

<http://clx.asso.fr/spip/?1-Pourquoi-creer-ses-propres>



1- Pourquoi créer ses propres paquetages

- Documentations - Technique - RPM : Faites-le vous même ! -



Date de mise en ligne : vendredi 7 janvier 2005

Copyright © Club LinuX Nord-Pas de Calais - Tous droits réservés

Avant de commencer à attaquer le vif du sujet, voici en quelques points, les raisons qui peuvent vous pousser à construire un paquetage pour votre distribution préférée.

Des raisons de créer ses RPM

Il y a plusieurs raisons pour lesquelles vous pouvez avoir envie de faire vos RPM. Si vous consultez cette page Web, c'est peut-être que vous avez déjà une raison de construire votre RPM. Et bien, vous allez en découvrir plusieurs autres.

1) Un logiciel dont vous êtes le héros (je veux dire *l'auteur*)

Eh oui, vous n'avez pas oublié qu'au siècle dernier, vous étiez fier de présenter votre application sous la forme d'un logiciel auto-installable sous Windows construit à l'aide d'[Inno Setup](#) ou d'[InstallShield Express](#). Alors maintenant que vous êtes passé au libre, présenter votre logiciel sous la forme d'un RPM, cela va de soi !

2) Un logiciel qui n'est pas disponible pour votre plate-forme

Vous possédez un Mac ? Un AMD 64 ? un Athlon ? Bref, une architecture non-Intel et vous aimeriez bien que ce logiciel trouvé sur le Net fonctionne sur votre ordinateur. Vous avez à votre disposition le paquet RPM pour architecture Intel (processeur Pentium et autre, i386, i686...) et le paquet RPM contenant les sources du programme. La solution est de démarrer du « SRPM » (ou *src.rpm*) et de compiler le logiciel pour votre plate-forme. Vous créez ainsi un *ppc.rpm* ou un *amd64.rpm* !

3) Un logiciel qui n'existe qu'en archive compressée

Ha la la, cette fois c'est pire encore ! Vous n'avez trouvé que les sources en *tar.gz* ou en *tar.bz2* contenant les sources ? Allez-vous le compiler pour votre plateforme ? Hé bien faites-en un RPM afin d'en faire profiter votre entourage dans un premier temps puis, bien plus de monde encore.

4) Un RPM qui existe mais contenant un programme qui ne fonctionne pas sur votre PC

Eh oui ! Ça arrive aussi ! Erreur dans le RPM ou bien configuration exorbitante exigée par l'auteur du paquet...

Je vous livre ici une anecdote : J'ai installé le RPM de [BillardGL](#) (jeu de billard comme son nom l'indique) sur mon ordinateur familial car il refusait de s'installer sur mon portable. Urpmi indiquait comme dépendance un noyau supportant les pilotes vidéo non libres nVidia accéléré avec son module GLX. J'ai donc installé un noyau supplémentaire adapté pour installer BillardGL.

Par erreur, j'ai lancé l'exécutable et quelle surprise !

Le lendemain, j'ai rallumé la machine en démarrant sur le noyau original (non-nVidia) de ma distribution. Via SSH, j'ai voulu configurer en français ce jeu à partir de mon PC portable (oui je suis un grand flemmard et monter un étage pour modifier un fichier de configuration était trop dur).

Par erreur, j'ai lancé l'exécutable et quelle surprise de voir qu'il est apparu sur l'écran de mon portable ! C'était « impossible », pour deux raisons :

- Le PC familial ne tournait pas avec le noyau spécialement compilé pour le support des cartes nVidia, et
- le portable dispose d'une carte graphique Intel845 de base, ne disposant pas d'accélération graphique.

Après quelques recherches, j'ai compris que l'auteur du paquet RPM a imposé l'exigence d'une carte nVidia et de son pilote "Full Patate" là où ce jeu fonctionnerait sur bien d'autres machines graphiquement moins performantes ! Après l'avoir installé de force sur mon portable, j'ai recommencé la même expérience avec un autre jeu de billard : [Foobillard](#). Même constat : il fonctionne honorablement sur mon portable qui ne remplit pourtant pas les exigences graphiques imposées par le RPM. Donc, il est possible de construire un paquet RPM qui n'exige pas le support des accélérations de la carte graphique.

De quoi avoir envie de mener ses propres expériences RPM non ?

1- Pourquoi créer ses propres paquetages

Post-scriptum :

2^Úme partie : [Les principes de la création d'un RPM](#)