

Installation et premiers pas avec Qt 5.4

Ce tutoriel est une reprise et une mise à jour des vidéos que j'avais fait pour [mon livre](#) sur la chaîne YouTube de mon éditeur, D-Booker, à l'occasion de la sortie de Qt 5.0 : [Installer le framework Qt 5](#). C'est également une reprise des vidéos que j'avais faite à l'occasion de la sortie de Qt 5.2, sur ma chaîne YouTube : [Installation et premier pas avec Qt 5.2 sur Windows](#) [Savoir utiliser la documentation de Qt](#). Je préfère passer au format tutoriel classique, pour faciliter les mises à jour.

La première partie est consacrée à l'installation de Qt sur Linux et Windows et la création de projets par défaut, pour vérifier que l'installation s'est bien passée. Il n'y a pas de version Mac, tout simplement par ce que j'en ai pas :)

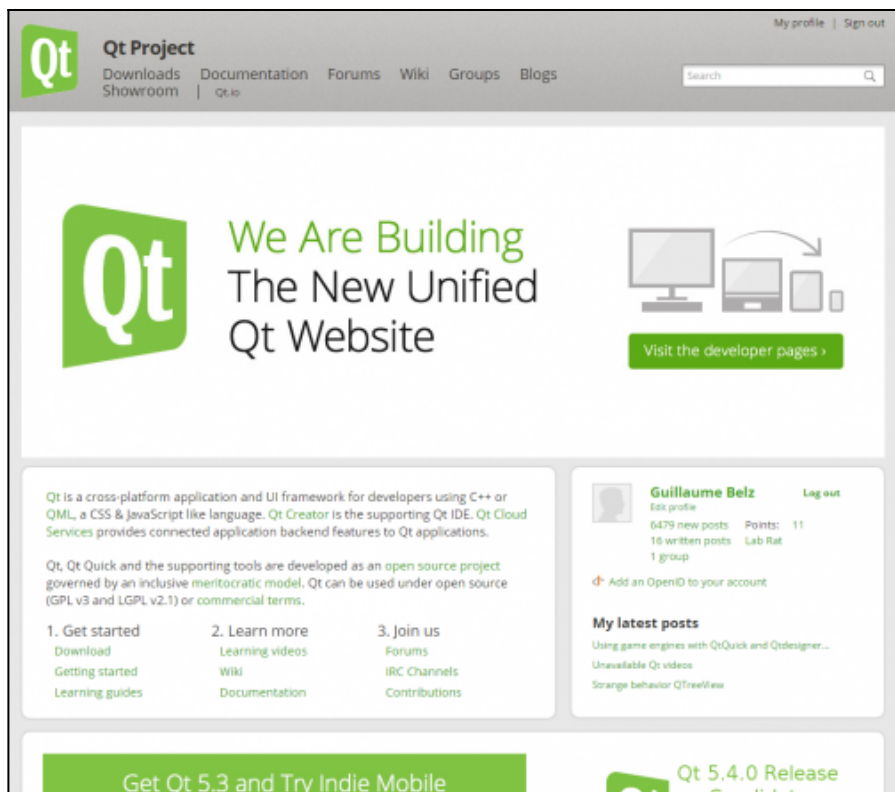
Dans la seconde partie, j'explique un peu plus en détail le fonctionnement de Qt et de son installation, pour aider ceux qui rencontre des problèmes lors de l'installation.

Pour terminer, je décris comment utiliser la documentation de Qt, qui est très bien faite et très riche. En particulier, il existe des codes d'exemple pour la majorité des fonctionnalités de Qt. Quand vous souhaitez réaliser quelque chose, la première chose à faire est probablement d'étudier ces codes d'exemple.

Installation de Qt

Télécharger l'installateur

Le téléchargement de Qt se fait sur le site du projet Qt : <http://qt-project.org/>.



Cette page d'accueil contient une série de liens en haut, en particulier :

- [Downloads](#), pour télécharger Qt ;
- [Documentation](#), pour les pages de documentation.

La page de téléchargement contient plusieurs versions de Qt. Il faut savoir que Qt existe sous licence libre GPL/LGPL (qui permet de créer des programmes libres ou commerciaux, avec quelques conditions d'utilisation) ou sous licence commerciale (qui donnent plus de libertés).


[Product](#)
[Services](#)
[Qt in Use](#)
[Developers](#)
[Licensing](#)
[Download](#)






Download Qt

Step 1

Get Qt in just a few steps. Determine your needs and select your license, then get ready to have your development world rocked!



Full Featured Evaluation

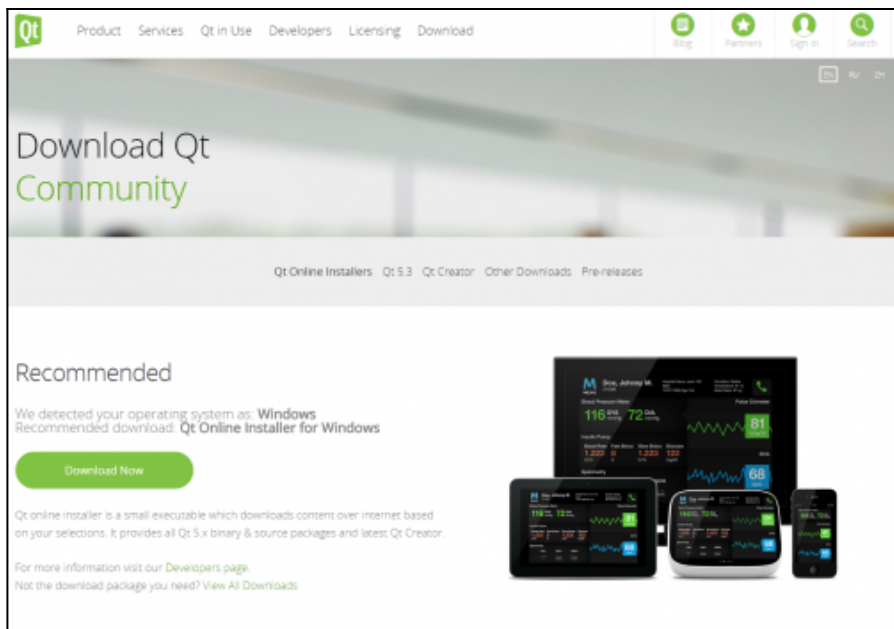
FREE 30-DAY TRIAL

Commercial Development

Download

	Community Projects compliant with GPL/GPL Download no license fee	Indie Mobile Mobile app development Buy Only 20K / \$25 per mo.	Professional Application Development Buy From 100K / \$148 per mo.	Enterprise Application development & Embedded device creation Contact Us Tailored offer - Perpetual
	Supported Platforms	Windows, Linux, Mac OS X, Android, iOS, WinRT, other embedded**	Android, iOS, Windows Phone, Windows RT	Windows, Linux, Mac OS X, Android, iOS, WinRT

Vous n'avez bien sûr pas besoin de télécharger la version commerciale. Cliquer sur le lien “Download” dans la colonne “Community”. Cela ouvre une nouvelle page :



Cette page reconnaît par défaut le système d'exploitation et propose la version *online*. Cliquer sur le lien pour lancer le téléchargement. La version *online* est en fait un simple petit programme (environ 13 Mo), qui vous permet de sélectionner ensuite les versions de Qt à installer (vous pouvez en installer plusieurs), les outils supplémentaires (l'éditeur Qt Creator, le compilateur MingW sous Windows, etc.).

Si vous ne souhaitez pas utiliser l'installateur proposé par défaut, vous pouvez cliquer sur le lien "View All Downloads". Cela permet d'afficher la liste de tous les téléchargement possible. Dans l'ordre :

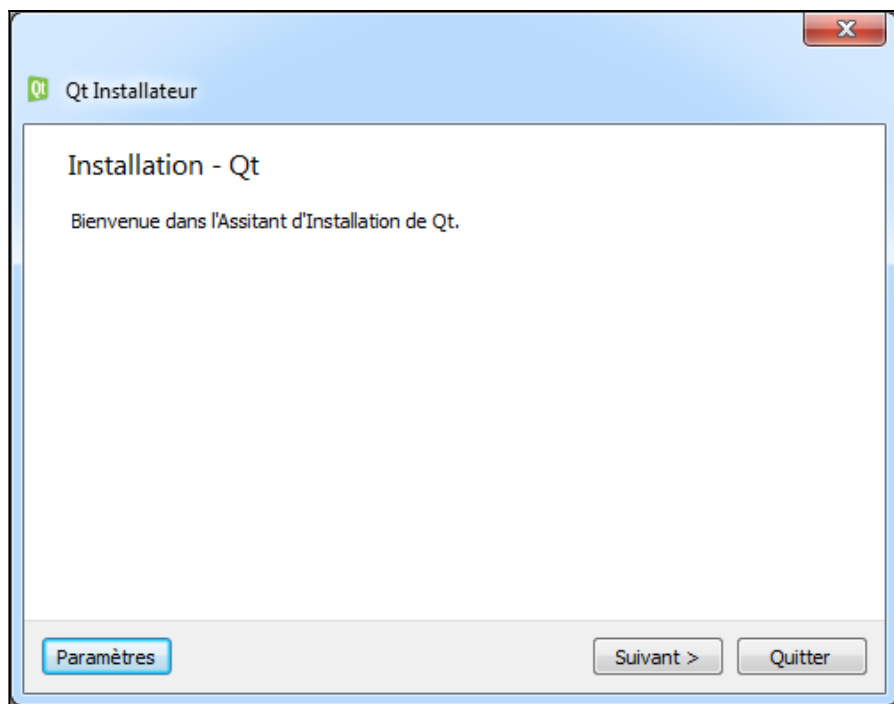
- Les installateurs *online* pour tous les systèmes (linux 32b et 64b, Mac et Windows) ;
- Les installateurs *offline* pour Linux, Mac et Windows ;
- Les sources de Qt (Qt étant un projet libre, ses sources sont librement accessibles) ;
- L'éditeur Qt Creator ;
- Les autres outils : le plugin pour Visual Studio, QBS, Qt Installer Framework, les archives des anciennes versions de Qt (ce

dernier lien permet en particulier de télécharger Qt 4 si nécessaire).

Dans ce tutoriel, nous verrons que l'utilisation de l'installateur *online* par défaut. Si vous n'utilisez pas la méthode par défaut, on va considérer que vous savez ce que vous faites...

Installer sous Windows

L'installateur `qt-opensource-windows-x86-1.6.0-6-online.exe` s'appelle `qt-opensource-windows-x86-1.6.0-6-online.exe`. Vous pouvez le lancer directement.

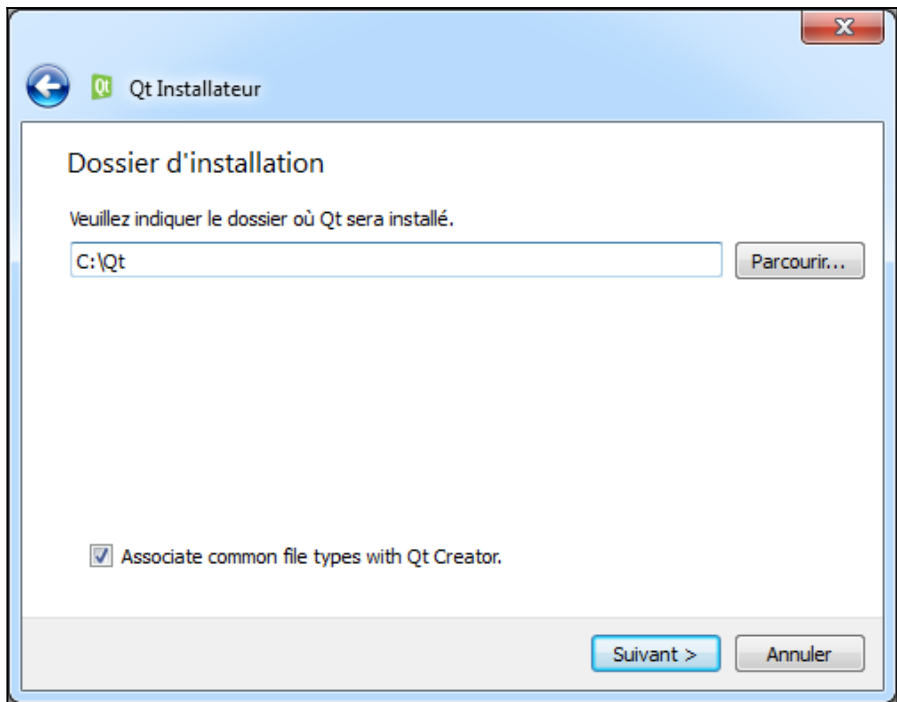


Le bouton “Paramètres” permet de configurer un serveur proxy si vous en utilisez un et d'ajouter des dépôts pour Qt. Normalement, vous n'avez

pas besoin de modifier les dépôts, le dépôt principal de Qt est configuré par défaut.

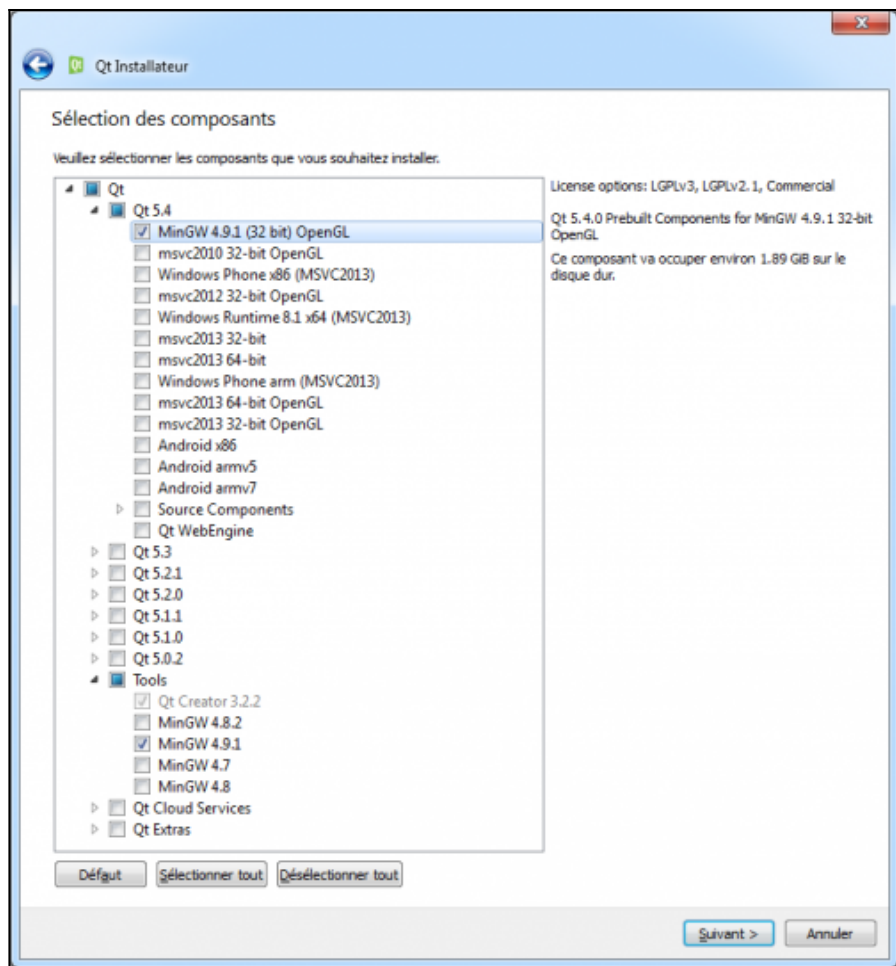
Après avoir cliqué sur “Suivant”, le programme recherche en ligne la liste des logiciels que vous pouvez installer (cela peut durer quelques dizaines de secondes à quelques minutes, en fonction de votre connexion).

La page suivante permet de choisir le dossier d'installation de Qt. Par défaut, le chemin est “C:\Qt”.



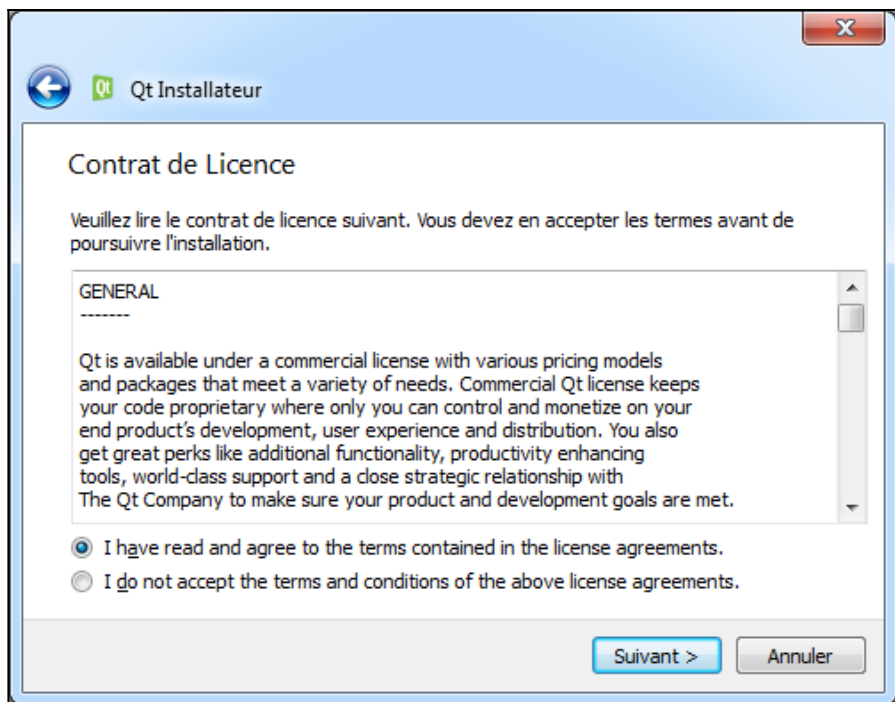
La page suivante permet de sélectionner la liste des outils à installer. Pour utiliser Qt, il faut installer trois outils :

- l'éditeur Qt Creator (il est installé automatiquement et il n'est pas possible de le désactiver) ;
- au moins une version de Qt ;
- un compilateur compatible avec la version de Qt installée.

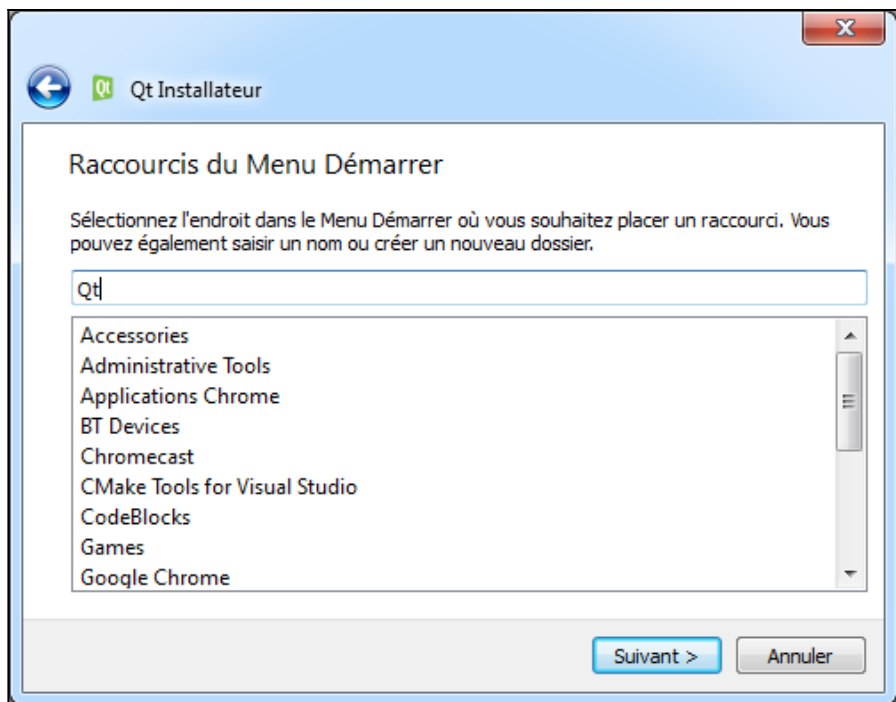


Par défaut, choisissez la dernière version de Qt (5.4) pour “MingW 4.9.1 (32 bit) OpenGL” et le compilateur “MingW 4.9.1”.

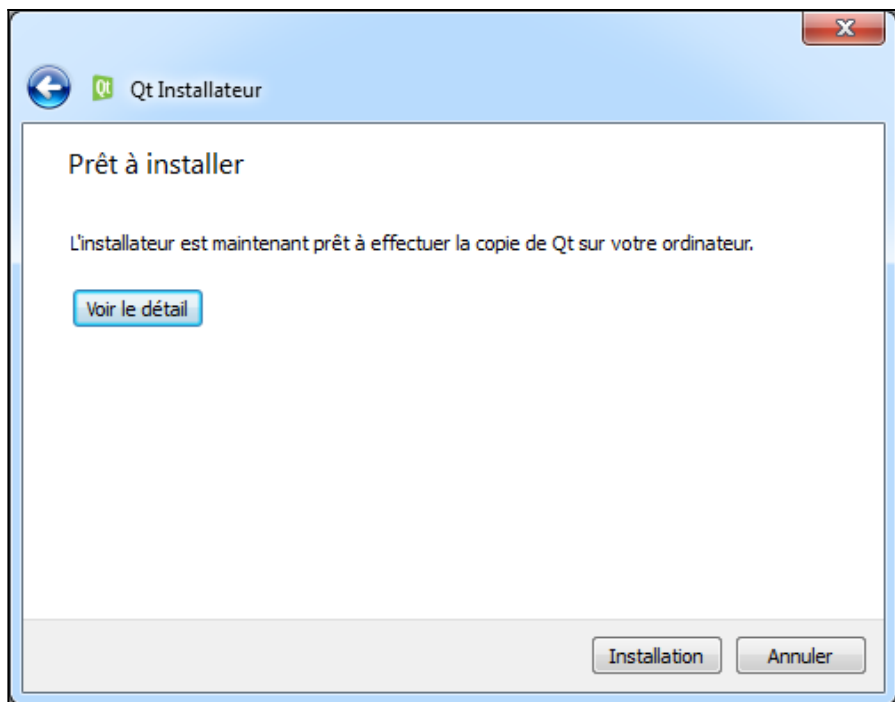
La page suivante permet de valider les licences utilisateurs. Cliquez sur suivant.



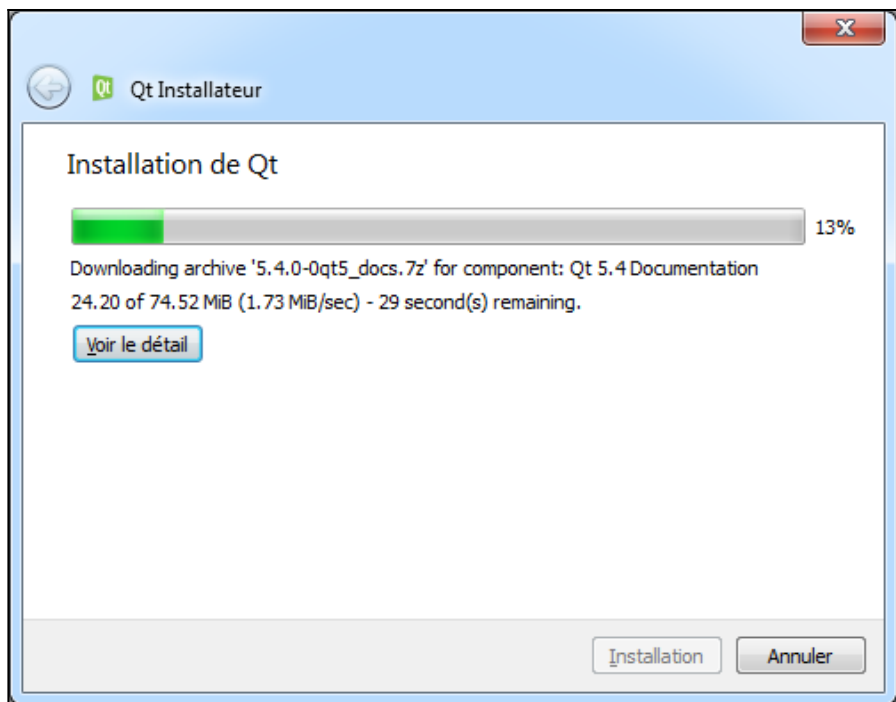
La page suivante permet de choisir le répertoire dans le menu Démarrer.



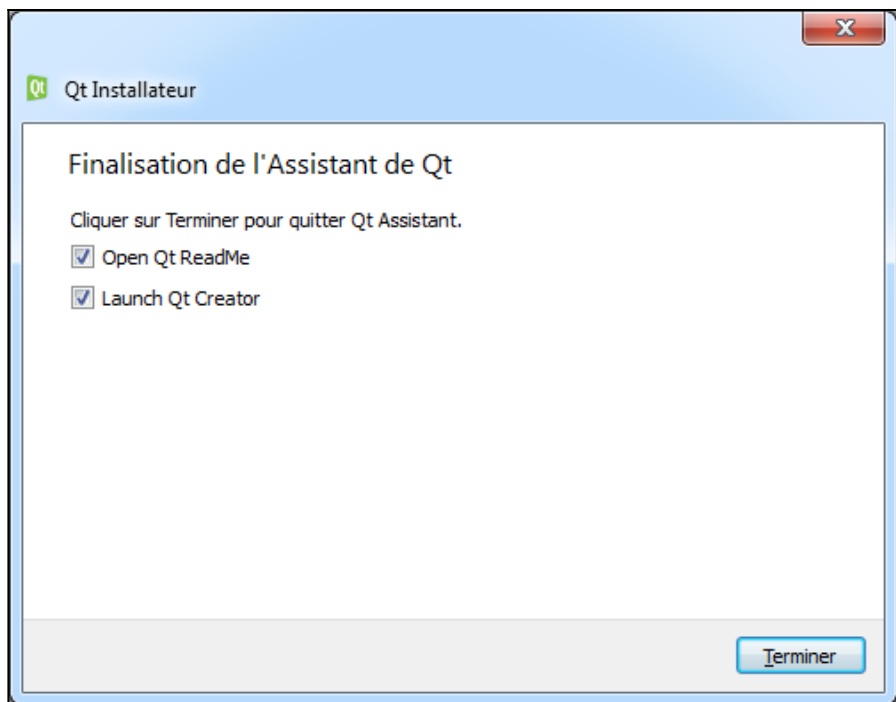
Une fois que tout cela est fait, l'installation est prête pour démarrer.



Le téléchargement puis l'installation se lancent. Selon votre connexion et le nombre de paquets que vous installez, cela peut prendre plusieurs minutes à plusieurs heures (si vous souhaitez installer beaucoup de paquets, il est probablement préférable de répéter l'installation plusieurs fois). L'installation sature le processeur, ne vous étonnez pas trop si Windows devient un peu lent pendant ce temps là. Allez vous balader dehors, il fait beau.

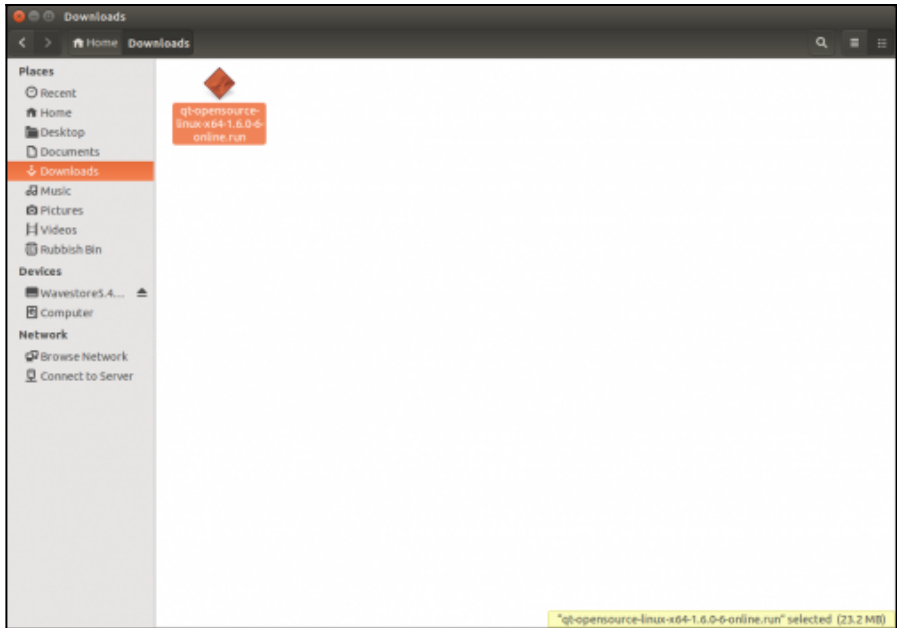


Une fois que l'installation est terminée, la page suivante propose d'ouvrir le fichier ReadMe (qui décrit les nouveautés de la dernière mise à jour de Qt) et de lancer Qt Creator. Vous pouvez décocher la case "Open Qt ReadMe" et cliquer sur Terminer pour lancer Qt Creator.



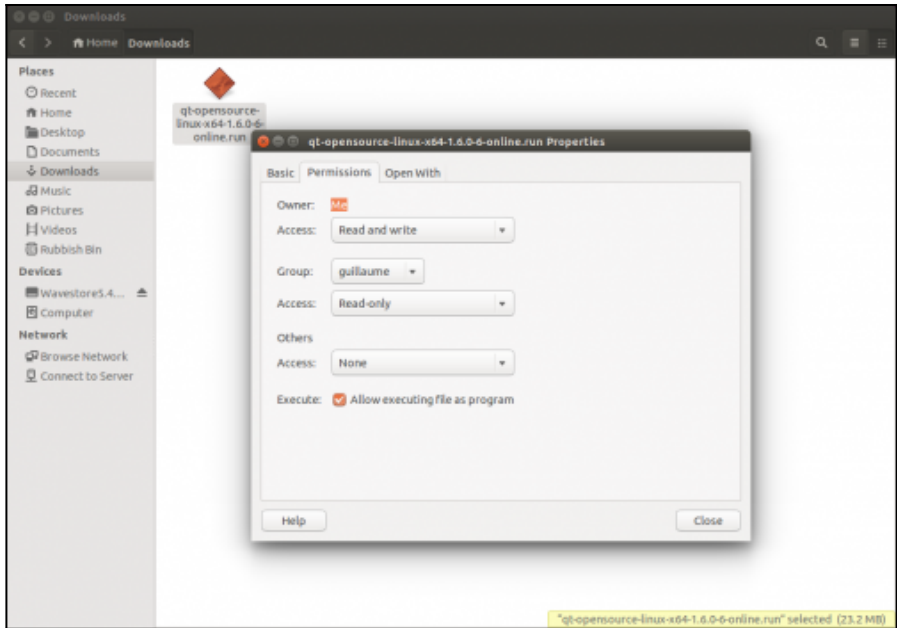
Installer sous Linux

L'installateur sous Linux s'appelle `qt-opensource-linux-x86-1.6.0-6-online.run` (pour 32-bit) ou `qt-opensource-linux-x64-1.6.0-6-online.run` (pour 64-bit). Par défaut, ce programme n'est pas exécutable, la première chose à faire est donc de modifier ses propriétés.

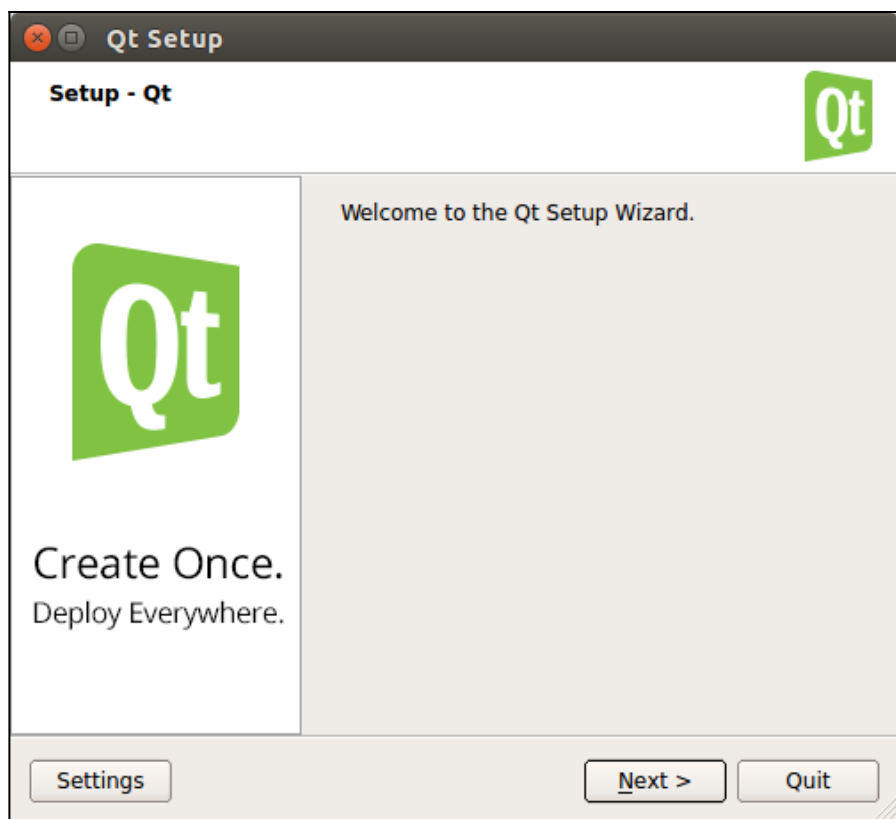


Cliquez avec le bouton droit de la souris et choisissez “Propriétés”. Dans l’onglet “Permission”, sélectionner “Allow executing file as program” (ou équivalent si vous avez une version en français). Vous pouvez également utiliser `chmod` en ligne de commande :

```
chmod +x qt-opensource-linux-x64-1.6.0-6-online.run
```

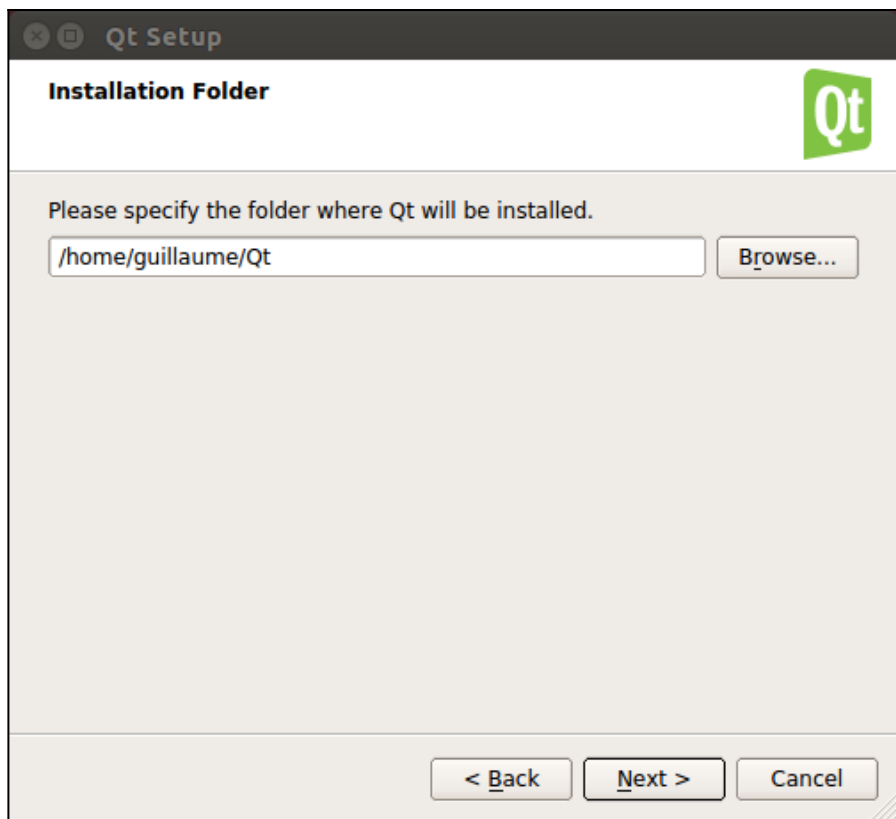


Vous pouvez ensuite lancer l'application. La première page contient un bouton “Settings” ou “Paramètres” pour configurer les dépôts ou un serveur proxy si nécessaire. Normalement, vous n'avez pas besoin de modifier les dépôts, le dépôt principal de Qt est configuré par défaut.



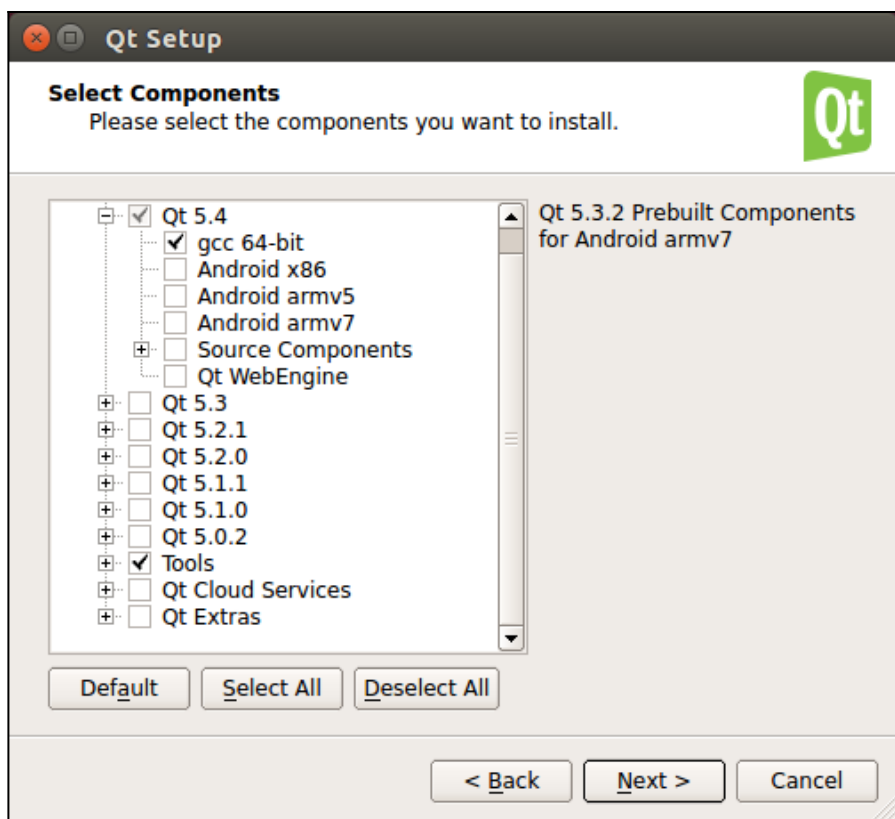
Après avoir cliqué sur “Suivant”, le programme recherche en ligne la liste des logiciels que vous pouvez installer (cela peut durer quelques dizaines de secondes à quelques minutes, en fonction de votre connexion).

La page suivante permet de choisir le dossier d'installation de Qt. Par défaut, le chemin est `"/home/username/Qt"`.



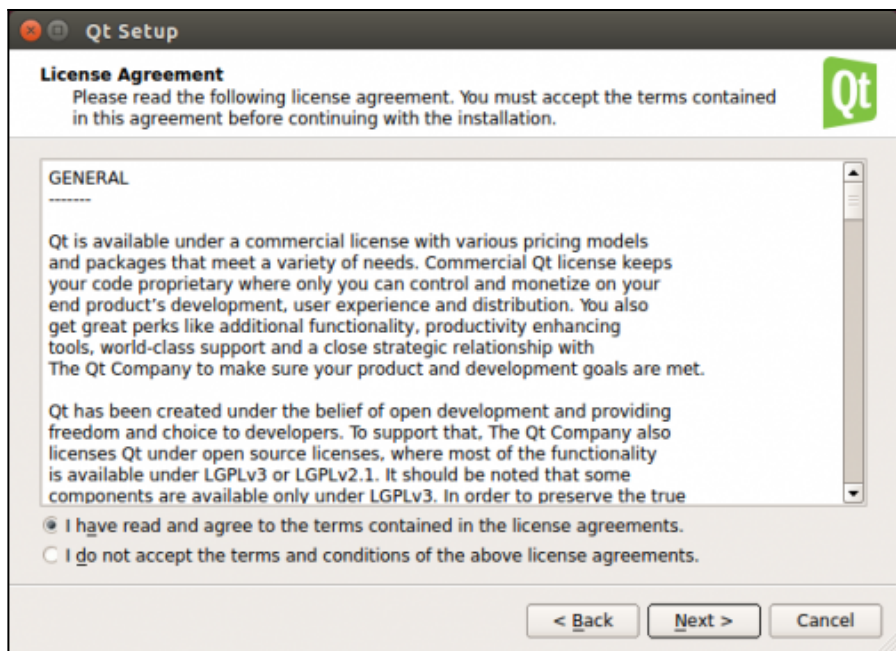
La page suivante permet de sélectionner la liste des outils à installer. Pour utiliser Qt, il faut installer trois outils :

- l'éditeur Qt Creator (il est installé automatiquement et il n'est pas possible de le désactiver) ;
- au moins une version de Qt .

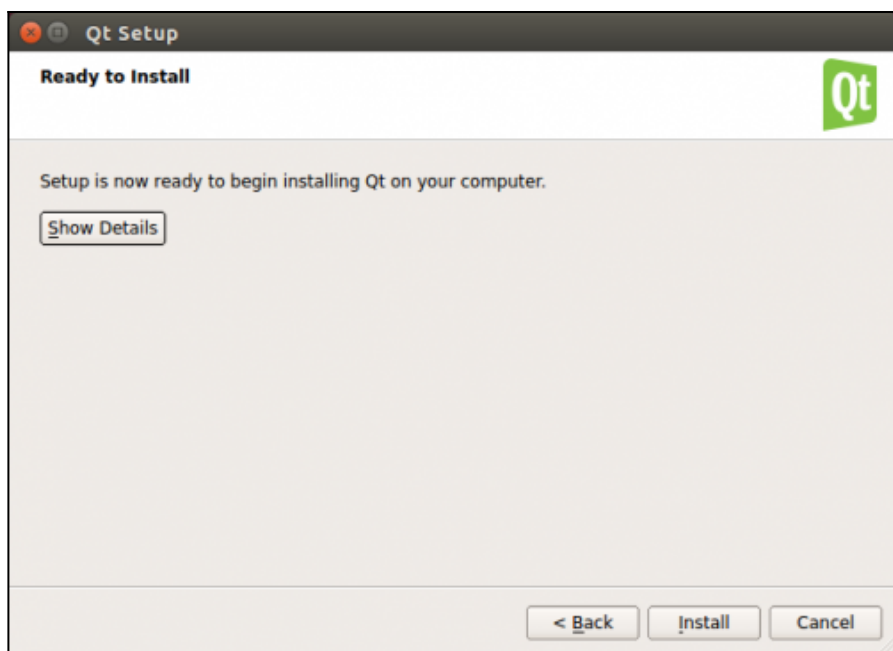


Par défaut, choisissez la dernière version de Qt (5.4) pour “GCC 32-bit” en 32-bit et “GCC 64-bit” en 64-bit.

La page suivante permet de valider les licences utilisateurs. Cliquez sur suivant.

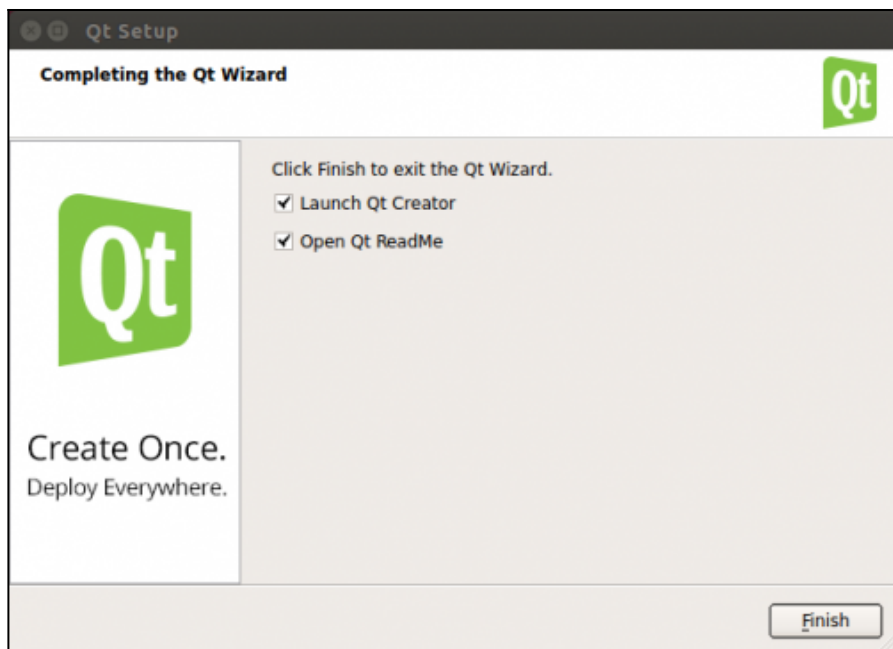


Une fois que tout cela est fait, l'installation est prête pour démarrer.



Le téléchargement puis l'installation se lancent. Selon votre connexion et le nombre de paquets que vous installez, cela peut prendre plusieurs minutes à plusieurs heures (si vous souhaitez installer beaucoup de paquets, il est probablement préférable de répéter l'installation plusieurs fois). L'installation sature le processeur, ne vous étonnez pas trop si Windows devient un peu lent pendant ce temps là. Allez vous balader dehors, il fait beau.

Une fois que l'installation est terminée, la page suivante propose d'ouvrir le fichier ReadMe (qui décrit les nouveautés de la dernière mise à jour de Qt) et de lancer Qt Creator. Vous pouvez décocher la case "Open Qt ReadMe" et cliquer sur Terminer pour lancer Qt Creator.



Pour Android et iOS

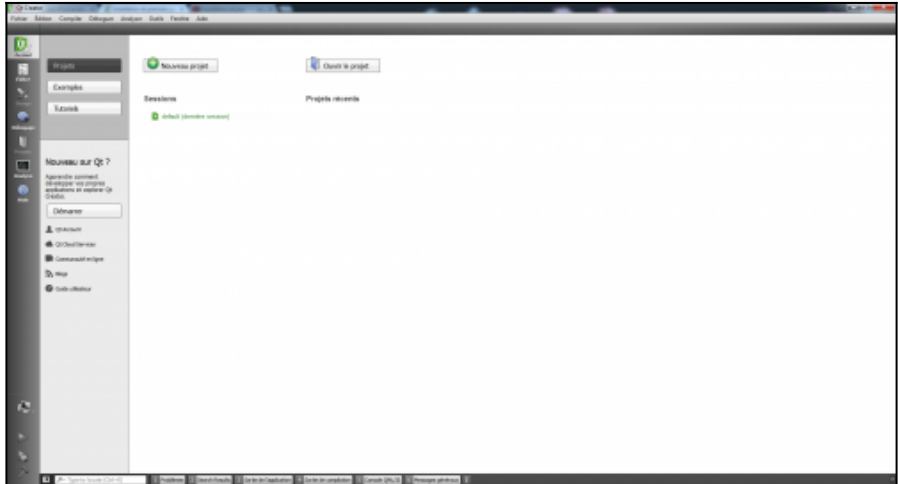
L'installation des outils pour développer des applications avec Qt pour Android et iOS est un peu plus compliqué (il est nécessaire d'installer les outils spécifiques de chaque plateforme, en plus de Qt). Pour Android, j'ai fait une série de tutoriel sur mon blog : <https://guillaumebelz.wordpress.com/category/android/>. Je ferais bientôt une mise à jour pour Qt 5.4; Pour iOS, je n'ai pas de Mac ni d'iPhone/iPad, je n'ai pas pu tester l'installation. Si quelqu'un utilise ces outils et qu'il souhaite compléter ce tutoriel, contactez moi.

Tester l'installation de Qt

La première chose à faire après avoir installer 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 relatif 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.

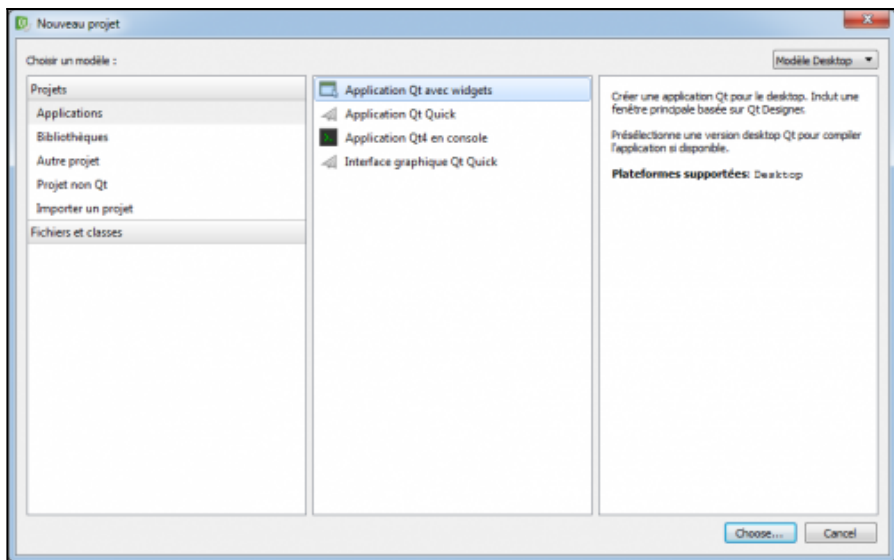
En dessous des icônes de mode (toujours dans la barre à gauche), une série d'icônes (actuellement grisés, puisque aucun projet n'est ouvert) permettent de haut en bas :

- 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) ;
- le simplement compiler le programme, sans le lancer (bien sûr, les boutons précédents pour lancer le programme le compile 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'erreurs ;
- 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.

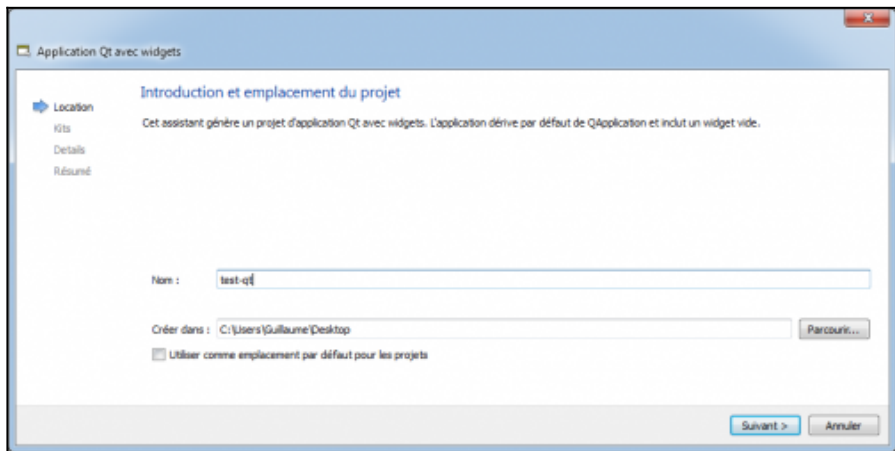
Pour les plus important :

- “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”, créer une répertoire dédié pour cela, par exemple

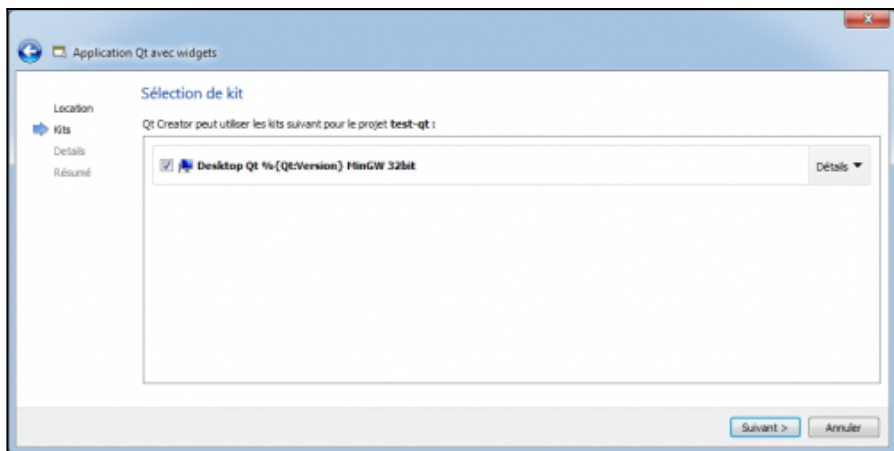
dans vos documents ou votre répertoire de travail.



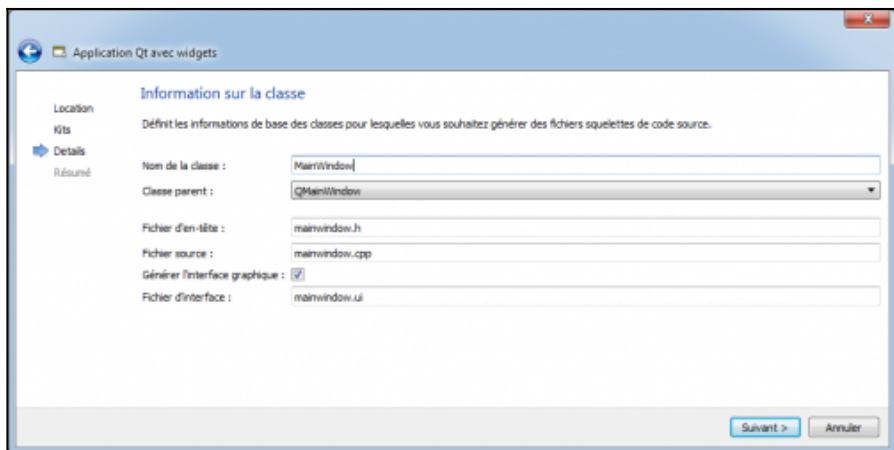
La page suivante permet de sélectionner les kits à utiliser pour compiler le programme. Il est possible de sélectionner plusieurs kits (voir la suite de ce tutoriel pour les explications sur les kits), pour le moment (si vous avez suivi les instructions de ce tutoriel et que c'est la première fois que 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 finalisée, 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.

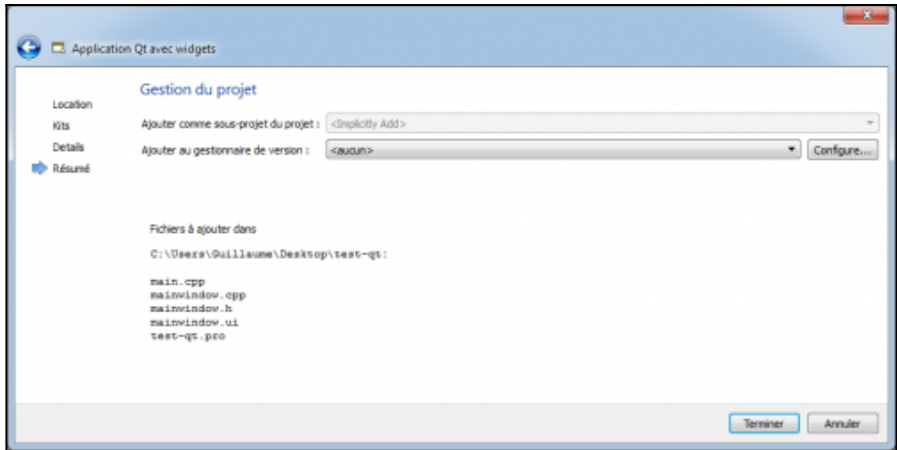


Le projet par défaut propose de créer une classe *MainWindow* (fenêtre principale). On va être gentil, on va pas le contrarier, on le laisse faire. Cliquez sur Suivant.



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.

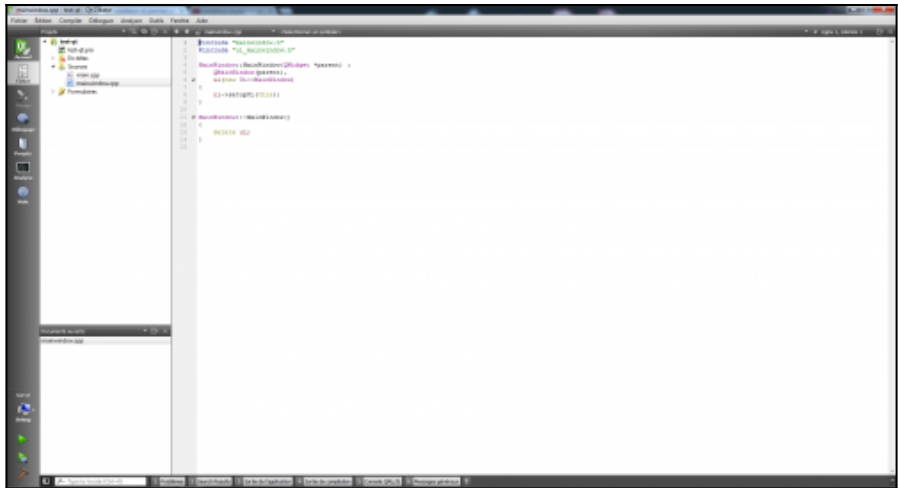
Cliquez sur Terminer pour créer le projet.



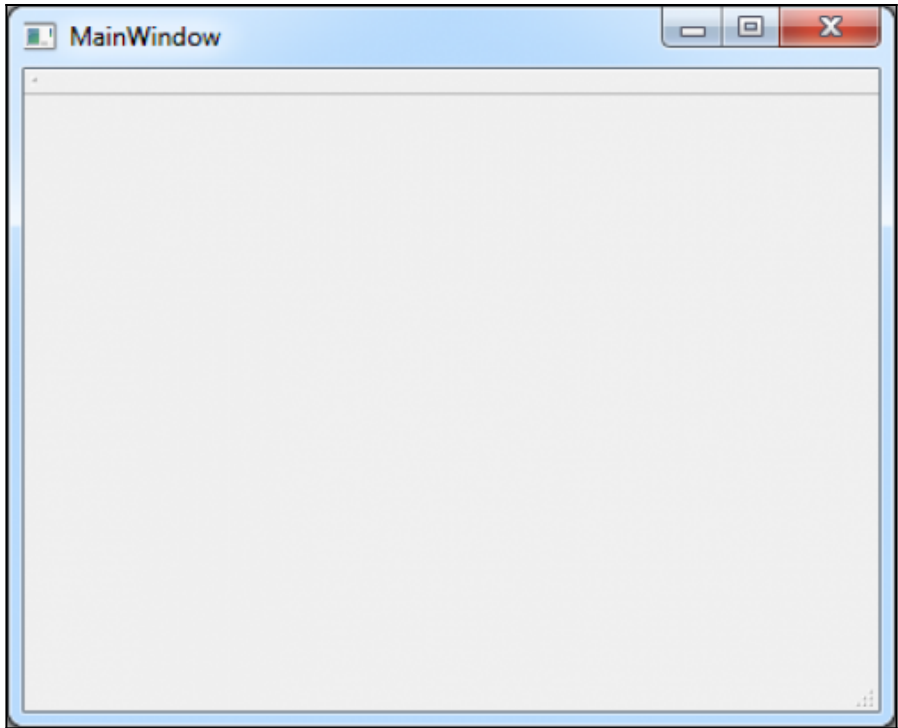
Qt Creator crée plusieurs fichiers et passe en mode “Éditer” pour afficher le contenu des fichiers. Un projet Qt contient généralement les fichiers suivants (cela peut changer en fonction du type de projet) :

- un fichier de projet `.pro` ou `.qmlprojet`, qui contient les informations sur le projet (en particulier le 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 vers 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.

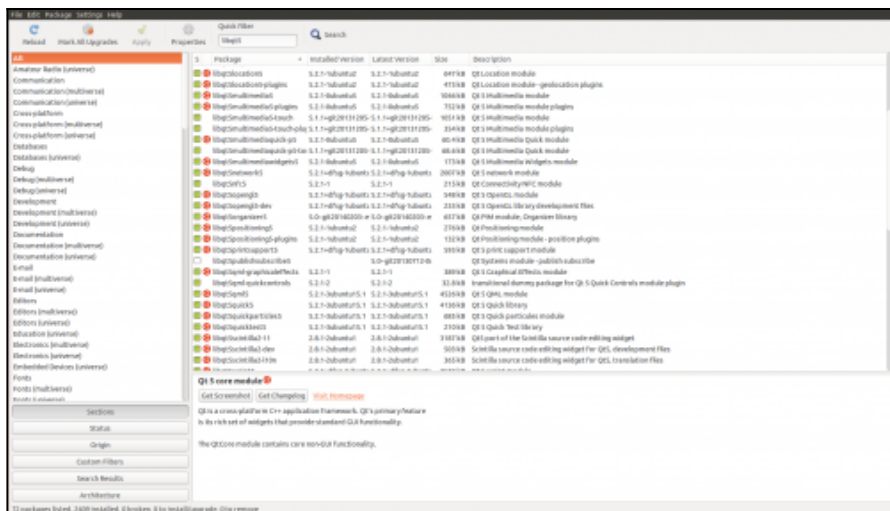


Installation alternative : Microsoft Visual C++

Plusieurs versions, allez sur <http://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx> et Visual Studio Express 2013 with Update 4 for Windows Desktop

Installer via les dépôts

Beaucoup de distributions Linux proposent Qt 5 directement dans les dépôts, vous pouvez donc commencer par là. Par exemple, avec Ubuntu, lancez synaptic et faites une recherche sur `libqt5` :



Comme vous pouvez le voir dans la copie d'écran, Ubuntu propose Qt 5, mais pas forcément Qt 5.4 (la version de Qt disponible dépendra de la distribution). Si vous n'avez pas besoin de la dernière version de Qt, vous pouvez installer celle des dépôts.

Vous pouvez également installer en ligne de commande, par exemple :

```
sudo apt-get install libqt5*
```

En complément, vous aurez besoin d'installer un compilateur C++. Vous pouvez installer GCC ou/et CLang, via synaptic ou en ligne de commande. Il est également possible d'installer d'autres méta-paquets, qui contiennent plusieurs outils de développement, comme les paquets "build-essential" ou "ubuntu-sdk".

Pour tester si vous avez un compilateur fonctionnel et quel est sa version, vous pouvez taper l'une des lignes suivantes :

```
g++ -v
clang++-3.5 -v
```

Par exemple, chez moi, cela retourne pour GCC les lignes suivantes. Vous pouvez voir à la dernière ligne que c'est la version 4.9.1 de GCC.

```
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/4.9/lto-wrapper
Target: x86_64-linux-gnu
Configured with: ../src/configure -v
--with-pkgversion='Ubuntu 4.9.1-16ubuntu6'
--with-bugurl=file:///usr/share/doc/gcc-4.9/README.Bugs
--enable-languages=c,c++,java,go,d,fortran,objc,obj-c++
--prefix=/usr --program-suffix=-4.9 --enable-shared
--enable-linker-build-id --libexecdir=/usr/lib
--without-included-gettext --enable-threads=posix
--with-gxx-include-dir=/usr/include/c++/4.9
--libdir=/usr/lib --enable-nls --with-sysroot=/
--enable-clocale=gnu --enable-libstdcxx-debug
--enable-libstdcxx-time=yes --enable-gnu-unique-object
--disable-vtable-verify --enable-plugin
--with-system-zlib --disable-browser-plugin
--enable-java-awt=gtk --enable-gtk-cairo
--with-java-home=/usr/lib/jvm/java-1.5.0-gcj-4.9-amd64/jre
--enable-java-home
--with-jvm-root-dir=/usr/lib/jvm/java-1.5.0-gcj-4.9-amd64
--with-jvm-jar-dir=/usr/lib/jvm-exports/java-1.5.0-gcj-4.9-amd64
--with-arch-directory=amd64
--with-ecj-jar=/usr/share/java/eclipse-ecj.jar
--enable-objc-gc --enable-multiarch
--disable-werror --with-arch-32=i686 --with-abi=m64
--with-multilib-list=m32,m64,mx32
--enable-multilib --with-tune=generic
--enable-checking=release --build=x86_64-linux-gnu
--host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 4.9.1 (Ubuntu 4.9.1-16ubuntu6)
```

Pour Clang :

```
Ubuntu clang version 3.5.0-4ubuntu2 (tags/RELEASE_350/final)
(based on LLVM 3.5.0)
Target: x86_64-pc-linux-gnu
Thread model: posix
Found candidate GCC installation:
```

```
/usr/bin/../../lib/gcc/i686-linux-gnu/4.9
Found candidate GCC installation:
/usr/bin/../../lib/gcc/i686-linux-gnu/4.9.1
Found candidate GCC installation:
/usr/bin/../../lib/gcc/x86_64-linux-gnu/4.8
Found candidate GCC installation:
/usr/bin/../../lib/gcc/x86_64-linux-gnu/4.8.3
Found candidate GCC installation:
/usr/bin/../../lib/gcc/x86_64-linux-gnu/4.9
Found candidate GCC installation:
/usr/bin/../../lib/gcc/x86_64-linux-gnu/4.9.1
Found candidate GCC installation:
/usr/lib/gcc/i686-linux-gnu/4.9
Found candidate GCC installation:
/usr/lib/gcc/i686-linux-gnu/4.9.1
Found candidate GCC installation:
/usr/lib/gcc/x86_64-linux-gnu/4.8
Found candidate GCC installation:
/usr/lib/gcc/x86_64-linux-gnu/4.8.3
Found candidate GCC installation:
/usr/lib/gcc/x86_64-linux-gnu/4.9
Found candidate GCC installation:
/usr/lib/gcc/x86_64-linux-gnu/4.9.1
Selected GCC installation:
/usr/bin/../../lib/gcc/x86_64-linux-gnu/4.9
Candidate multilib: .;@m64
Candidate multilib: 32;@m32
Candidate multilib: x32;@mx32
Selected multilib: .;@m64
```

Configurer Qt Creator

Quelques points de vocabulaire

Juste 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 utiliser pour éditer vos fichiers. Un éditeur avancé proposera au moins la coloration syntaxique (afficher le code selon un code de couleur, pour faciliter la lecture) et l'auto-complétion (proposer des syntaxes correspondant à ce que vous êtes entrain 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 fournit avec Qt, il permet en particulier de lancer directement la compilation, d'accéder au l'aide (en appuyant sur F1), etc.
- **SDK** (ou kit de développement) est un ensemble d'outils pour développer. Par exemple, lorsque vous télécharger 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 résumé, 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.

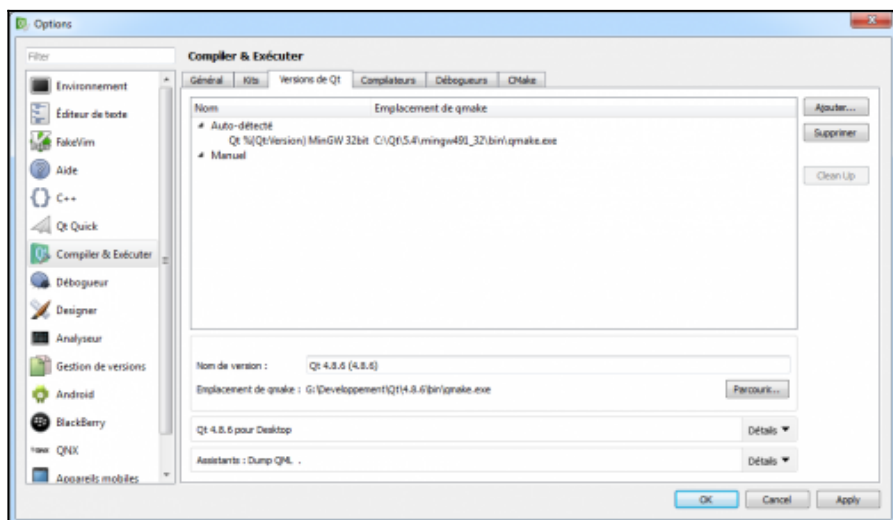
Pour pouvoir gérer plusieurs configurations différentes, Qt Creator utilise un système de kits. Un kit est tout simplement l'association d'un compilateur, d'un débogueur et d'une version de Qt. Un kit est valide lorsque tous ces éléments sont compatibles. Si Qt Creator ne vous propose pas de kit lors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

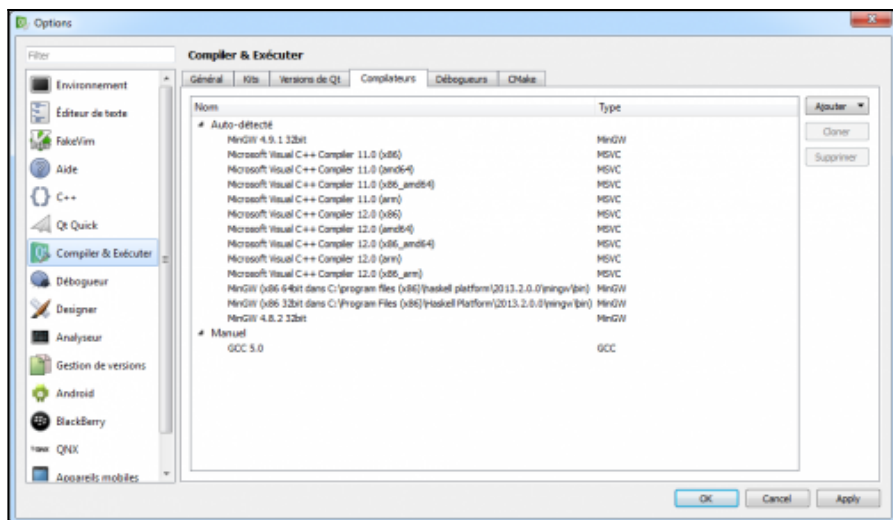
Comme vous l'avez surement compris, chacun de ces onglets permet de configurer les différents outils utilisés pour la compilation.

L'onglet **Versions de Qt** affiche les différentes versions de Qt installée. 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.



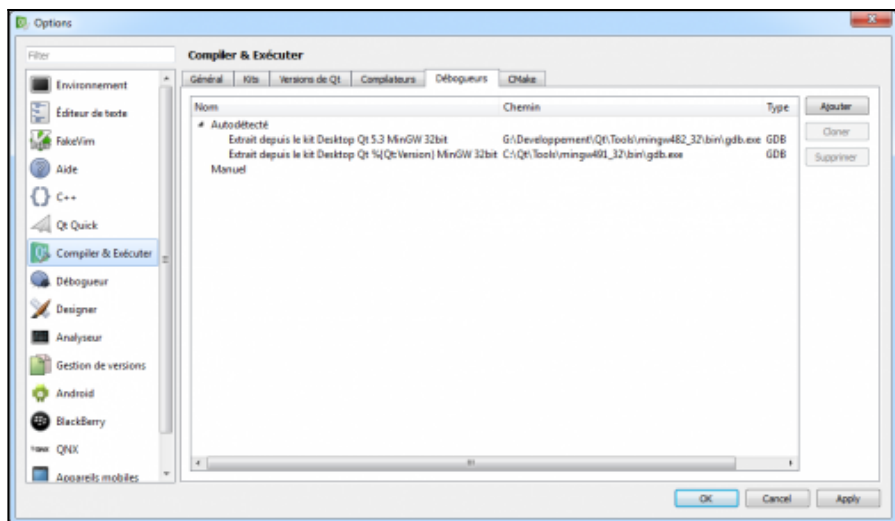
S'il manque une version de Qt, vous pouvez cliquer sur le bouton "Ajouter" et aller dans le répertoire de la version de Qt manquante, puis dans le sous-répertoire bin et sélectionner qmake. Par exemple, si vous avez suivi la procédure d'installation décrite ci-dessus et que Qt n'est pas reconnu, il faudra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



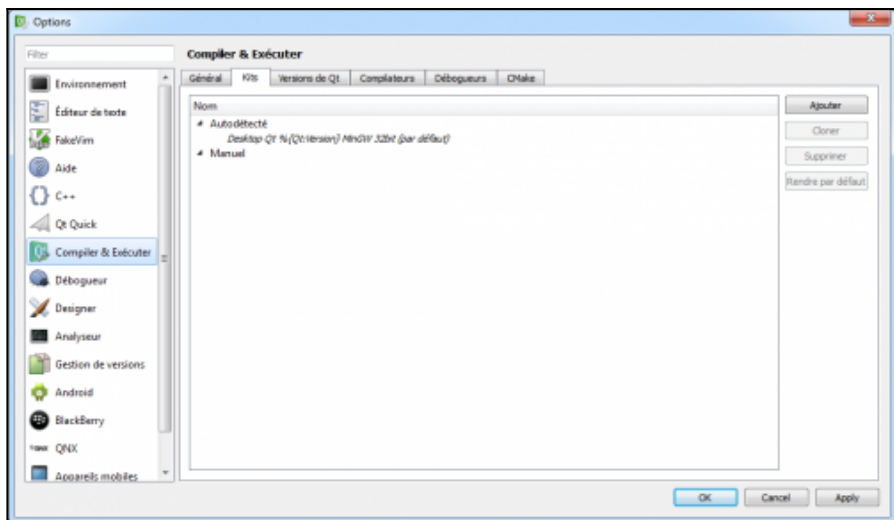
Les compilateurs se déclinent en plusieurs versions, il faudra bien choisir la version correspondant à la version de Qt utilisée. Le nom du compilateur correspondant à une version de Qt est indiqué dans la version de Qt.

Pour le **Débogueur**, si vous avez installé les compilateurs MingW, GCC ou Clang, le débogueur est inclut 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.



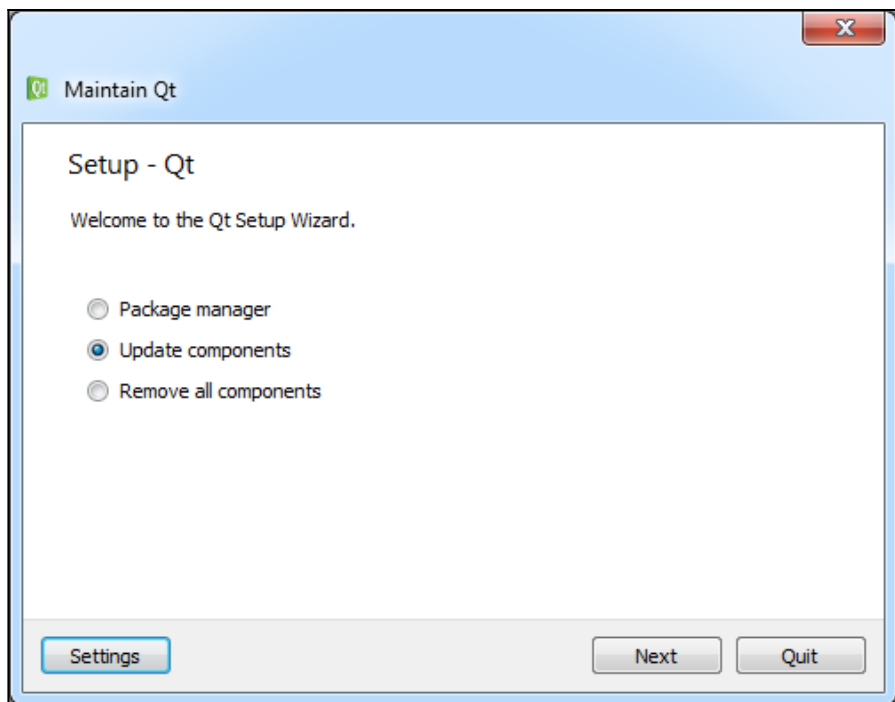
Si un kit n'est pas valide, il y aura un triangle rouge sur le kit et un message indiquera le problème. Il faut obligatoirement au moins un kit valide pour compiler. Pour qu'un kit soit valide, il faut utiliser un compilateur compatible avec une version de Qt.

L'une des erreurs les plus courantes est d'installer une version de Qt pour MSVC sans installer MSVC (l'erreur vient du cours C++ de OpenClassroom).

Remarque : si on vous demande sur un forum avec quelle version de Qt et quel compilateur vous avez compilé votre programme, il faut donner les versions du kit que vous utilisez. Certain font parfois l'erreur d'aller dans le menu "Aide" puis "À propos de Qt Creator..." et donne la version de Qt utilisée pour compiler Qt Creator, pas leur programme.

Mettre à jour Qt

La méthode la plus simple pour mettre à jour Qt est de lancer l'application `MaintenanceTool` qui se trouve dans le répertoire d'installation de Qt.



Avec cet outil, vous pouvez choisir “Mettre à jour les modules” (ou “Update component”, selon la version de Qt) pour mettre à jour automatiquement les outils (Qt Creator, Installer, etc.) Pour mettre à jour les versions de Qt et compilateur, choisissez “Gestionnaire de paquets” (ou “Manage component”). Le reste de la procédure est identique à une installation.

Savoir lire la documentation de Qt