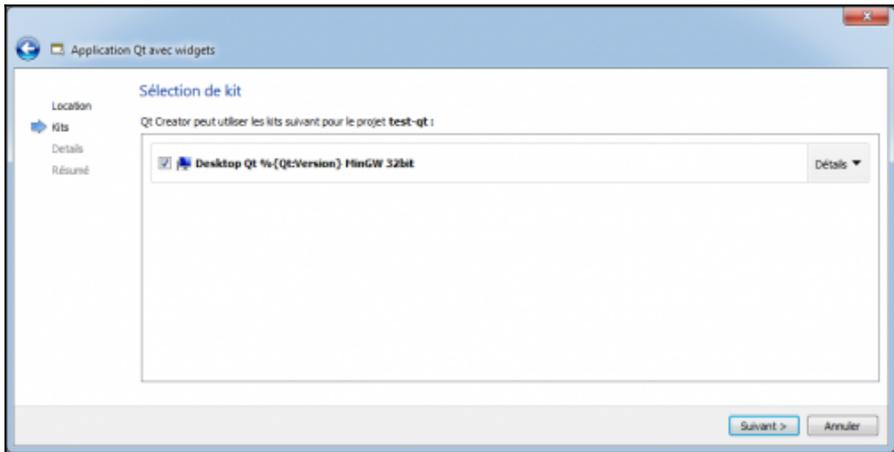


La page suivante permet de sélectionner les kits à utiliser pour compiler le programme. Il est possible de sélectionner plusieurs kits (voir la suite de ce tutoriel pour les explications sur les kits), pour le moment (si vous avez suivi les instructions de ce tutoriel et que c'est la première fois que vous installez Qt), vous n'avez qu'un seul kit disponible : “Qt MinGW”.

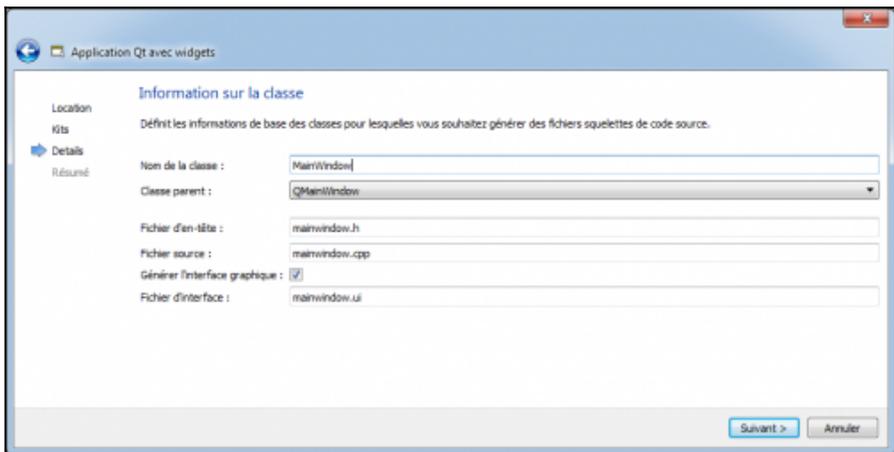
Si vous n'avez aucun kit disponible dans cette page, c'est qu'il y a eu un problème lors de l'installation (Qt Creator n'a pas réussi à trouver une version de Qt et un compilateur compatibles ensemble). Voir la suite de ce tutoriel pour les explications sur les kits.

Remarque : Qt 5.4 n'est pas encore totalement finalisé, il est possible

que vous ayez une erreur dans le message affiché, comme c'est le cas sur la copie d'écran. Rien de grave.

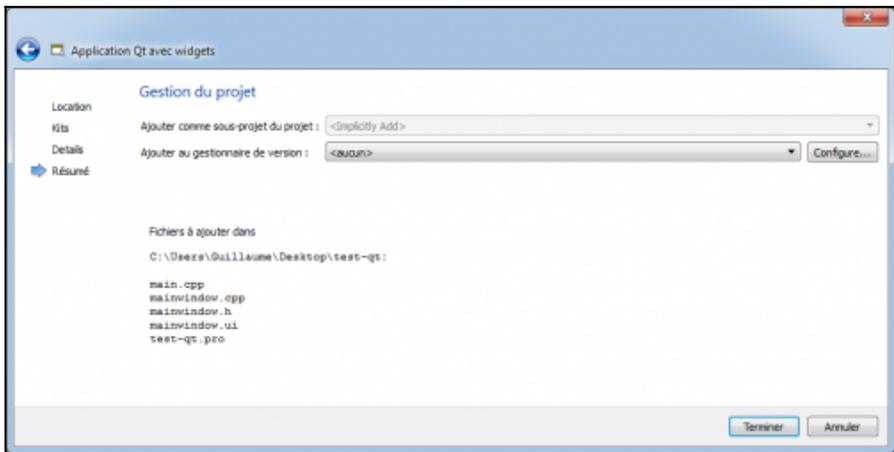


Le projet par défaut propose de créer une classe *MainWindow* (fenêtre principale). On va être gentil, on ne va pas le contrarier et le laisser faire. Cliquez sur Suivant.



Pour terminer, il est possible d'ajouter le projet dans un gestionnaire de versions. Cela n'est pas nécessaire pour ce test, mais n'hésitez pas à utiliser un tel gestionnaire, c'est très pratique et utile.

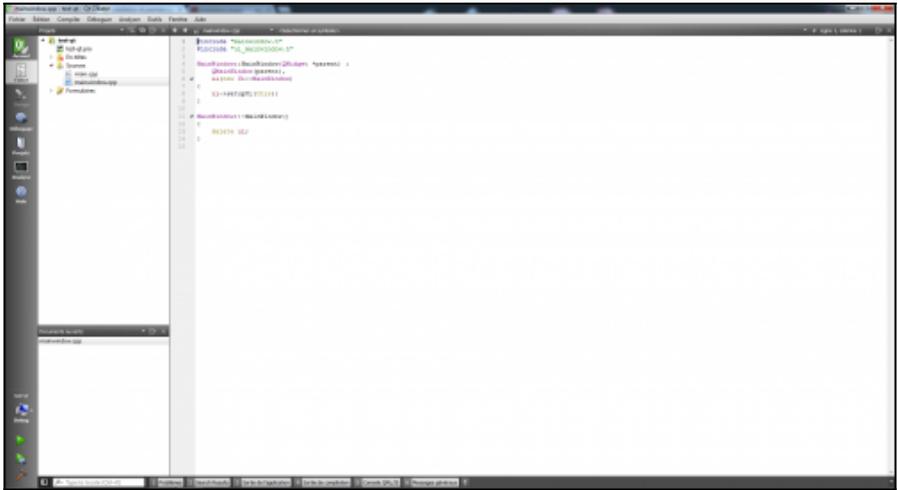
Cliquez sur “Terminer” pour créer le projet.



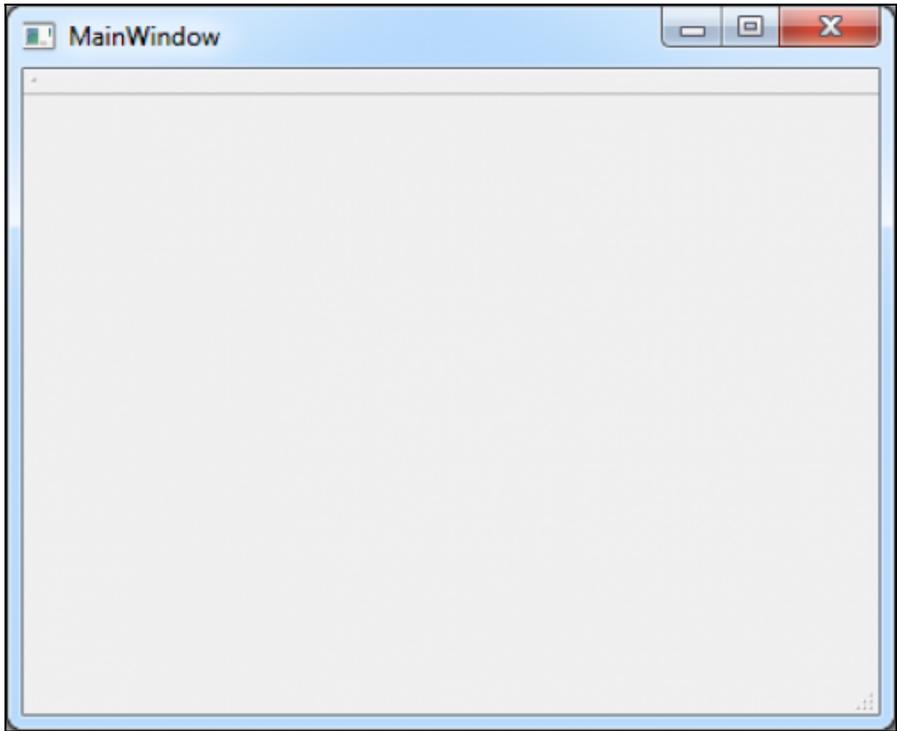
Qt Creator crée plusieurs fichiers et passe en mode “Éditer” pour afficher le contenu des fichiers. Un projet Qt contient généralement les fichiers suivants (cela peut changer en fonction du type de projet) :

- un fichier de projet `.pro` ou `.qmlprojet`, qui contient les informations sur le projet (en particulier la liste des fichiers et les modules Qt à utiliser) ;
- les fichiers C++ d'en-tête (`.h`) et d'implémentation (`.cpp`). En particulier, le projet contient le fichier `main.cpp`, qui est le point de démarrage du programme ;
- les fichiers de formulaire `.ui`.

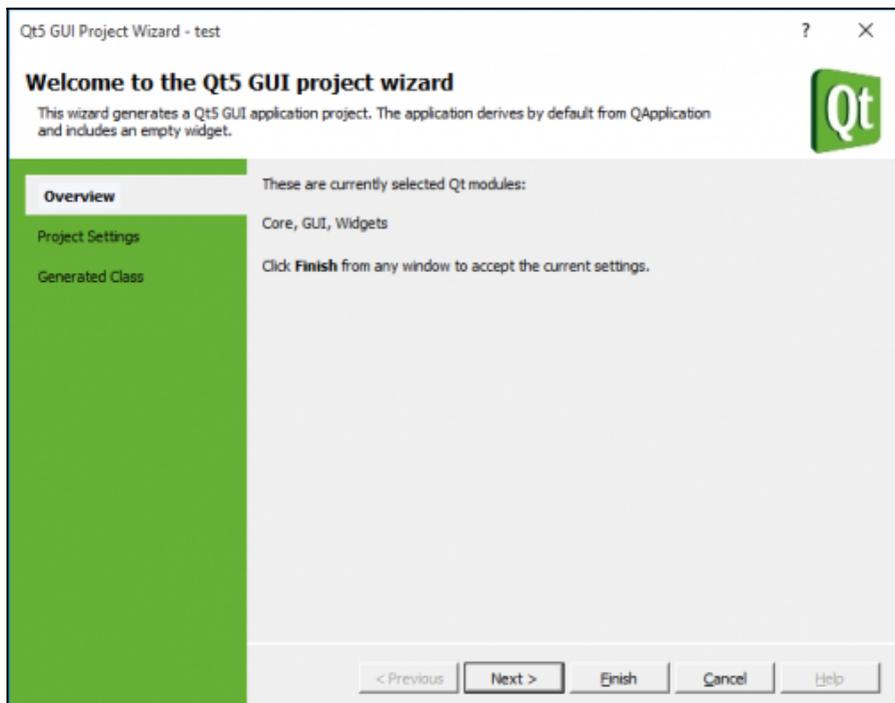
Cliquez sur les différents fichiers pour voir comment Qt Creator les affiche. Par exemple, les fichiers `.h` et `.cpp` sont affichés dans un éditeur de texte avec coloration syntaxique. Les fichiers `.ui` sont affichés en utilisant un éditeur graphique (mode Design).



Cliquez sur le triangle vert en bas à gauche, dans le menu "Compiler" puis "Exécuter" ou appuyez sur Ctrl+R pour lancer le programme. Si tout s'est bien passé, une fenêtre "MainWindow" devrait s'ouvrir.



Avec Visual Studio 2013



Welcome to the Qt5 GUI project wizard

This wizard generates a Qt5 GUI application project. The application derives by default from QApplication and includes an empty widget.



Overview

Project Settings

Generated Class

Select the modules you want to include in your project. The recommended modules for this project are selected by default.

- | | | |
|---|--|---|
| <input type="checkbox"/> 3D | <input type="checkbox"/> Multimedia Widgets | <input type="checkbox"/> SQL |
| <input type="checkbox"/> ActiveQt container | <input type="checkbox"/> Network | <input type="checkbox"/> SVG |
| <input type="checkbox"/> ActiveQt server | <input type="checkbox"/> OpenGL | <input type="checkbox"/> System Info |
| <input type="checkbox"/> Bluetooth | <input type="checkbox"/> Organizer | <input type="checkbox"/> Test |
| <input type="checkbox"/> Concurrent | <input type="checkbox"/> Phonon | <input type="checkbox"/> Versit |
| <input type="checkbox"/> Contacts | <input type="checkbox"/> Print Support | <input type="checkbox"/> WebKit |
| <input type="checkbox"/> Core | <input type="checkbox"/> Publish and Subscribe | <input type="checkbox"/> Webkit Widgets |
| <input type="checkbox"/> Declarative | <input type="checkbox"/> QML | <input checked="" type="checkbox"/> Widgets |
| <input checked="" type="checkbox"/> GUI | <input type="checkbox"/> Quick | <input type="checkbox"/> XML |
| <input type="checkbox"/> Help | <input type="checkbox"/> Script | <input type="checkbox"/> XML Patterns |
| <input type="checkbox"/> Location | <input type="checkbox"/> Sensors | |
| <input type="checkbox"/> Multimedia | <input type="checkbox"/> Service Framework | |

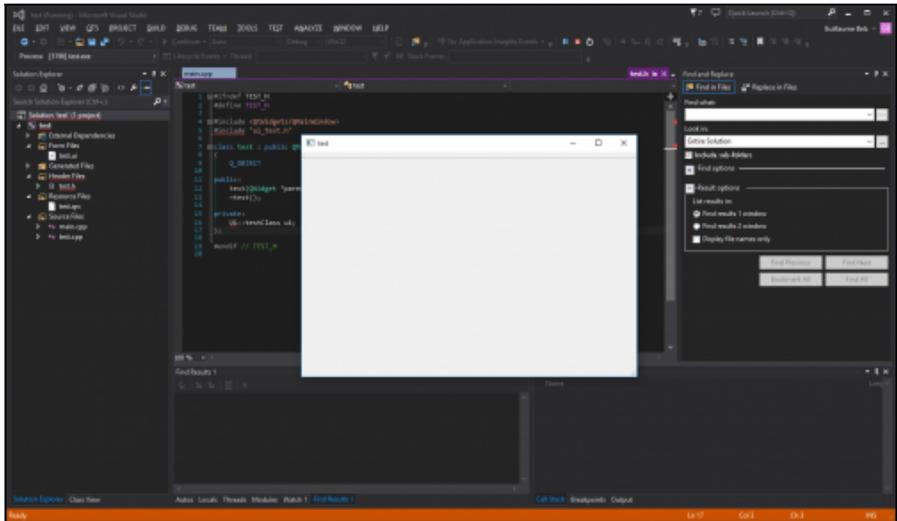
< Previous

Next >

Finish

Cancel

Help



[Revenir à la page principale](#)