Un updater avec Qt

Par Thibaut Cuvelier

Date de publication : 15 juin 2009

Dernière mise à jour : 18 octobre 2009

Comment approfondir mes connaissances dans Qt ? Comment créer un updater avec Qt ? Quelques questions auxquelles cette série espère répondre.

N'hésitez pas à commenter cet article !
Commentez

I - Objectifs....3
II - Contenu....3
III - Présentation....4
IV - Prérequis....4
V - Divers....4

Copyright ® 2009 Thibaut Cuvelier. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.
https://tcuvelier.developpez.com/tutoriels/qt/updater/01-introduction/

I - Objectifs

L'objectif de cette série sera de créer un updater, un programme qui permet de mettre à jour une application depuis une source située en général sur Internet. Évidemment, il a aussi pour objectif de réduire au maximum les données échangées, pour soulager le serveur et le client, qui doit alors moins attendre.

D'abord, nous allons commencer par un updater tout simple. Une simple fenêtre avec un bouton pour lancer la mise à jour, qui s'effectuera tout aussi simplement en remplaçant les fichiers en local par la dernière version disponible sur le serveur, sans vérifier qu'elle est bien nécessaire. Les fichiers seront téléchargés par le protocole **HTTP**.

II - Contenu

Ensuite, nous améliorerons le programme : il pourra vérifier sur le serveur que la mise à jour est nécessaire en comparant le poids et les dates, ces informations seront transmises sous forme de tableau par un script sur le serveur.

Puis, le programme téléchargera un XML et le parsera, afin de vérifier la présence de mises à jour. Il vérifiera aussi un hash **MD5** des fichiers, pour vérifier que les versions locales ne sont pas corrompues.

Postérieurement, nous ajouterons une base de données en local qui contiendra ces données, et qui servira à la comparaison. L'utilisateur aura la possibilité de vérifier l'intégrité des fichiers par rapport à la dernière mise à jour.

Subséquemment, les données transmises seront compressées par LZMA, pour économiser la bande passante.

Puis, nous intégrerons cela dans un assistant, avec votre logo, et l'icône du bouclier sous Vista lorsqu'une élévation des privilèges est requise.

Ensuite, nous autoriserons la traduction de l'application, pour que tous puissent l'avoir dans leur langue, du chinois à l'anglais, en passant bien sûr par le français et le russe.

C'est alors que nous nous intéresserons à la création de mises à jour : avant, tout se passait sur le FTP. Maintenant, vous aurez une autre application qui vous permettra de créer des mises à jour et de les mettre sur le serveur, toujours par le protocole **FTP**.

Alors, nous enverrons un mail à un serveur, qui se chargera de l'envoyer à tous les inscrits à votre newsletter pour les informer de l'existence de la nouvelle mise à jour.

Mais votre application sera de plus en plus utilisée, et vos serveurs tomberont de plus en plus vite à genoux. Il vous faudra y remédier : la meilleure solution est de répartir la charge de travail sur plusieurs serveurs, et même clients. Nous utiliserons les *torrents* et *Metalinks* pour ce faire.

Il deviendra alors intéressant d'avoir des statistiques sur le téléchargement de vos mises à jour : l'updater pingera votre serveur, celui-ci stockera ces statistiques, et votre créateur de mise à jour pourra récupérer ces statistiques et en faire un graphique.

Cependant, vous n'aimeriez pas que tous puissent contrôler le contenu de ce ping ? Ou de toute autre transaction ? C'est pour cela que le **SSL** peut être utilisé avec le **protocole HTTP** : le résultat est le **HTTPS**, protocole que Qt gère aussi simplement que le **HTTP**.

Finalement, certaines de vos mises à jour sont réservées à ceux qui ont payé votre produit : vous n'aimeriez pas que n'importe quel petit malin puisse les utiliser sans payer ? Nous crypterons donc ces fichiers dès leur envoi depuis votre PC, et les décrypterons sur le PC du client.

Copyright ® 2009 Thibaut Cuvelier. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.
https://tcuvelier.developpez.com/tutoriels/qt/updater/01-introduction/

Voici le programme de cette série. Qt, même s'il permet beaucoup de facilités, n'en propose pas à tous les niveaux : par exemple, il n'est pas trivial de construire un graphique. C'est pourquoi je vous présenterai deux autres librairies, prévues pour fonctionner avec Qt : Qwt et Qxt.

La première permet de gérer les graphiques, justement. La seconde, tout ce qui est envoi de mail, en ce qui concerne notre exemple.

III - Présentation

Chacun des articles présentés sera divisé en deux parties : une partie théorique et une partie pratique.

La théorie sera faite pour pouvoir être lue indépendamment de la série, afin que chacun puisse s'en servir comme référence future, sans être perturbé par la présence d'éléments se référant à l'article précédent, obligeant sa relecture. Ceci ne signifie pas que les exemples ne seront pas de la partie : ils ne se rapporteront pas directement à notre updater.

Ensuite viendra la partie pratique : elle utilisera cette théorie, qui deviendra un prérequis pour comprendre le code. Certaines parties sont ardues, et difficilement explicables hors d'un exemple, cette partie permettra de faire le point dessus, de comprendre toutes les subtilités des techniques développées.

Le code de l'updater sera disponible à tous sous licence GPL, sur le SVN de Développez. Le code sera arrêté à chaque article et fourni dans une archive.

IV - Prérequis

Cette série ne commencera pas à un niveau nul : pour la suivre, vous serez censés avoir lu, compris et intégré **Débuter dans la création d'interfaces graphiques avec Qt4**. Vous serez ainsi familiarisé avec la conception de GUI sous Qt, avec ses principes fondateurs et avec la manière de l'utiliser.

Vous devrez évidemment disposer d'un environnement de développement qui vous convient : il s'agit d'un éditeur de texte, ainsi que d'un compilateur récent. Aussi, vous aurez installé Qt en version 4.5, Qwt en version 5.2, Qxt en version 0.5 et QCA en version 2.0. Cela signifie que vous aurez les sources et les binaires compilés, prêts à être utilisés. Vous devrez aussi avoir un plugin pour QtSql compilé : celui pour QSLite 3, livré avec Qt, sera amplement suffisant.

Qwt, Qxt et QCA se compilent comme tout autre projet Qt. Référez-vous à leur documentation pour plus d'informations.

- **Installer Qt avec MinGW**
- **Compiler Qt sur toutes les plateformes (et cross-compiler Qt)**
- **Installer Qt avec support de MySQL, de FireBird et d'OpenSSL**

- **Qt**
- **Qwt**
- **Qxt**
- **QCA**

V - Divers

Un tout grand merci à **yan**, **Alp**, **superjaja** et à **Amnell** pour leurs encouragements et conseils lors de la rédaction de cet article !

Un tout grand merci à **evarisnea** pour sa relecture orthographique !

Copyright ® 2009 Thibaut Cuvelier. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.
https://tcuvelier.developpez.com/tutoriels/qt/updater/01-introduction/