Coup de projecteur sur Qt Creator, l'IDE multi-plateformes

Qt Quarterly

par Kavindra Palaraja traducteur : Thibaut Cuvelier Qt Quarterly

Date de publication : 18/03/2009

Dernière mise à jour : 31/08/2009

Qt Creator est un environnement de développement intégré, sans coût, multi-plateformes pour le développement de projets Qt. Il est disponible sur Linux, Mac OS X et Windows.

Cet article est une traduction autorisée de **Spotlight on Qt Creator: The Cross-Platform IDE**, par Kavindra Palaraja.

I - L'article original

Qt Quarterly est une revue trimestrielle électronique proposée par Qt Software à destination des développeurs et utilisateurs de Qt. Vous pouvez trouver les **versions originales**.

Cet article est la traduction de l'article **Spotlight on Qt Creator: The Cross-Platform IDE** de Kavindra Palaraja paru dans la Qt Quarterly Issue 28.

Cet article est une traduction d'un des tutoriels écrits par **Nokia Corporation and/or its subsidiary(-ies)** incluse dans la documentation de Qt, en anglais. Les éventuels problèmes résultant d'une mauvaise traduction ne sont pas imputables à Nokia ou Qt Software.

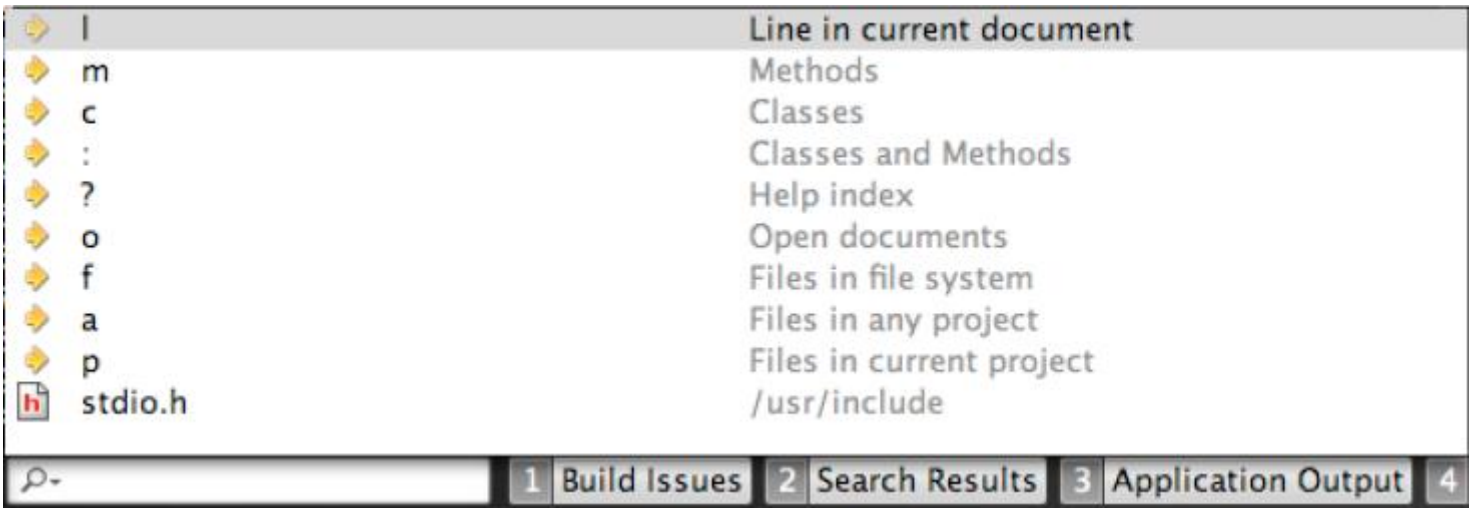
II - Sans encombrement

Développé avec les idées de vitesse et d'efficacité en tête, Qt Creator est perclus de fonctionnalités très utiles, comme un éditeur intelligent de texte, un navigateur de code puissant, un frontal pour le débugger GNU *gdb*, ainsi que l'intégration avec Qt Designer et Qt Help. Cet article met ces quelques fonctions en valeur, et indique la manière dont Qt Creator fonctionne.

Si l'on regarde de plus près à l'interface utilisateur de Qt Creator, on remarquera que la majorité de l'écran est laissée pour le travail, et se minimise le plus possible. Vous pouvez toujours enlever une partie pour faire de la place.

III - Le locator

Lors du développement d'application, il est pratique de pouvoir garder une trace de vos fichiers et de naviguer à travers le code source en même temps. Locator vous permet d'associer des fichiers à des combinaisons de touches.



Vous pouvez aussi voyager à travers votre code pour trouver une classe, une méthode, une ligne, et regarder la documentation de Qt, comme Qt Creator connaît tous les symboles dans les fichiers ouverts. Utilisez simplement **Ctrl + K** suivi d'un des préfixes pour vous lancer.

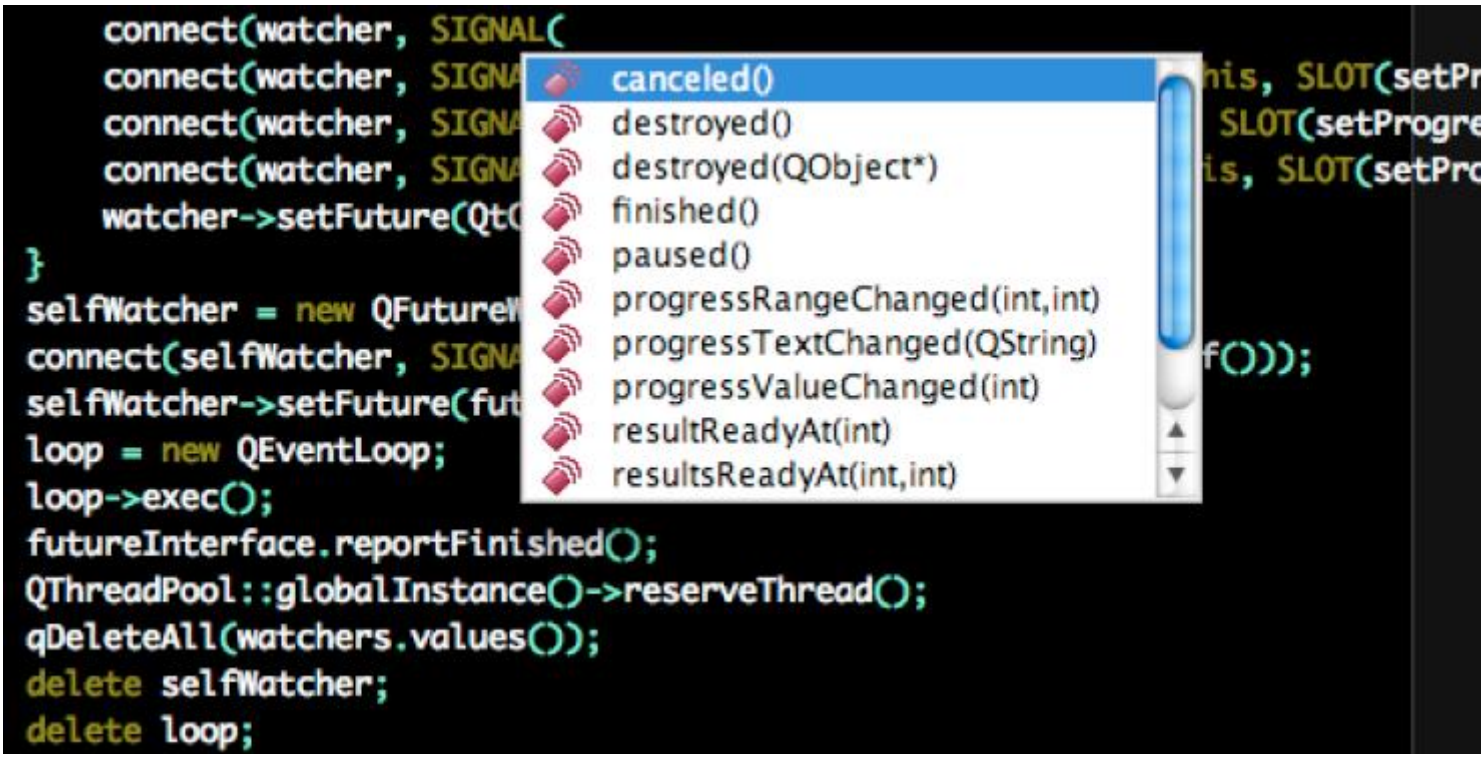
Qt Creator fournit un certain nombre de préfixes pour aider la navigation.

- l, espace, numéro de ligne : aller à la ligne du document actuel
- :, espace, nom de symbole : aller à la définition du symbole
- O, espace, nom de document : aller au document ouvert
- ?, espace, sujet de l'aide : aller vers l'aide
- F, espace, nom de fichier : ouvrir un fichier du système
- A, espace, nom de fichier : ouvrir un fichier d'un projet actuellement chargé

IV - Coloration syntaxique et auto complétition

L'éditeur intelligent inclut la coloration syntaxique pour les mots-clés, les directives du préprocesseur, et d'autres informations contextuelles.

Si vous travaillez avec de nombreux fichiers et projet concurramment, vous apprécierez l'auto complétition. Elle ne s'arrête pas aux classes et fonctions avec les types des paramètres : elle complète aussi les signaux et les slots.

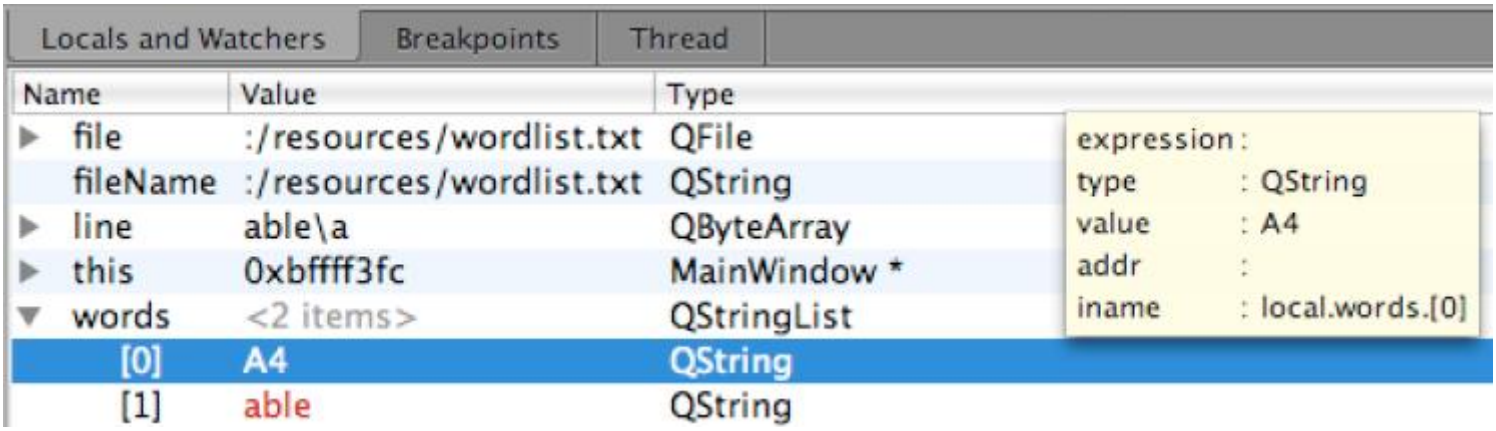


Quand vous codez, Qt Creator collecte tous les symboles de votre code, dans une base de données structurée. Ensuite, il scanne les symboles de cette base pour vous montrer les plus pertinents et syntaxiquement corrects. Tout ceci est effectué en temps réel, quand vous codez.

V - L'intégration avec gdb

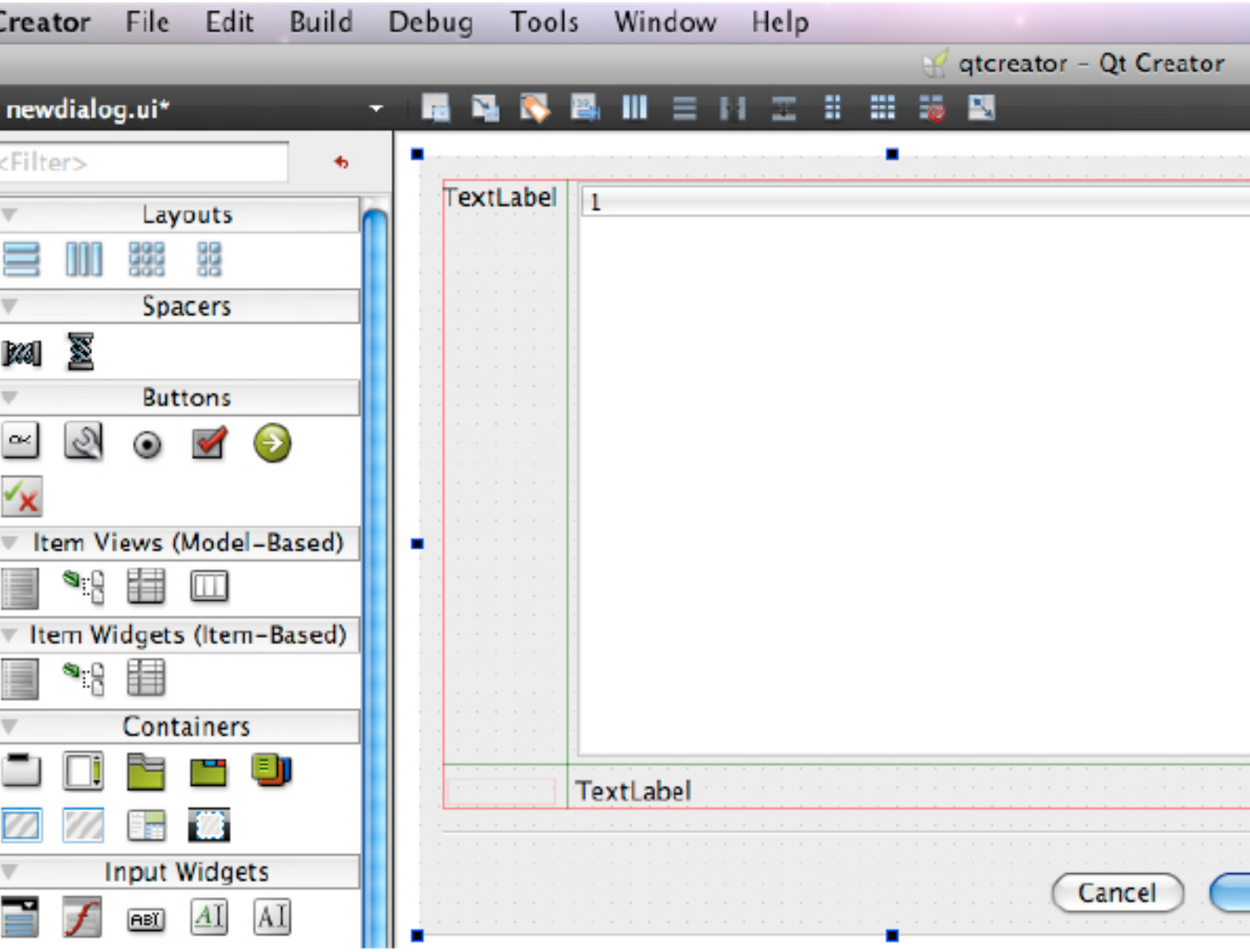
Qt Creator n'a pas son propre débuggeur. À la place, il fournit une interface graphique au débuggeur GNU, gdb. Cette interface vous permet de traverser votre code, ligne par ligne, instruction par instruction, de stopper le déroulement d'un programme, de mettre des points d'arrêt, d'examiner le contenu de la pile, les variables locales et globales ...

Vu que Qt Creator est prévu pour simplifier le processus de déboggage d'applications Qt, il fournit une fonctionnalité supplémentaire : l'affichage des vrais objets des classes Qt et des conteneurs QTL.



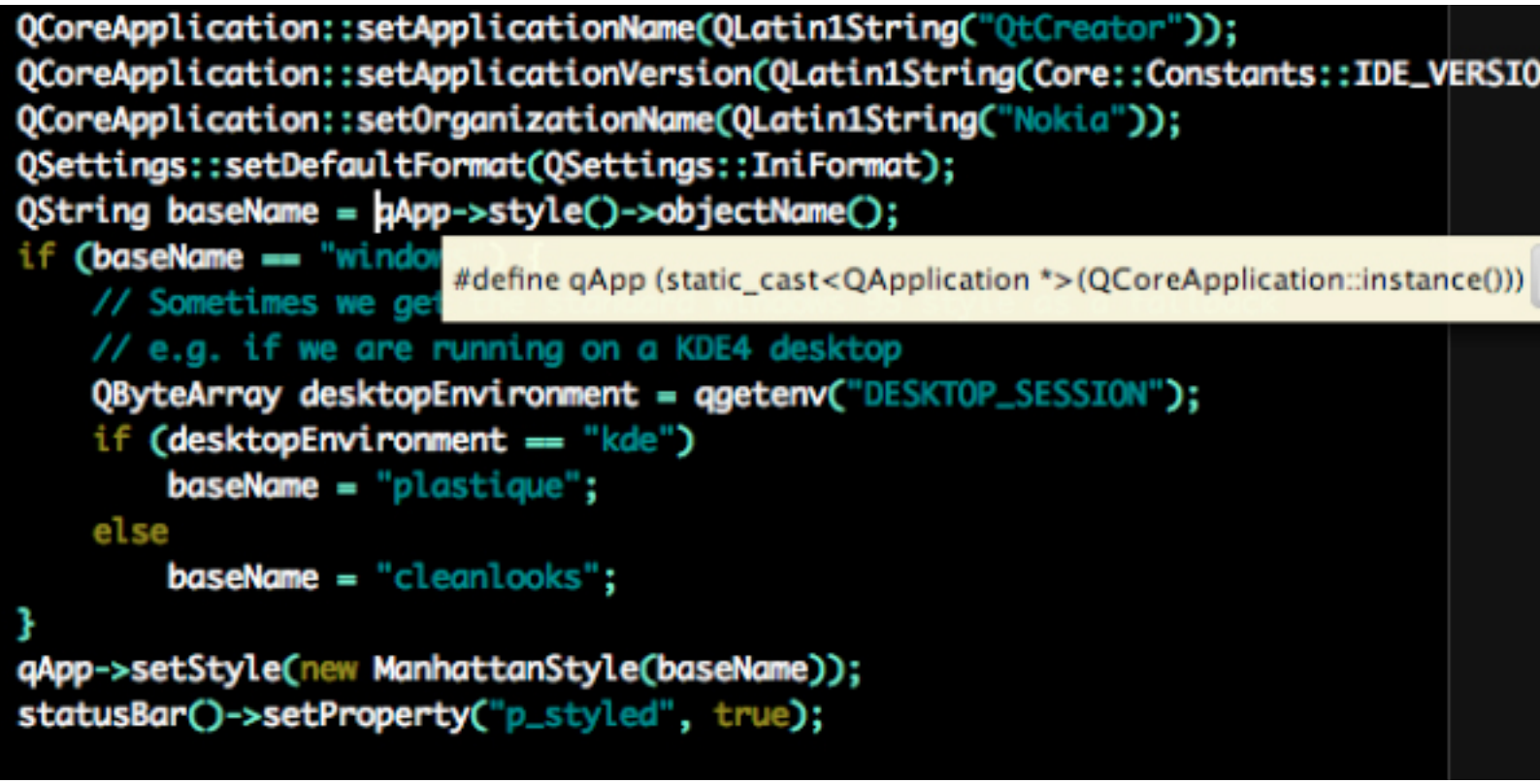
VI - Intégration avec Qt Designer

Le module d'intégration avec Qt Designer est aussi fourni. Ainsi, vous pouvez dessiner un formulaire, écrire le code nécessaire, compiler et lancer votre application en une seule et unique interface.

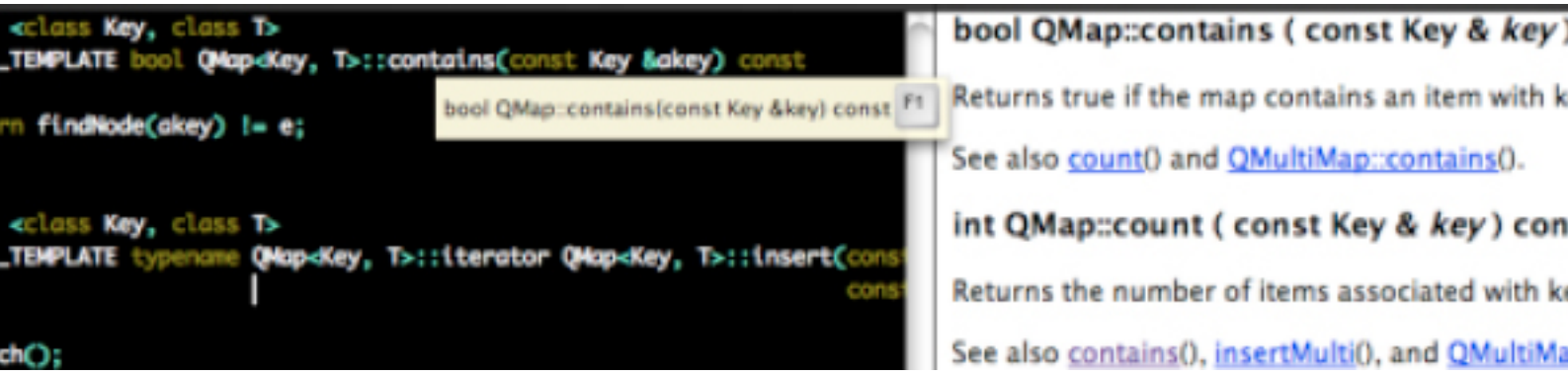


VII - Intégration avec Qt Help

Qt Creator est livré avec le plugin d'intégration avec Qt Help. Pour regarder cette aide, il vous suffit de passer en mode *Help*. Pour l'aide contextuelle, placez votre pointeur sur une classe ou une fonction Qt, et appuyez sur F1.



L'aide contextuelle est affichée dans un panneau sur la droite, pour que vous puissiez la voir en même temps que votre code.



VIII - Fichiers projet et assistant projet Qt4

Qt Creator utilise uniquement le format *.pro* de *qmake*, ce qui permet d'éviter d'apprendre un format supplémentaire, et de simplifier le processus de migration : il vous suffit de charger le fichier *.pro*.

L'opération de création de projet est simplifiée par l'assistant de génération de projet. Ceci fournit trois options : l'application en console, l'application avec une GUI et la librairie.

IX - Appuyez sur Échappe pour vous échapper

La touche Échappe a une fonction spéciale sous Qt Creator. Dans le mode *Edit*, la première fois que vous appuyez dessus, l'éditeur reçoit le focus. La seconde, toutes les fenêtres secondaires seront fermées.

X - Le futur

Il est prévu d'améliorer la robustesse face à du code C++, d'ajouter de la génération de code et du refactoring. Au final, Qt Creator aura toutes les fonctionnalités requises pour le développement pour l'embarqué et le mobile, comme l'accès à distance, la cross-compilation, le débuggage et l'émulation.

XI - Divers

Au nom de toute l'équipe Qt, j'aimerais adresser le plus grand remerciement à Qt Software et à Nokia pour nous avoir autorisé la traduction de cet article !

J'adresse ici un tout aussi grand remerciement à **ram-0000** et à **buggen25** pour leur relecture !