

Configurer Qt Creator 3.4.2 pour Qt 5.5

Quelques points de vocabulaire

Juste pour faciliter les discussions sur les forums, voici quelques notions à avoir :

- **langage de programmation** : un programme est écrit dans un langage de programmation. Avec Qt, ce langage sera le C++ ou parfois Python (pour PyQt). Qt propose aussi deux autres langages : le QML et le JavaScript.
- **bibliothèque** (et pas “librairie”, traduction incorrecte du terme anglais “library”) : une bibliothèque est un ensemble d'outils pour étendre les fonctionnalités d'un langage. Pour le C++, vous connaissez probablement le STL (la bibliothèque standard du C++). Qt est aussi une bibliothèque (voire même un ensemble de bibliothèque, on parle alors de framework).
- **compilateur** : c'est le programme utilisé pour convertir votre code en programme exécutable. Le compilateur (ou plus précisément, l'ensemble des outils de compilation) seront généralement appelé automatiquement par l'IDE. Mais sachez qu'il est possible d'appeler soi-même ces outils.
- **éditeur** : c'est le logiciel que vous utilisez pour éditer vos fichiers. Un éditeur avancé proposera au moins la coloration syntaxique (afficher le code selon un code de couleurs, pour faciliter la lecture) et l'auto-complétion (proposer des syntaxes correspondant à ce que vous êtes en train d'écrire).
- **IDE** (ou EDI, selon l'humeur des gens, pour “Integrated Development Environment” ou “Environnement de Développement Intégré”) est un éditeur “intelligent”, qui

propose des outils pour faciliter le développement. Qt Creator est l'IDE fourni avec Qt, il permet en particulier de lancer directement la compilation, d'accéder à l'aide (en appuyant sur F1), etc.

- **SDK** (ou kit de développement) est un ensemble d'outils pour développer. Par exemple, lorsque vous téléchargez Qt, vous téléchargez en réalité le "Qt SDK", qui contient en particulier le framework Qt, le compilateur MingW ou GCC, l'IDE Qt Creator.

Donc, pour résumer, Qt n'est pas un langage. Ce n'est pas non plus un compilateur ou un éditeur. C'est simplement un framework. Ne confondez pas les choses et ne dites pas que vous avez écrit un programme en Qt (puisque ce n'est pas un langage - il faudrait dire "écrit un programme en C++ avec Qt").

Kits, compilateurs et versions de Qt

Donc, pour créer un programme, il faut :

- un compilateur qui convertit le code C++ en programme ;
- une version de Qt compatible avec le compilateur ;
- accessoirement, un débogueur ("accessoirement" parce que généralement, le débogueur est fourni avec le compilateur, pas parce que le débogueur est accessoire...).

L'IDE Qt Creator permet de gérer plusieurs compilateurs et versions de Qt en même temps. Cela est particulièrement utile si vous faites de la compilation vers Android ou iOS. Si vous avez suivi la procédure d'installation de Qt décrite ci-dessus, Qt Creator a normalement pu configurer correctement votre environnement de compilation et vous devriez pouvoir compiler directement une application.

Pour pouvoir gérer plusieurs configurations différentes, Qt Creator utilise un système de kits. Un kit est tout simplement l'association d'un compilateur, d'un débogueur et d'une version de Qt. Un kit est valide lorsque tous ces éléments sont compatibles. Si Qt Creator ne vous

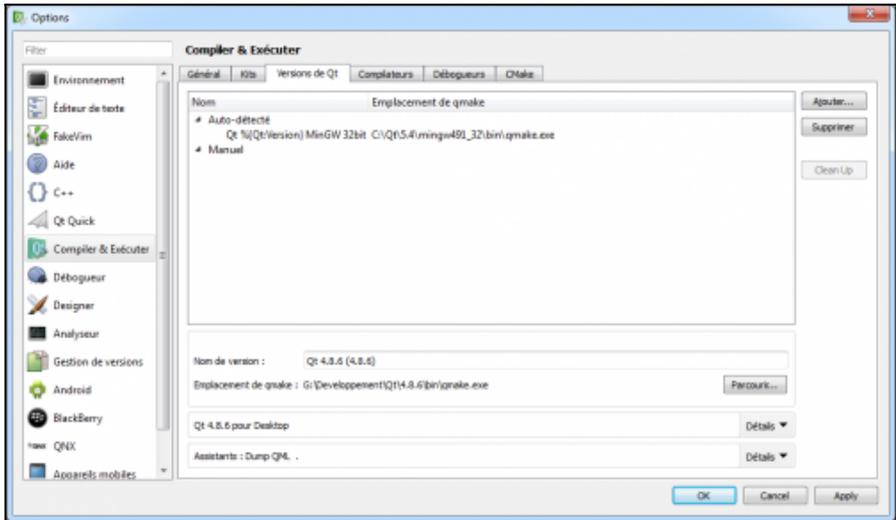
propose pas de kit lors de la création d'un projet, c'est peut-être parce que vous n'avez pas de kit valide (cela arrive souvent lorsque les gens installent Qt pour Microsoft Visual C++, mais sans installer ce compilateur).

Pour connaître les kits utilisables sur votre ordinateur et les configurer, allez dans le menu "Outils" puis "Options..." puis "Compiler & Exécuter". Ce dialogue possède des onglets qui vont nous intéresser :

- Kits ;
- Versions de Qt ;
- Compilateurs ;
- Débogueur.

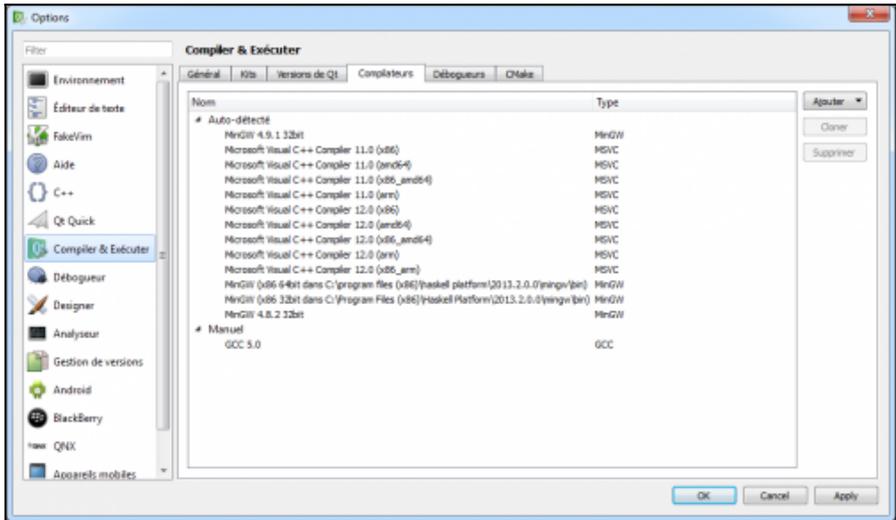
Comme vous l'avez sûrement compris, chacun de ces onglets permet de configurer les différents outils utilisés pour la compilation.

L'onglet `Versions de Qt` affiche les différentes versions de Qt installées. Une version de Qt est identifiée par un numéro de version de Qt (actuellement Qt 5.4, mais vous pouvez également avoir Qt 5.3, 5.2, etc.) et par le compilateur utilisé pour compiler Qt (MingW 4.9.1, MSVC 2013 32b OpenGL, etc.) Il faut compiler votre programme avec le même compilateur utilisé pour compiler Qt.



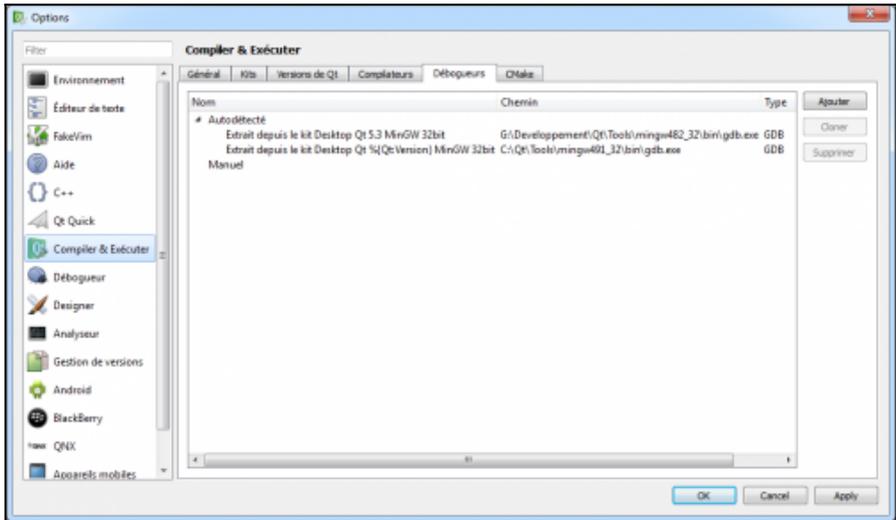
S'il manque une version de Qt, vous pouvez cliquer sur le bouton "Ajouter" et allez dans le répertoire de la version de Qt manquante, puis dans le sous-répertoire bin et sélectionner qmake. Par exemple, si vous avez suivi la procédure d'installation décrite ci-dessus et que Qt n'est pas reconnu, il faudra ajouter `C:\Qt\5.4\mingw491_32\bin`.

L'onglet `Compilateur` affiche la liste des compilateurs connus. Il faut bien sûr au moins un compilateur valide pour compiler un programme. Qt Creator trouvera les compilateurs installés dans les répertoires par défaut. Si ce n'est pas le cas, cliquez sur le bouton "Ajouter" et allez chercher l'application g++ (pour MingW et GCC), clang++ (pour LLVM/Clang) et cl.exe pour MSVC.



Les compilateurs se déclinent en plusieurs versions, il faudra bien choisir la version correspondant à la version de Qt utilisée. Le nom du compilateur correspondant à une version de Qt est indiqué dans la version de Qt.

Pour le Debugueur, si vous avez installé les compilateurs MingW, GCC ou Clang, le débogueur est inclus dedans et devrait être directement reconnu (s'il est installé dans un répertoire par défaut). Pour MSVC, il faut installer le Windows SDK en complément.



Un débogueur ne peut fonctionner qu'avec le compilateur correspondant. Si vous utilisez MSVC, vous ne pouvez pas utiliser GDB par exemple.

Une fois que vous avez vérifié que tous les outils sont individuellement reconnus par Qt Creator, vous pouvez configurer les kits. De la même façon que pour les autres outils, Qt Creator créera différents kits par défaut, s'il arrive à trouver des outils compatibles entre eux.

