Activité : Mise en oeuvre de Qt

Thierry Vaira <tvaira@free.fr>

01/02/2016

Table des matières

Site : tvaira.free.fr

Mise en oeuvre de Qt

Présentation Qt

Qt est une **bibliothèque logicielle orientée objet** développée en **C++** par Qt Development Frameworks, filiale de Digia.

Qt est une plateforme de développement d'interfaces graphiques (GUI - *Graphical User Interface*) fournie à l'origine par la société norvégienne Troll Tech, rachetée par Nokia en février 2008 puis cédée intégralement en 2012 à Digia (www.qt.io).

Qt permet la portabilité des applications (qui n'utilisent que ses composants) par simple recompilation du code source. Les environnements supportés sont les Unix (dont Linux) qui utilisent le système graphique X Window System, Windows et Mac OS X.

Qt est principalement dédiée au développement d'interfaces graphiques en fournissant des éléments prédéfinis appelés widgets (pour windows gadgets) qui peuvent être utilisés pour créer ses propres fenêtres et des boîtes de dialogue complètement

prédéfinies (ouverture / enregistrement de fichiers, progression d'opération, etc). Qt dispose d'un moteur de rendu graphique 2D performant.

Qt fournit également un ensemble de classes décrivant des éléments non graphiques : accès aux données, connexions réseaux (*socket*), gestion des fils d'exécution (*thread*), analyse XML, etc.

La dernière version est Qt 5.0 qui est sorti le 19 décembre 2012. Bien que marquant des changements majeurs sur bien des points, le passage de Qt4 à Qt5 casse au minimum la compatibilité au niveau des sources.

Version de Qt

Vous pouvez vérifier la version installée sur votre poste :

```
$ dpkg-query -l | grep qt[0-9]
...
ii  libqt4-xmlpatterns:i386 4:4.8.1-0ubuntu4.9 Qt 4 XML patterns module
ii  qt4-demos               4:4.8.1-0ubuntu4.9 Qt 4 examples and demos
ii  qt4-designer            4:4.8.1-0ubuntu4.9 graphical designer for Qt 4 applications
ii  qt4-dev-tools           4:4.8.1-0ubuntu4.9 Qt 4 development tools
ii  qt4-doc                 4:4.8.1-0ubuntu4.9 Qt 4 API documentation
ii  qt4-linguist-tools      4:4.8.1-0ubuntu4.9 Qt 4 Linguist tools
ii  qt4-qmake               4:4.8.1-0ubuntu4.9 Qt 4 qmake Makefile generator tool
...
```

Ici, on utilisera donc **Qt4** !

Qt4 (et Qt5) sépare sa bibliothèque en modules :

- QtCore : pour les fonctionnalités non graphiques utilisées par les autres modules ;
- QtGui : pour les composants graphiques, maintenant QtWidgets (qt5) ;
- QtNetwork : pour la programmation réseau ;
- etc ...

Environnement de Développement Intégré (EDI)

Qt Creator est l'environnement de développement intégré dédié à Qt et facilite la gestion d'un projet Qt. Son éditeur de texte offre les principales fonctions que sont la coloration syntaxique, le complètement, l'indentation, etc... Qt Creator intègre en son sein les outils Qt Designer et Qt Assistant. Il intègre aussi un mode débuggage.

Remarque : même si Qt Creator est présenté comme l'environnement de développement de référence pour Qt, il existe des modules Qt pour les environnements de développement Eclipse et Visual Studio. Il existe d'autres EDI dédiés à Qt et développés indépendamment de Nokia, comme QDevelop et Monkey Studio.

Structure générale des classes Qt

L'API Qt est constituée de classes aux noms préfixés par `Q` et dont chaque mot commence par une majuscule (`QString`, `QObject`, ...). Il faut savoir que la classe `QObject` est la classe mère de toutes les classes Qt.

Par exemple, la classe `QString` est une classe Qt pour **manipuler les chaînes de caractères**. Pour pouvoir utiliser une classe Qt, il ne faudra oublier d'inclure son fichier de déclaration :

```
#include <QString>

...

QString myString;
```

qmake

Qt se voulant un environnement de développement portable, il a été nécessaire de concevoir un moteur de production spécifique. C'est ainsi qu'est conçu le programme `qmake`.

Ce dernier prend en entrée un fichier (avec l'extension `.pro`) décrivant le projet (modules Qt, liste des fichiers sources, dépendances, paramètres passés au compilateur, etc...) et génère un fichier de projet spécifique à la plateforme. Ainsi, sous les systèmes UNIX/Linux, `qmake` produira un `Makefile`.

Le fichier de projet est fait pour être très facilement éditable par un développeur. Il consiste en une série d'affectations de variables.

La génération d'une application se fait en plusieurs étapes :

- création d'un répertoire et des sources (cela pourra dépendre de l'EDI utilisé) : le nom initial du répertoire détermine le nom du projet et donc de l'exécutable qui sera produit (par défaut)
- générer un fichier `.pro` qui décrit comment générer un `Makefile` pour compiler ce qui est présent dans le dossier courant : `$ qmake -project`
- générer un `Makefile` à partir des informations du fichier `.pro` : `$ qmake`
- fabriquer l'application en appelant l'outil make : `$ make`

Remarque : on exécute la commande `qmake` seulement si le fichier `.pro` est modifié

Exemple

On crée maintenant un programme de test :

```
$ mkdir hello
$ cd hello

$ vim main.cpp
```

```
#include <QDebug>

int main()
{
    qDebug("Hello wordl!");

    return 0;
}
```

On fabrique et on exécute le programme de test :

```
// Génère le fichier de projet .pro
$ qmake -project

// Génère le fichier Makefile
$ qmake

// Fabrique l'exécutable
$ make

// On teste
$ ./hello
Hello wordl!
```

Ressources

- Documentation générale de Qt
- Présentation Qt
- Cours Qt

Retour au sommaire

