

QLocale : à propos du temps (et des dates, et des langues, et des...)

par [Jeremy.Katz](#) Traducteur : [Louis du Verdier](#)

Date de publication : 28/03/2011

Dernière mise à jour :

L'un des buts de Qt est de rendre simple ou, du moins, de faciliter le portage d'un système d'exploitation à un autre. Une application Windows transférée sur un bureau Linux ? Bien sûr. Une application Mac OS X sous Symbian ? OK.

I - L'article original.....	3
II - Une application, beaucoup de localisations.....	4
III - Analysons le problème.....	5
IV - Planification d'une solution.....	6
V - Remerciements.....	7

I - L'article original

Les *Qt Labs Blogs* sont des blogs tenus par les développeurs de Qt, concernant les nouveautés ou les utilisations un peu extrêmes du framework.

Nokia, Qt et leurs logos sont des marques déposées de *Nokia Corporation* en Finlande et/ou dans les autres pays. Les autres marques déposées sont détenues par leurs propriétaires respectifs.

Cet article est la traduction du billet **QLocale: It's about time (and dates, and languages, and...)**, par Jeremy.Katz.

II - Une application, beaucoup de localisations

Le portage d'un système d'exploitation à un autre est donc facilité. Qu'en est-il de passer de la Norvège à l'Allemagne ? De l'Australie aux États-Unis d'Amérique ? Y aurait-il une raison valable pour laquelle cela ne se ferait pas automatiquement ? Bien qu'il y ait un grand nombre de détails à connaître, la classe QLocale veut en encapsuler un grand nombre.

La raison pour laquelle je suis en train de mettre cela en place est que QLocale ne suffit pas toujours. Bien que la classe ne soit pas toute neuve, quelques cas d'usage sont parvenus récemment, soulignant le problème. Ces cas de figure sont accompagnés par liens vers d'autres implémentations de données de localisation. Un exemple est la classe MeeGo Touch **MLocale**. Un autre ? **Les API de localisation** de Symbian. Il serait idéal de fournir quelques-unes de ces fonctionnalités sur toutes les plateformes de Qt.

III - Analysons le problème

Voici une courte liste des éléments manquants à QLocale, sans ordre particulier :

- de la pertinence ;
- de la collation ;
- l'écriture de scripts ;
- les scripts numériques ;
- un calendrier ;
- les journées de travail de la semaine ;
- le premier jour de la semaine ;
- un format temporel sur 24 heures ;
- la gestion des fuseaux horaires ;
- le format des numéros de téléphone ;
- les endonymes.

De plus, on nous a parlé de localisations mixtes. Pour illustrer : je suis là, en Norvège, en train d'écrire les règles d'utilisation de l'anglais US, mais, quand je cherche où est l'épicerie la plus proche, j'aime assez avoir les unités de distance en kilomètres. La base de données générée avec Qt n'a pas de *en_US* pour certaines choses, de même pour les autres locales. MeeGo Touch divise la localisation en **plusieurs catégories**. Cette division mène à la question suivante : étant donné la capacité d'aller d'une localisation (en_US, par exemple) à la gestion d'un élément de donnée en particulier (comme une date), Qt a-t-il besoin d'intégrer une capacité à effectuer l'opération inverse ? Si je sais qu'on est le « January 1, 2011 », est-ce que savoir que c'est une date formatée sous la forme d'une date en_US est important ?

Un autre problème soulevé est la valeur des localisations intégrées dans Qt. Sont-elles utiles ou bien Qt devrait compter sur la connaissance des localisations du système d'exploitation ? Est-ce que les **modificateurs et les codesets** documentés devraient être supportés d'une manière plus significative ?

IV - Planification d'une solution

Les questions avec lesquelles je vous quitte sont les suivantes : de quoi avez-vous besoin dans le support des localisations de Qt ? J'encourage vivement la validation ou le rejet des caractéristiques présentées ici. Si j'ai oublié quelque chose, n'hésitez pas à le souligner !

V - Remerciements

Merci à **dourouc05** pour son soutien et son aide durant la traduction, et à **Claude LELOUP** pour sa relecture attentive.