

Trolltech DevDays 2006 : Tests des applications Qt

par Matthieu Brucher (<http://matthieu-brucher.developpez.com/>) (Blog)

Date de publication : 01/11/2006

Dernière mise à jour :

Tester les applications Qt, résumé de la conférence d'Harald Fernengel lors des Trolltech DevDays 2006.

- I - Qui est Harald Fernengel ?
- II - Résumé de la présentation
 - II-A - Tester avec Qt
 - II-B - Débugger la mémoire
- III - Conclusion

I - Qui est Harald Fernengel ?

Harald Fernengel est le responsable des releases de Qt, donc concerné au premier point par le déploiement des applications Qt.

II - Résumé de la présentation

Depuis la version 4.2 de Qt, QTestLib fait partie des outils standards de Qt, auparavant, il s'agissait d'une Qt Solution.

II-A - Tester avec Qt

QTestLib a été conçu pour être léger, mais pour permettre des tests cross-platform et cross-compiler.

Un test est un exécutable, ou plus précisément une classe qui sera instantiée et chacun de ses slots privés sera appelé à son tour. On utilise la macro **QTEST_MAIN** pour générer une fonction main automatiquement. C'est donc un hack du mécanisme signal/slot.

Les tests peuvent générer du texte coloré ou un rendu XML, chaque message ne peut être interrompu par un autre message - ils sont donc atomiques et thread-safe -. Enfin, les macros **QVERIFY** et **QCOMPARE** permettent d'effectuer des tests, respectivement si le test réussi ou si 2 valeurs sont égales.

On peut aussi réaliser des tests sur un ensemble de données. Pour cela, on crée une fonction qui chargera les données - à l'aide de **QTest::addColumn** et **QTest::newRow** -, puis la fonction de test elle-même chargera les données à l'aide de **QFETCH**.

Les tests graphiques ou plutôt événementiels sont possibles. Ce ne sont pas des événements réels, mais des événements Qt. Il est aussi possible de générer des listes d'évènements.

Outre ces éléments, il est possible d'espionner les signaux émis à l'aide de **QSignalSpy**.

II-B - Débugger la mémoire

Pour débbugger les fuites mémoire, Trolltech conseille principalement valgrind, sous Linux, ou encore MudFlap lorsque valgrind n'est pas disponible - fourni avec GCC -. Sous Windows, on peut aussi utiliser Purify ou tout autre utilitaire. Il est fortement probable que des warnings parasites se produisent, indiquant que de la mémoire non initialisée est utilisée, cela est dû à une utilisation partielle de certains tableaux. Il est donc indispensable de faire attention à ce qu'on débbugue.

III - Conclusion

Plusieurs bibliothèques existent, basées sur Qt, mais pour des tests simples, cette bibliothèque est suffisante. Pour plus de détail, veuillez consulter [cette page](#).

