Paquet PET Puppy linux

PET (Puppy's Extra Treats) est le système de paquet unifiée adoptée en Puppy version 2.14 et versions ultérieures.

Si vous avez compilé une application à partir des sources, ou téléchargé une archive binaire précompilé, et l'avoir testé dans puppy et maintenant vous voulez mettre à la disposition de tous , lisez la suite ...

Vous voulez savoir ce qui est intérieur d'un package en PET? Vous voulez savoir comment installer un package PET? Lire la suite ...

Tout d'abord, lisez la page **de gestion de paquets Puppy** pour une introduction à la gestion des paquets.

Comment faire pour installer un paquet PET

Rappelez-vous toujours qu'il y a deux types de PETS, les «officiels» et les «non officiel».

Les paquets pet officiel , ainsi que ceux de la suite Unleashed, peuvent être installés en démarrant le gestionnaire de paquets PETget (voir l'icône marquée "install" sur le bureau, ou alors dans le menu "Configuration"). Vous verrez alors une liste de tous les paquets officiels et vous choisissez ce que vous voulez.

D'autre part, les paquets en PET, officiels ou non, peut être installé juste en cliquant sur eux.
The Rox-Filer (gestionnaire de fichiers) execute tout nom de fichier qui se termine par '. PET et si vous cliquez dessus puis ROX-Filer lancera PETget. Tout package PET, officiels ou non, peut être installé de cette façon.
De même que, SeaMonkey, (le navigateur Web), Cliquez sur n'importe quel ". Pet" fichier sur une page Web et SeaMonkey vous proposera de télécharger pour immédiatement l'ouvrir dans PETget. Celui-ci signifie que vous n'avez qu'à cliquer sur un paquet en PET sur une page web et il est installé!

En fait, toute application qui lit les informations de type MIME du fichier / etc / mailcap et / etc / mime.type reconnaît les Fichiers Pets, et les lancent avec PETget. Idem pour le mime-type base de données dans / usr / share / mime.

Comment faire pour créer un package PET

Il y a une suite d'outils, scripts "PET", pour créer des paquets PET et pour la conversion de Paquet d'autre version de linux (.deb, .rpm) etc...

Tout d'abord, pour répondre à la question la plus importante: vous télécharger un paquet source à partir d'Internet exemple (paquet source en « .tar.gz ») , et vous voulez le compiler puis créez un package PET. Comment?

C'est très simple: à la place de faire votre "make install" comme d'habitude faites :

```
#New2dir make install
```

"new2dir" est un script qui va exécuter la commande "make install" et fera, en outre créer un répertoire avec tous les fichiers installés (et les répertoires). Je n'ai pas besoin de donner des détails de plus ici, le script explique exactement ce qu'il fait et comment il le fait.

Par exemple, si vous aviez compilé "abiword-2.5.6" new2dir va créer un répertoire "abiword-2.5.6-i486» avec tous les fichiers installés.

L'étape suivante est également simple:

```
# dir2pet abiword-2.5.6-i486
```

Cela va créer un paquet PET, dossier «abiword-2.5.6-i486.pet". Ça y est, tout est terminé.

Après avoir créé un paquet pour le PET, vous pouvez le rendre disponible pour d'autres. .

Outils pour fichier PET

Puppy a un ensemble de scripts pour travailler sur les paquets PET:

Script	Exemple	Description
dir2pet	dir2pet abiword-2.5.6-i486	Convertir un répertoire à un paquet PET
new2dir	new2dir make install	Créer un répertoire des fichiers installés
pet2tgz	pet2tgz abiword-2.5.6-i486.pet	Convertir un paquet pet en Archive tar.gz
pup2pet	pup2pet abiword-2.5.6.pup	Convertir un paquet PUP en paquet PET
tgz2pet	tgz2pet abiword-2.5.6.tar.gz	Convertir un tar.gz. en paquet PET

Ils sont tous situés dans /usr/bin. Ils sont tous faciles à utiliser. Il faut simplement se poser des questions simples et vous informer ce qui se passe à chaque étape.

Un paquet PET est en fait une archive (. Tar.gz) , sauf qu'il a un fichier md5sum(« « « fichier de contôle d'intégrité » » ») annexé à la fin du fichier.

Pet2tgz ne fait que supprimer ce md5sum . Tgz2pet ne fait qu' ajouter le md5sum.

Le script "pup2pet» est beaucoup plus sophistiqué. Il permet de convertir n'importe paquet PUP (ancienne version de puppy) en un paquet PET. Il va vous poser quelques questions et il est préférable que la conversion se fait par la même personne qui a créer le DotPup, mais les questions ne sont pas difficiles et n'importe qui peut facilement deviner les bonnes réponses.

Le script "new2dir" est très intelligent. Il exécutera tout ce qui est spécifié sur la ligne de commande, généralement "make install" et il faudra surveiller ce que les fichiers puissent s'installer et donc les copier dans un répertoire séparé.

Par exemple, le paquet source "abiword-2.5.6": un répertoire séparé appelé "abiword-2.5.6-i486» sera créé et les fichiers copiées. L'option "-i486» est le type de processeur pour laquelle le paquet est compilé - autres possibilités sont "-i586" et "-i686».

"new2dir" peuvent faire davantage. Il peut scinder «développement», «Documentation» et «international» en composants du paquet installé dans des répertoires distincts. Il y a un répertoire cible principale, par exemple "abiword-2.5.6-i486», mais éventuellement tout les composants peuvent être séparer en répertoire "abiword_DEV-2.5.6-i486», «abiword_DOC-2.5.6-i486" ou "abiword_NLS - 2.5.6-i486 ». Le choix appartient au créateur du paquet PET - l'idée est de réduire la taille du paquet PET principal. Normalement, si un paquet a bibliothèques partagées il est préférable d'effectuer cette division en composant, pour le "développement" au moins.
Le script "dir2pet" peut être appliqué à chacun de ces répertoires.

Structure d'un paquet PET

Dans la page **de gestion de paquets Puppy** Je donnais l'exemple du paquet torsmo.

Pour rappel, le paquet torsmo vient de ces deux fichiers:

```
/ usr / local / bin / torsmo
/ usr/local/lib/X11/mini-icons/torsmo.xpm
```

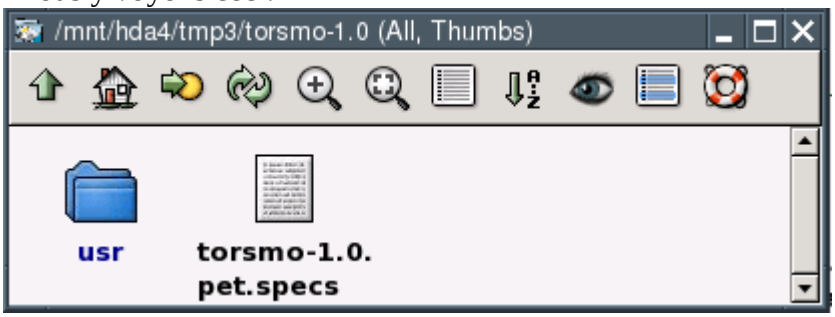
Prenons le paquet torsmo-1.0.pet et convertissons le, pour pouvoir le décomprimer

```
# pet2tgz Torsmo-1.0.pet
```

Puis décomprimons le avec:

```
# Tar-zxf torsmo-1.0.tar.gz
```

Nous y voyons ceci:



Usr "" répertoire contient les sous-répertoires et les deux fichiers.

La plupart des paquets PET ont un fichier "specs" qui est un fichier texte contenant certaines informations qui est utilisé par PETget à des fins de gestion de paquets. Les scripts "dir2pet" et "pup2pet" créer ce fichier.

Notez que vous pouvez également mettre une image de 16x16 XPM dans le paquet, petget le placera dans /usr/local/lib/X11/mini-icons (l'emplacement standard pour le menu d'icônes 16x16).

Vous pouvez également placer un XPM 48x48 ou PNG il sera placé dans / usr/local/lib/X11/pixmaps (limitation: il doit être nommé 'something48.xpm' ou 'something48.png')
Dans l'exemple torsmo.xpm sera placé dans le "usr/local/lib/X11/mini-icons"

L'image ci-dessus montre deux fichiers de plus en option: "pinstall.sh" qui s'exécute immédiatement après les fichiers des packages sont installés, et "puninstall.sh" qui s'exécute immédiatement après les fichiers sont désinstallés. (Notez que PETget va copier le script de désinstallation dans / root / .packages / et renommer en "torsmo-1.0.remove")

Juste pour compléter la procédure, après avoir ouvert le paquet PET torsmo ci-dessus, entrait dans l'intérieur, et que vous avez fait vos modification par exemple, on peut recréer notre paquet pet:

```
# tar -c -f torsmo-1.0.tar torsmo-1.0/
# gzip torsmo-1.0.tar
# tgz2pet torsmo-1.0.tar.gz
```

Le script post-installation

Il s'agit d'un script optionnel que vous auriez à créer vous-même. Très peu d'emballages PET besoin de cela. Idem pour le post-script de désinstallation.

Mais, si vous n'avez pas besoin de créer un pinstall.sh "script", voici quelques remarques:

La post-installation script a besoin de plus amples précisions. Il n'y a aucune différence entre un paquet de PET autonome et un paquet de PET dans la suite Unleashed, à l'exception de là où ils sont utilisés. les Paquets Unleashed sont utilisés pour créer un puppy live-CD fichier ISO, tandis que les paquets PET autonome sont disponibles sur Internet pour téléchargement individuel et l'installation par le gestionnaire de paquets PETget.
Dans les deux cas, la post-installation script sera exécuté après que le paquet a été installé.

Voici le format de base, tirée du navigateur Dillo:

```
#!/bin/sh
#post-install script.
#creatuppy: current directory is rootfs-complete, which has the final
filesystem.
#pupget: current directory is /.
#dillo is by default the default internal html viewer.
#if no other browser, then dillo will also have to be the default web
browser...
if [ "`ls -1 ./usr/local/bin/ | grep --extended-regexp
"opera|mozstart|links|hv3"`" = "" ];then
 echo "Configuring Dillo as the default web browser..."
  echo '#!/bin/sh' > ./usr/local/bin/defaultbrowser
  echo 'exec dillo "$@"' >> ./usr/local/bin/defaultbrowser
 echo -n "dillo" > /tmp/rightbrwsr.txt
fi
```

Il est important de savoir quel est le répertoire "actuel" lors de l'exécution du script. Les répertoires d'exécution des script ne sont pas identique en une version officiel ou non, c'est pourquoi le script a un point sur le devant "/ usr / local / bin", de manière à ce qu'il fonctionne dans les deux cas.

Ce «point devant" est la seule particularité que vous avez besoin de se rappeler lorsque vous créez un pinstall.sh

Le format de script est identique pour puninstall.sh

Les fichiers .desktop pour les menus du bureau

Un paquet PET qui doit avoir une entrée de menu du bureau, fichier doit avoir un fichier '. Desktop'. L'emplacement habituel pour ce fichier est /usr/share/applications/.

Prenez ce paquet torsmo titre d'exemple. Si il exige une entrée de menu (c'est une entrée dans le «menu» en bas à gauche de l'écran) il doit posséder :

```
/ usr / local / bin / torsmo (executable)
/ usr/local/lib/X11/mini-icons/torsmo.xpm (icône)
/ usr / share / applications / torsmo.desktop (’’’raccourci’’’)
```

Quand vous utiliser pup2pet il créer automatiquement le fichier ‘.desktop‘

Depuis Puppy v2.14 et +, le menu est créé dynamiquement à chaque fois un paquet PET est installé ou supprimé, ou chaque fois qu'une nouvelle version de Puppy est démarré (ou une version simulée de mise à niveau en utilisant "pfix = clean" paramètre de démarrage). Il ya un petit programme qui lit tous les fichier '. Desktop' et génère le fichier de configuration du gestionnaire de fenêtres (qui dispose de l'information menu). Raul (rarsa sur le Forum Puppy) a créé le moteur pour les fichiers ‘. Desktop’ dans puppy et a également écrit ces programmes générateur (un pour chacun des gestionnaires de fenêtre principale).

C'est cette génération dynamique qui le rend si grand. Installer un paquet, redémarrez le gestionnaire de fenêtres, et il ya la nouvelle entrée du menu, dans un endroit approprié dans le menu aussi bien! Désinstaller le paquet, redémarrez le gestionnaire de fenêtres, l'entrée de menu a disparu.

Vous pouvez ouvrir un fichier '. Desktop' de fichier et voyez par vous-même, mais au fond il s'agit d'un fichier texte qui spécifie tout le nécessaire pour créer une entrée de menu, comme le texte et l'icône, la localisation dans la hiérarchie des menus et le fichier exécutable à lancer.

Ce système informatique ». Desktop 'fait partie de la spécification menu XDG. De plus amples informations se trouvent ici: http://www.freedesktop.org/Standards/desktop-entry-spec et: http://www.freedesktop.org/wiki/Standards_2fmenu_2dspec

Menu fichier de configuration

Le gestionnaire de fenêtre par défaut dans la "norme" de sortie de Puppy est le gestionnaire de fenêtres JWM. Celui si a un fichier de configuration / root / .jwmrc , il contient toutes les spécifications pour la mise en page des menus et des entrées. Comme mentionné ci-dessus, ce fichier est généré automatiquement chaque fois que le menu doit être mis à jour. Alors, que faire si vous souhaitez modifier ce fichier config?

Autres gestionnaires de fenêtres sont l'équivalent des fichiers menu de configuration, par exemple IceWM a / root / .icewm / menu et fvwm95 a / root/.fvwm95. Ils mai, contiennent plus que les spécifications pour le menu, de sorte que vous mai voulez les modifier.
Quelques gestionnaires de fenêtres ne supportent pas un menu popup, et dans ce cas un programme de menu séparé peut être utilisé, par exemple "fbpanel" - il a aussi un menu équivalent du fichier de configuration.

Quand un paquet PET est installé ou supprimé, PETget appelle un petit programme appelé "fixmenus" (situé dans / usr / sbin). Celui-ci lit le gestionnaire de fenêtre (ou fbpanel) modèle de configuration des fichiers de / etc / xdg / templates et génère les fichiers de configuration réelle.
Par exemple, le fichier modèle pour JWM est / etc / xdg / templates / _root_.jwmrc et qui génère à partir de / root / .jwmrc. Conséquent, si vous souhaitez modifier manuellement le fichier de config JWM, éditez le fichier modèle.

Le fichier modèle est nommée dans une certaine manière qui montre sa destination. Les caractères "_" seront converties en caractères «/». Ainsi, le _root_.jwmrc "" fichier de modèle va générer "/ root / .jwmrc".

Le fichier modèle devrait être auto-explicatif lorsque vous les regardez. Si vous êtes familier avec l'extension ". Jwmrc" fichier puis le fichier de modèle sera aussi familier, et vous devriez être en mesure de voir clairement les parties que vous pouvez modifier.
Pour les développeurs, si vous souhaitez créer un package de gestionnaire de fenêtres en PET, par exemple fluxbox, il faut créer un fichier de modèle dans / etc / xdg / templates.
