

Compiler Qt

Sous-titre

par Jonathan Courtois Thibaut Cuvelier ([Site web](#)) ([Blog](#))

Date de publication :

Dernière mise à jour :

I - Introduction.....	3
II - Compilation de Qt.....	3
II-A - Compilation de base.....	3
II-A-1 - Windows.....	3
II-A-1-a - Configuration.....	3
II-A-1-a-i - Microsoft Visual C++.....	3
II-A-1-a-ii - GCC.....	4
II-A-1-a-iii - Commande configure.....	4
II-A-1-b - Compilation.....	4
II-A-2 - Linux.....	5
II-A-3 - Mac OS X.....	5
II-A-3-a - Compilation.....	5
II-B - Compilation avancée.....	5
II-C - Nettoyage du répertoire de compilation.....	6
II-C-1 - Windows.....	6
II-C-2 - Linux.....	6
II-C-3 - Mac OSX.....	6
II-D - Un seul répertoire de sources pour de nombreuses compilations.....	6
II-E - Monsieur, j'ai mal configuré !.....	6
III - Cross-compilation (principes de base pour compiler Qt pour une plateforme mobile).....	6
IV - Compilation de Qwt.....	6
V - Conclusion.....	6

I - Introduction

L'objectif de cet article est de présenter les différentes méthodes permettant de compiler les sources de Qt en fonction de votre système d'exploitation et du compilateur que vous souhaitez utiliser. Il décrira comment compiler Qt sur les systèmes Windows, Linux et Mac OSX avec les différents paramètres de compilation et comment nettoyer le répertoire de compilation pour obtenir des binaires fonctionnels les plus légers possible.

Cet article peut également servir de tutoriel pour les personnes souhaitant compiler les binaires de Qt afin de les mettre à disposition de tous dans la rubrique Binaires Qt.

II - Compilation de Qt

La compilation de Qt passe tout d'abord par la récupération du code source de la bibliothèque se trouvant dans le répertoire source du FTP de Nokia.

Choisissez les sources que vous souhaitez compiler, dans notre exemple, il s'agit du fichier "qt-everywhere-opensource-src-4.6.2.zip".

En effet, les fichiers qui sont proposés sur la page de téléchargement de Nokia sont :

- Le SDK de Qt qui contient l'ensemble des outils dans une seule installation ce qui inclut beaucoup de fichiers qui ne nous intéressent pas ;
- Le framework qui est déjà compilé pour une plateforme spécifique avec aucune option de configuration.

Ces deux téléchargements ne nous intéressent pas dans le cadre de la compilation des binaires.

II-A - Compilation de base

Dans un premier temps, nous allons voir comment compiler Qt pour obtenir des binaires de base (sans option de configuration) équivalents à ceux que propose Nokia pour une plateforme spécifique sur sa page de téléchargement (LGPL > Qt librairies).

II-A-1 - Windows


Décompresser les sources téléchargées sur votre ordinateur. Pour l'exemple, les sources de Qt seront dans le répertoire :

```
C:\qt\4.6.2
```

II-A-1-a - Configuration


II-A-1-a-i - Microsoft Visual C++

Si vous ne possédez pas Visual Studio, une version express (gratuite) peut être téléchargée sur le site de microsoft.

 **Remarque** : j'utiliserais 200X dans les commandes et chemins suivants qui peut être remplacé par 2005 ou 2008 selon la version de Visual Studio que vous utilisez.

Pour lancer la configuration de Qt avec Visual Studio, il est nécessaire d'utiliser l'invite de commande fourni avec l'installation dans laquelle toutes les variables d'environnement requises sont initialisées.

Elle se trouve dans le menu démarrer : Microsoft Visual Studio 200X > Visual Studio Tools > Invite de commande de Visual Studio 200X,

 **Remarque** : les chemins proposés imposent que Windows soit installé sur C: et que VS soit installé dans le répertoire par défaut.

II-A-1-a-ii - GCC

La compilation des sources de Qt avec GCC sur windows se fait à l'aide de MinGW (Minimalist GNU for Windows) et MSYS qui est un shell complet pour windows faisant parti du projet MinGW.

Si vous ne possédez pas MinGW, télécharger la dernière version sur la page du projet. Cochez la case "g++ compiler" et "mingw32-make" durant l'installation. La dernière version de MSYS est disponible sur cette page. Il est nécessaire après l'installation d'indiquer à MSYS dans quel répertoire est installer MinGW (post install process).

Executer maintenant le shell de MSYS que l'on peut généralement trouver dans le menu démarrer : MinGW > MSYS > MSYS

II-A-1-a-iii - Commande configure

La configuration de la compilation se fait à l'aide de la commande configure. Pour une compilation de base, nous ne passons aucun paramètre à la commande à l'exception de la plateforme sur laquelle nous souhaitons compiler car il se peut que celle désirer ne soit pas votre plateforme par défaut (%QMAKESPEC%).


Information : Les noms de toutes les plateformes supportées pour les sources que vous avez téléchargées peuvent être trouvés dans le répertoire mkspecs.

Visual 200X :

```
C:\Qt\4.6.2>configure -platform win32-msvc200X
```


GCC :

```
C:\Qt\4.6.2>configure -platform win32-g++
```

 **Remarque** : la configuration des sources peut prendre quelques minutes.

II-A-1-b - Compilation

Il est maintenant temps de lancer la compilation qui elle peut durer plusieurs heures en fonction de votre machine.

 **Remarque** : en général, on utilise nmake pour Visual Studio, mais make peut également être utilisé avec celui-ci.

Visual 200X :

```
C:\Qt\4.6.2>nmake
```

GCC :

```
C:\Qt\4.6.2>mingw32-make
```


Une fois la compilation terminée, que vous ayez compilé avec Visual Studio ou GCC, ajoutez "c:\Qt\4.6.2\bin" à la variable d'environnement PATH.

II-A-2 - Linux

II-A-3 - Mac OS X

Décompresser les sources téléchargées sur votre ordinateur. Pour l'exemple, les sources de Qt seront dans le répertoire :

```
/home/utilisateur/Qt/4.6.2/configure
```

 *Remarque : la configuration des sources peut prendre quelques minutes.*

II-A-3-a - Compilation

Il est maintenant temps de lancer la compilation qui elle peut durer plusieurs heures en fonction de votre machine.

```
/home/utilisateur/Qt/4.6.2/make && sudo make install
```


Il ne reste plus qu'à modifier la variable d'environnement PATH pour définir le répertoire où Qt a été compilé.

```
PATH=/home/utilisateur/Qt/4.6.2/bin:$PATH  
export PATH
```

II-B - Compilation avancée

Nous souhaitons maintenant réaliser une compilation avancée en ajoutant des options au moment de la configuration. La méthode de compilation est la même que celle présentée dans la partie précédente, à l'exception que les commandes suivantes sont ajoutées à la commande configure. Les options de configuration en gras dépendent de votre plateforme de compilation.

- **-debug-and-release** : compile et link 2 bibliothèques Qt, une avec debug activé et l'autre sans ;
- **-shared** : utilisation dynamique de la bibliothèque Qt. Peut être changé en **-static** en fonction des besoins ;
- **-fast** : création des MakeFiles uniquement pour la bibliothèque et les sous-répertoires ciblés ;
- **-nomake examples** : ne compile pas les exemples ;
- **-nomake demos** : ne compile pas les demos ;
- **-plugin-sql-x** : active le support d'une base de donnée ou x peut être mysql (MySQL), psql (PostgreSQL), oci (Oracle), odbc (ODBC), tds (TDS), db2 (DB2), sqlite (SQLite), sqlite2 (SQLite 2), ibase (InterBase) ;
- **-platform** : le système d'exploitation et le compilateur que vous utilisez pour la compilation ;
- **-qt-zlib** : utilise zlib avec Qt ;
- **-qt-y** : active le support des images d'une extension particulière ou y est gif (GIF), libpng (PNG), libmng (MNG), libtiff (TIFF) et libjpeg (JPEG) ;
- **-openssl** : support de OpenSSL ;
- **-dbus** : support de D-BUS ;
- **-phonon** : compile le module phonon ;
- **-webkit** : compile le module webkit ;
- **-script** : compile le module QtScript ;
- **-scripttools** : compile le module QtScriptTools ;
- **-native-gestures** : utilise les gestes natifs sous Windows 7 ;
- **-saveconfig myconf** : lance la configuration et sauvegarde les paramètres dans le fichier `configure_myconf.cache`.

 **Remarque :** Pour les bibliothèques tels que OpenSSL, Dbus, SQL, Oracle... elles doivent être installés sur la machine pour que cela fonctionne.

Pour la mise a disposition des binaires dans la rubrique de Binaires Qt, il est nécessaire que le path de compilation soit long (256 caractères). Sinon le patcher ne peut pas modifier les binaires comme il faut.

La commande de configuration peut alors ressembler à la suivante :

```
configure -debug-and-release -shared -fast -nomake examples -nomake demos -plugin-sql-mysql -plugin-sql-psql -plugin-sql-oci -plugin-sql-odbc -plugin-sql-tds -plugin-sql-db2 -plugin-sql-sqlite -plugin-sql-sqlite2 -plugin-sql-ibase -qt-zlib -qt-gif -qt-libpng -qt-libmng -qt-libtiff -qt-libjpeg -openssl -dbus -phonon -webkit -scripttools -native-gestures -saveconfig myconf -platform win32-msvc2008
```

II-C - Nettoyage du répertoire de compilation

II-C-1 - Windows

II-C-2 - Linux

II-C-3 - Mac OSX

II-D - Un seul répertoire de sources pour de nombreuses compilations

II-E - Monsieur, j'ai mal configuré !

III - Cross-compilation (principes de base pour compiler Qt pour une plateforme mobile)

IV - Compilation de Qwt

V - Conclusion