

Trolltech DevDays 2006 : Qt in depth

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

Date de publication : 01/11/2006

Dernière mise à jour :

Qt in depth, résumé de la conférence de Bradley Hughes lors des Trolltech DevDays 2006.
Téléchargez la présentation associée [ici](#).

- I - Qui est Bradley Hughes ?
- II - Résumé de la présentation
- III - Conclusion

I - Qui est Bradley Hughes ?

Bradley Hughes est le responsable du développement cross-plateforme de la base de Qt.

II - Résumé de la présentation

Pour commencer, chaque classe d'un certain poids dans Qt utilise l'idiome P_IMPL. Il s'agit de déporter dans une classe privée les variables et les fonctions dépendants d'une plateforme. Cela permet d'assurer une compatibilité binaire entre les différentes versions de la bibliothèque. Cette implémentation permet aussi d'implémenter le Copy-On-Write.

Avant de voir le COW, il faut observer que Qt dispose d'opérations dites atomiques. Ces opérations ne peuvent pas être interrompues par un autre thread. Elles permettent entre autres d'implémenter un comptage de référence ou les **QMutex**. Ces classes atomiques ne sont pas disponibles dans l'interface publique à cause de problèmes entre les architectures.

Ces opérations atomiques permettent d'implémenter le COW, ou encore partage implicite selon la terminologie Trolltech. Le COW est le fait de ne pas copier en profondeur un nouvel élément, sauf lorsqu'on veut le modifier. Cela permet de gagner du temps lors d'une copie, même s'il est coûteux de tester l'unicité et de faire une copie lorsqu'on en a besoin.

Qt4 est optimisé pour la concurrence et non pas pour la sécurité d'un thread - ce qu'on appelle thread-safety -. C'est au programmeur de poser les protections nécessaires, afin que les performances soient assurées.

Enfin, l'implémentation de l'idiome P_IMPL est récursif. QObject possède un pointeur vers l'implémentation comme tous ses descendants, mais il s'agit du même pointeur, qui peut être casté différemment selon l'objet en cours d'utilisation.

III - Conclusion

Qt in depth est plus intéressant d'un point de vue conceptuel que pour la programmation avec Qt. De nombreuses classes sont inaccessibles, mais connaître ce fonctionnement permet de mieux comprendre la bibliothèque. Pour plus de détails, veuillez consulter [cette page](#).

