

<http://clx.asso.fr/spip/?3-La-demarche>



3- La démarche

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



Date de mise en ligne : dimanche 16 janvier 2005

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

En pratique

1) Prérequis

Tout d'abord, vous devez installer quelques paquetages :

- *rpm* ;
- *rpm-build* ;
- *spec-helper* ;
- *libtool* ;
- *rpmlint*.

Le paquetage rpm est probablement déjà installé ;)

Pour les autres paquetages, ouvrez une console en se "*loguant*" en root, tapez cette commande :

```
urpmi rpm-build spec-helper libtool rpmlint
```

Il faut bien sûr accepter l'installation des dépendances que l'utilitaire *urpmi* vous réclame.

Décidez maintenant quel utilisateur (ici monpseudo) va compiler le paquet : ça ne doit jamais être root !

Passez donc sous ce login en tapant :

```
su monpseudo
```

Ensuite une deuxième commande va installer les répertoires utiles.

```
mkdir -p ~/rpm/{BUILD,RPMS/i586,RPMS/noarch,SOURCES,SRPMS,SPECS,tmp}
```

ATTENTION : Remplacez **i586** par votre architecture.

Ces répertoires sont installés dans votre répertoire personnel (/home/monpseudo selon l'exemple choisi).

2) La configuration générale

En "standard" sur Mandrake 10, vous trouverez les fichiers suivants :

```
~/.rpmrc :  
buildarchtranslate: i386: i586  
buildarchtranslate: i486: i586  
buildarchtranslate: i586: i586  
buildarchtranslate: i686: i586
```

```
~/.rpmmacros :  
%_topdir $HOME/rpm  
%_tmppath $HOME/rpm/tmp  
%_signature gpg  
%_gpg_name Mandrakelinux  
%_gpg_path ~/.gnupg  
%distribution Mandrakelinux  
%vendor Mandrakesoft
```

Ces deux fichiers se trouvent dans le répertoire `$HOME` de l'utilisateur. Comme vous le voyez, le premier `.rpmrc` donne une série d'équivalences. Les développeurs de [Mandriva](#) ont pris comme plate-forme de référence le i586.

Le second fichier contient des variables générales, à savoir :

- `topdir` contient le chemin vers le dossier rpm donc `/home/monpseudo/rpm`.

C'est à partir de cet endroit que RPM s'attend à trouver les sous-dossiers nécessaires à la fabrication d'un rpm (voir les prérequis).

- `tmppath` désigne le dossier dans lequel RPM copiera ses fichiers temporaires pendant la fabrication de l'archive.

Les variables ci-dessous ne sont utiles que pour le référencement Mandrake.

- `signature`, c'est le logiciel GPG qui servira à authentifier votre paquet.
- `gpg name`, la clé de MandrakeSoft servira à signer le paquet qui sera proposé sur leur serveur [\[1\]](#).
- `gpg path`, vos fichiers de configuration gpg se trouvent là (ici, `~/gnupg` signifie dans le dossier `$HOME` (le `~`), sous-dossier caché `gnupg`).
- `distribution` la distribution pour laquelle votre paquet est conçu.
- `vendor` c'est vous si vous ne travaillez pas pour MandrakeSoft.

3) Les fichiers SPECS

Le fichier SPEC est un fichier texte (donc votre éditeur de texte préféré fera l'affaire) qui indique à l'application RPM comment construire votre rpm. Il se découpe en "sections". Voici des extraits commentés du squelette proposé par Mandrakeclub que nous avons commentés :

```
# D'abord on définit quelques variables (name, version et release)
# c'est plus simple lorsque l'on change de nom ou version de n'avoir qu'un seul endroit à changer
%define name    skel
%define version 0.1
%define release 8mdk

# Pour les Applications X, les menus Mandriva comportent une classification
# (donc l'endroit de l'arborescence où poser le raccourci).
%define section menu mandrake/sous-menu mandrake
# la liste est sur http://qa.mandriva.com/twiki/bin/view/Main/MenuSystem_Fr
%define title Skel

# Le nom du logiciel avec la première lettre en capitale pour qu'il apparaisse comme ça dans les menus
Mandrake

# SUMMARY courte description en 60 caractères
# Pas d'accents. Inutile de répéter le nom du logiciel
%define Summary ma description
Summary: %Summary
Name: %name
Version: %version
Release: %release
License: GPL
Group: Groupe/Sousgroupe
```

```
# Group définit l'arborescence de présentation dans RPMDRAKE "choix Mandrake".
# La liste est sur http://qa.mandriva.com/twiki/bin/view/Main/MandrivaGroups_Fr
URL: http://la.homepage.de.mon.appli.com
# il est bien de rajouter également en commentaire le lien direct vers l'URL
# de téléchargement
Source0: %name-%version.tar.bz2
# La source 0 est généralement un tar.bz2 contenant un dossier du nom de
# l'appli et la version (ici : on aura donc un dossier skel-0.1 ). Dedans,
# se trouvent les fichiers sources prêts à être compilés.
# La source 1 contient souvent 3 icônes nommés skel-16.png skel-32.png
# skel-48.png (rappel skel = nom de mon appli) où le chiffre donne la taille
# de l'icône (tar -cjf %name-icons.tar.bz2 *png permet de créer cette source)
Source1: %name-icons.tar.bz2
# Vous pouvez aussi avoir des source2,3,4...
Buildroot: %_tmppath/%name-buildroot
# La notion de buildroot est très importante : c'est là que RPM créera un
# pseudo système de fichier racine à l'intérieur duquel il pourra mettre un
# pseudo /usr/bin, un pseudo /etc bref faire sembler de travailler sur / sans
# risquer de mettre en danger le vrai /
Buildrequires: foo-devel bar-devel Requires: foo
# Utilisez ldd pour connaître les dépendances nécessaires au
# logiciel (Requires) ou à sa compilation (buildrequires).
# NE METTEZ PAS des dépendances implicites. Exemple:
# Buildrequires: gtk+-devel XFree86-devel <- Mauvais !
# Puisque gtk+-devel a déjà besoin de XFree86-devel, vous n'avez pas besoin
# d'ajouter cette seconde dépendance pour votre logiciel.
# Si RPM trouve les librairies (lisez la dernière partie de la sortie console)
# lui-même, vous n'avez rien à ajouter non plus.
%description
# Ici, vous allez mettre une description plus longue
# La section "prep" (pour "preparation") commence ici :
%prep
%setup -q
%setup -q -T -D -a1 # unpack icons
##patch1 -p0
# setup est une macro de décompression et installation des sources
# c'est là que vous patcherez éventuellement vos sources au besoin
%build
%configure
%make
%install
%__rm -rf %buildroot
%makeinstall
# Menu
# Voilà la macro qui crée les entrée dans les menus Mandrake
# CAT crée dans un fichier tout ce qu'il y a entre EOF et EOF
# install le pose où ça va bien
%__install -d %buildroot/%_menudir
%__cat << EOF > %buildroot/%_menudir/%name
package(%name): \
command="%_bindir/%name" \
needs="X11" \
```

```
icon="%name.png" \
section="%section" \
title="%title" \
longtitle="%Summary"
EOF
# icon
# ici ce sont les icônes qui sont installés là où menudrake saura aller les
# chercher en fonction de la taille écran.
%__install -D -m 644 %{name}-48.png %buildroot/%_liconsdir/%name.png
%__install -D -m 644 %{name}-32.png %buildroot/%_iconsdir/%name.png
%__install -D -m 644 %{name}-16.png %buildroot/%_miconsdir/%name.png
%pre
%post
%{update_menus}
# On demande à votre environnement graphique de mettre à jour son menu pour
_ # notre nouveau logiciel y apparaisse tout de suite (sans attendre un reboot)
%postun
%{clean_menus}
# même chose en cas de désinstallation
%clean
%__rm -rf %buildroot
# la section ci-dessous consiste à installer les fichiers à partir de buildroot
# vers le véritable système de fichier donc à leur place définitive
%files -f %name.lang
%defattr(0755,root,root,0755)
# Les fichiers désignés ci-dessous seront exécutables
# tous ceux qui sont dans "bindir") donc on leur met le chmod +X (exécutable)
%_bindir/*
%defattr(0644,root,root,0755)
# à partir d'ici ce sont les non-exécutables donc pas de chmod +X
%doc COPYING LICENSE README INSTALL Changelog AUTHORS
%_menudir/*
%_miconsdir/*
%_iconsdir/*
%_liconsdir/*
%_datadir/%name/*
%_mandir/man1/%name.1*
# Ci-dessous, on décrit ce qui change d'une version à l'autre, cela apparaîtra
# dans rpmdrake
%changelog
Sat Nov 1 2003 Han Boetes <han@linux-mandrake.com> 0.1-8mdk
- more macros and other little improvements.
```

4) La compilation

Selon les cas, la compilation se fera de plusieurs façons.

Si vous démarrez d'un fichier src.rpm (ou "srpm"),il vous faudra taper simplement :

```
rpm --rebuild lenomdemonfichier.src.rpm
```

3- La démarche

Et si tout va bien, vous obtiendrez au final une archive `lenomdemonfichier.i586.rpm` (ou une autre architecture, ça dépend de votre besoin exprimé dans vos fichiers de configuration)

Si vous partez d'un `tar.bz2`, il vous faut taper :

```
rpm -ba lenomdemonfichier.spec
```

â€” C'est tout ?

â€” Oui, c'est tout !

â€” T'es sûr ?

â€” Bon si vous insistez, voilà d'autres possibilités...

- `rpm -ba` fait toute la compilation (a signifiant "all", en français "tout") mais elle peut être décomposée en étapes :
- `rpm -bp` ne fait par exemple que la préparation avant compilation
- `rpm -bc` va jusque la compilation mais n'installe rien dans buildroot
- `rpm -bi` va jusqu'à l'installation dans buildroot mais ne crée pas les rpm
- `rpm -bb` construit juste le `i586.rpm` mais pas le `src.rpm` (les étapes précédentes sont donc faites)
- `rpm -bs` construit le `src.rpm` mais pas le `i586.rpm`
- `rpm -bl` vérifie la présence des fichiers dont il aura besoin pour l'installation (il faut donc "construire" avant).

Dans tous les cas, on indique en paramètre le fichier spec et l'endroit où il se trouve. Parfois celui-ci est directement dans le dossier "taré" des sources (le `sources.tar.bz2`), on peut alors soit extraire ce fichier SPEC pour pouvoir lancer notre compilation par `rpm -ba` soit remplacer le B de `rpm -ba` par un T (`rpm -ta`) qui demande à rpm d'aller chercher le SPEC dans le fichier tar des sources.

Post-scriptum :

4^{ème} partie : [Quelques exemples](#)

Retour à la 2^{ème} partie : [Les principes de création d'un RPM](#)

[1] La mise en oeuvre se fera par Mandrakesoft si les développeurs responsables des archives décident d'intégrer votre paquet à la distribution. Cf <http://qa.mandriva.com/twiki/bin/view/...>