Installer Qt4 sous Windows

par Nicolas Joseph

Date de publication : 18 Juillet 2005

Dernière mise à jour :

Cet article a pour but d'expliquer l'installation de Qt4 sous Windows afin de developper une application graphique avec Dev-cpp.

I - Introduction

Depuis la version 4, la bibliothèque **Qt** (développée par Trolltech) est disponible gratuitement sous **Windows** (uniquement pour le développement d'applications à but non commercial). Autre restriction imposée par **Trolltech**, la compilation ne peut se faire qu'à l'aide de Mingw c'est pourquoi Dev-cpp est un IDE de choix pour le développement d'applications basées sur cette bibliothèque.

II - Téléchargement

Avant de nous lancer dans l'installation de **Qt**, il faut vous assurer d'avoir tous les packages nécessaires à cette installation :

- Dev-cpp de préférence avec le package **Mingw**
- Qt version 4 minimum

III - Installer Dev-cpp

Une fois ces packages en votre possession, commencez par installer **Dev-cpp**. Je suppose par la suite qu'il est installé dans le répertoire *C:\Dev-Cpp*.

IV - Installer Qt

Une fois l'intallation de **Dev-cpp** correctement effectuée, vous pouvez installer **Qt**. La première chose que vous demande l'installeur de **Qt** est le répertoire où se trouve **Mingw**, il faut préciser le répertoire racine de **Dev-cpp** (*C:\Dev-Cpp* dans notre exemple). Ensuite, il ne vous reste plus qu'à préciser le répertoire d'installation de **Qt** (*C:\Qt* sera utilisé par la suite).

V - Création du template

Le but de cette étape est d'obtenir une entrée pour **Qt** dans la boîte de dialogue "Nouveau projet" de **Dev-cpp**. Pour la gestion des projets, **Dev-cpp** utilise un système de templates qui est en fait un fichier texte regroupant les options spécifiques à chaque type de projet. Voici un exemple de fichier template utilisé pour les projets "Windows Application" :

```
[Template]
ver=1
Name=Windows Application
IconIndex=0
Description=A standard Windows application
Catagory=Basic

[Unit0]
CName=main.c
CppName=main.cpp
C=winapp_c.txt
Cpp=winapp_c.txt

[Project]
UnitCount=1
Type=0
Name=Windows App
```

Nous allons donc nous inspirer de cet exemple pour créer un template **Qt**. Certains attributs ne seront pas présents et d'autres vont être ajoutés.

V-A - [Template]

La partie *Template* permet de personnaliser ce qui va être affiché dans la boîte de dialogue de création d'un nouveau projet. Pour notre type de projet, voici un exemple de ce qui peut être fait :

```
[Template]
ver=1
Name=Qt
Icon=Qt.ico
Description=Create a GUI using the Qt library.
Catagory=GUI
```

Plutôt qu'un long discours, voici le résultat :

Donc, *Name* est le texte affiché sous l'icône dont le chemin de l'image est spécifié par l'attribut *Icon* (j'ai trouvé cette icône dans le répertoire *C:\Qt\examples\tools\qtdemo*). Le champ *Description* est affiché dans la partie... Description et enfin *Category* permet de choisir l'onglet sous lequel va se trouver notre projet.

V-B - [Unit]

Cette partie permet d'associer un fichier au projet, *Unit* doit être suivi d'un chiffre qui sera incrémenté pour chaque fichier. Dans notre cas, nous aurons qu'un seul fichier et par conséquent qu'une seule section nommée Unit0 :

```
[Unit0]
CppName=main.cpp
Cpp=qt_cpp.txt
```

Deux attributs sont à renseigner : le premier est le nom par défaut sous lequel sera enregistré le fichier, quant au second, c'est le nom du fichier qui servira de modèle, dont voici un exemple :

qt_cpp.txt
```
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    QPushButton hello("Hello world!");
    hello.resize(100, 30);

    hello.show();
    return app.exec();
}
```

Une fois la création du projet validé, voici le résultat :

V-C - [Project]

Pour terminer, il nous reste le plus important : les options concernant l'ensemble du projet. Voici la denière partie du fichier template :

```
[Project]
UnitCount=1
Type=1
IsCpp=1
CppCompiler=...
Linker=...
ProjectIcon=Qt.ico
```

Voici le rôle de chaque attribut :

Pour récupérer les options du compilateur et du linker, il faut commencer par ouvrir une fenêtre de commande Dos puis placez vous dans le dossier *C:\Qt\examples\tutorial\t1* :

```
C:\>cd C:\Qt\examples\tutorial\t1
```

Puis tapez les commandes suivantes :

```
C:\Qt\examples\tutorial\t1>qmake -project
C:\Qt\examples\tutorial\t1>qmake
```

Maintenant dans le répertoire *C:\Qt\examples\tutorial\t1*, vous obtenez plusieurs nouveaux fichiers dont un nommé *Makefile.Release*, ouvrez le avec votre éditeur de texte préféré. Ensuite, il suffit de copier les valeurs des attributs DEFINE, CXXFLAGS et INCPATH les uns à la suite des autres pour former les options du compilateur (CppCompiler) et ceux de la ligne LFLAGS et LIBS pour obtenir la valeur de l'attribut linker.

Pour mieux comprendre, regardez ce fichier Makefile.Release et le fichier Qt.template qui en résulte.

VI - Conclusion

Voilà, maintenant il ne vous reste plus qu'à faire un simple Fichier->Nouveau->Projet pour obtenir un projet utilisant **Qt**. Téléchargez l'archive zippée qu'il vous suffit d'extraire dans le répertoire template de **Dev-cpp**.

VII - Remerciements

Merci à Bestiol pour la relecture attentive de cet article.