

The REBOL Documentation Project

-- FR - Documentation REBOL - Divers --

Divers

Inflation de code sur Rebol 3.0 ?

Philippe Le Goff

Première publication : 8 juin 2006, et mis en
ligne le jeudi 8 juin 2006

Résumé :

Traduction des articles 21 et 274 du Blog Rebol 3.0 au sujet de l'accroissement de taille de Rebol

Inflation de code dans Rebol 3.0

Carl Sassenrath, CTO REBOL Technologies 4-May-2006 18:39 GMT Article #0021

Au cas où vous l'auriez manqué, j'ai posté une note sur la blog à propos de l'inflation de la taille de REBOL. (NdT : La voici ci-dessous).

"REBOL Bloat Busting" *Carl Sassenrath, CTO REBOL Technologies 4-May-2006 16:44 GMT Article #0274*

J'ai eu quelques retours de personnes qui étaient inquiètes à propos du fait que REBOL 3.0 risquait de devenir plus gros. Des commentaires du genre :

J'aime le fait que REBOL/core soit seulement de 220K et qu'il ne nécessite pas d'installation ou de librairies particulières. S'il vous plaît, ne changez rien à cela.

Et, celui-ci, encore plus abrupt :

J'ai une inquiétude par-rapport au fait que REBOL 3.0 va tendre à être "comme les autres" — c'est à dire comme Python, Ruby, etc., conçu en cédant aux exigences du patron idiot de la société.

A ceci, je réponds : pas d'inquiétude !

REBOL est tout à la fois puissant mais léger, et nous allons faire en sorte de conserver cela. C'est le style de REBOL.

Jusqu'ici, malgré plusieurs nouvelles caractéristiques et nouveaux types de données, REBOL 3.0 est actuellement plus petit et plus rapide que les versions précédentes. Je ne peux pas promettre que REBOL 3.0 reste plus petit (il y a encore beaucoup à faire) mais mon opinion est ... nous allons conserver ce qui fait REBOL : la légèreté et la petite taille.

Je pense que l'une des aspects du problème est le terme "Programmer en grand" qui a été utilisé dans la Roadmap de REBOL 3.0. Ce terme a une signification particulière, qui peut induire une incompréhension.

"Programmer en grand" ne signifie pas que le système REBOL ou ses programmes doivent être grands. Vous pouvez encore écrire un programme de 10K qui fait des choses incroyables.

Actuellement, cette expression signifie que si vous écrivez des programmes de grande taille, vous aurez de meilleurs mécanismes internes pour les gérer. Par exemple, les modules (espaces de noms) vous permettent d'écrire des morceaux de code sans vous soucier que des mots globaux entrent en collision avec des bouts de votre code. Les tâches vous permettent de créer facilement des sections de code asynchrones. Les attributs des objets vous laissent contrôler la manière dont sont accédés les champs des objets et permettent d'associer de la documentation à vos objets (comme avec les fonctions).

Donc, conservons l'esprit REBOL. Grossir c'est moche. Rester petit, c'est plus élégant.

Note personnelle :

Je pense que la plupart des gros systèmes et logiciels sont non maintenables et cela de manière insensée. En conséquence, non seulement ils plantent (ex mon receveur satellite TV se plante presque quotidiennement), mais ils ne peuvent être ni adaptés, ni améliorés rapidement vis-à-vis des besoins de l'utilisateur. Je pense que le monde est condamné si cette approche (toujours plus) continue à dominer la conception des logiciels modernes.

Je ne peux attendre le jour où REBOL 3.0 nous permettra d'avoir nos petites reblets en action non seulement sur l'ordinateur (n'importe lequel), mais aussi ce même code utilisé sur la TV et le téléphone portable.

"Je veux un téléphone portable programmable en REBOL, et je le veux maintenant !".

(traduction : Philippe Le Goff)