



Debian Testing : comment l'utiliser efficacement ?

Version 1.0, par [Shankarius](http://ubunteros.tuxfamily.org/spip.php?article177) - <http://ubunteros.tuxfamily.org/spip.php?article177>

La branche « Testing » de Debian représente la future version « Stable » en développement. C'est pour beaucoup un excellent choix : les logiciels y sont récents et elle est généralement très stable. Cependant, du fait qu'elle soit en développement, elle nécessite une certaine attention et certaines connaissances pour être appréciée et utilisée sereinement.

Sommaire :

- Être prévenu sur l'état des paquets qui seront installés/mis à jour
- Le `sources.list` et le fichier de préférence des priorités
 - Le `sources.list`
 - Le fichier de préférence des priorités (`/etc/apt/preferences`)
 - Ajouter un dépôt extérieur à Debian
 - Attribuer une priorité particulière à un paquet
- L'utilisation des outils Apt dans le cadre d'un système utilisant les préférences
 - Les informations
 - Les mises à jour
 - Installer un paquet provenant d'une version supérieure à Testing (soit Unstable/Sid ou Experimental)
 - Revenir à une version inférieure d'un paquet
 - La simulation
- Astuces
 - Faire une mise à niveau du système vers le bas
 - `Apt-get` et les paquets recommandés

Être prévenu sur l'état des paquets qui seront installés/mis à jour

Avec Debian Testing les mises à jours sont quotidiennes, et même si les paquets sont restés quelques jours/semaines dans la branche « Unstable » (ou Sid) pour être testés avant d'être reversés dans Testing ils peuvent quand même parfois être bogués.

L'installation d'`Apt-listbugs` est donc recommandée. Il permet d'être prévenu sur l'état des paquets qui seront installés/mis à jour, en indiquant la nature, l'état, la gravité du bogue ainsi que les architectures concernés, le tout en anglais...

Quelques explications sur l'état des bogues :

- ▶ Done : le bogue a été corrigé.
- ▶ Fixed : le bogue a été corrigé pour une prochaine version (Pour Testing c'est en général la version qui va être installé).
- ▶ Pending : le bogue est en train d'être corrigé mais la version qui va être installé contient peut être encore le bogue. Il est donc sage de faire une vérification sur le site <http://www.debian.org/Bugs/> armé du numéro du bogue.

Tout cela en gardant à l'esprit qu'un bogue ne touche pas forcément toutes les configurations matériels, inutile donc de tomber dans la paranoïa...

Le sources.list et le fichier de préférence des priorités

Il nous faut tout d'abord répondre à une question : quelle est la différence entre utiliser la branche « Testing » et son nom de code (actuellement Lenny) dans le sources.list ?

C'est simple, si on utilise « testing » dans le sources.list on restera toujours dans la branche « Testing » de Debian. Si par contre on utilise « lenny », lorsque Lenny deviendra la version « Stable » de Debian on passera donc de la version Testing à la version Stable.

Le branche « Testing » étant particulièrement instable les semaines suivants la sortie de la version Stable il est recommandé d'utiliser dans le sources.list le nom de code de la branche « Testing » (actuellement Lenny) et de faire le cas échéant une mise à niveau vers la nouvelle version Testing une fois l'ouragan passé (un délai d'environ 2 mois est généralement suffisant).

Le sources.list

Debian Testing est une version en développement, de ce fait elle n'est pas toujours complète et des paquets peuvent, pour diverses raisons, ne pas être présent ou être retirés temporairement des dépôts Testing (par exemple les pilotes propriétaires Nvidia). De plus, même si Debian Testing est généralement très stable, il peut parfois arriver qu'un ou plusieurs paquets mettent à mal cette stabilité. Deux solutions s'offrent donc à vous :

1. Attendre que le/les paquets incriminés soit mis à jours et/ou reversés dans Testing (l'inconvénient est que de part la politique de Debian qui veut que les paquets passent un certain temps dans la branche Unstable avant d'être reversés dans Testing il faille attendre plusieurs jours/semaines/mois).
2. Installer les paquets directement à partir des dépôts de Sid ou de la version Stable, ce qui permet la plupart du temps de s'affranchir du délai d'attente et de conserver un système fonctionnel.

Il est donc quasi indispensable d'avoir les dépôts de la version Stable et de la version Unstable (Sid) dans le sources.list.

Cerise sur le gâteau, avoir accès aux dépôts de Sid et même d'Experimental permet de pouvoir tester et utiliser facilement certains paquets plus récents à sa convenance, ce qui fait partie des avantages que peut offrir Debian Testing

par rapport à d'autres distribution figées.

Un sources.list complet se présentera donc comme ceci :

```
## Stable (Etch)
deb http://ftp.fr.debian.org/debian/ stable main non-free contrib
# deb-src http://ftp.fr.debian.org/debian/ stable main non-free contrib

## Testing (Lenny)
deb http://ftp.fr.debian.org/debian/ lenny main non-free contrib
# deb-src http://ftp.fr.debian.org/debian/ testing main non-free
contrib

## Unstable (Sid)
deb http://ftp.fr.debian.org/debian/ sid main contrib non-free
# deb-src http://ftp.fr.debian.org/debian/ sid main contrib non-free

## Security
deb http://security.debian.org/ lenny/updates main contrib non-free
# deb-src http://security.debian.org/ lenny/updates main contrib non-
free
deb http://security.debian.org/ stable/updates main contrib non-free
# deb-src http://security.debian.org/ stable/updates main contrib non-
free

## Multimedia
deb http://www.debian-multimedia.org/ stable main
deb http://www.debian-multimedia.org/ lenny main
deb http://www.debian-multimedia.org/ unstable main

## Experimental
deb http://ftp.fr.debian.org/debian/ experimental main
```

ATTENTION

Ce sources.list s'accompagne obligatoirement d'un fichier de préférence des priorités (voir le chapitre suivant).

ATTENTION

Si vous faites une mise à jour de la liste des paquets (apt-get update) après avoir gonflé votre sources.list vous aurez sûrement droit à un message d'erreur de ce type : « *Dynamic MMap ran out of room* ». En effet la taille du cache d'Apt est par défaut trop petite, la solution est tout simplement de l'augmenter :

```
# echo 'APT::Cache-Limit 40000000;' >>
/etc/apt/apt.conf.d/00configperso
```

Autrement dit on ajoute l'option « *APT::Cache-Limit* » en lui fixant une valeur de 40000000 bits (39Mo) au fichier

`/etc/apt/apt.conf.d/00configperso` (en le créant s'il n'existe pas).

Le fichier de préférence des priorités (`/etc/apt/preferences`)

Bien entendu en ajoutant les dépôts d'Unstable/Sid au `sources.list` on se retrouvera en Sid à la première mise à jour du système ce qui n'est pas le but. C'est là qu'intervient le fichier `/etc/apt/preferences`. Celui-ci permet d'attribuer un ordre de priorité aux différents dépôts renseignés dans le `sources.list` ainsi qu'à des paquets spécifiques comme nous le verrons plus loin.

Le fichier `/etc/apt/preferences` correspondant au `sources.list` précédent se présentera ainsi (à créer si besoin) :

```
Package: *
Pin: release o=Debian,a=testing,l=Debian-Security
Pin-Priority: 900

Package: *
Pin: release o=Unofficial Multimedia Packages,a=testing,l=Unofficial
Multimedia Packages
Pin-Priority: 900

Package: *
Pin: release o=Debian,a=testing,l=Debian
Pin-Priority: 900

Package: *
Pin: release o=Debian,a=stable,l=Debian-Security
Pin-Priority: 800

Package: *
Pin: release o=Unofficial Multimedia Packages,a=stable,l=Unofficial
Multimedia Packages
Pin-Priority: 800

Package: *
Pin: release v=4.0*,o=Debian,a=stable,l=Debian
Pin-Priority: 800

Package: *
Pin: release o=Unofficial Multimedia Packages,a=unstable,l=Unofficial
Multimedia Packages
Pin-Priority: 90

Package: *
Pin: release o=Debian,a=unstable,l=Debian
Pin-Priority: 90
```

Ce fichier permet d'avoir, dans l'ordre, une priorité pour les dépôts Testing/Lenny > Stable/Etch > Unstable/Sid > Experimental.

Vous noterez que l'entrée correspondant au dépôt Experimental n'y est pas

renseignée. Ce dépôt ayant par défaut une priorité fixée à 1 c'est inutile mais rien n'empêche de le faire.

ATTENTION

Pour vérifier que tout soit dans l'ordre il faut bien penser après chaque manipulation du sources.list et/ou du fichier preferences à utiliser la commande :

```
$ apt-cache policy
```

Le résultat ne doit montrer aucun dépôt avec une priorité de 500 à l'exception des dépôts notés Translation-fr.

ATTENTION

L'installation de paquets provenant des dépôts Unstable/Sid ou Experimental demande une attention particulière quand aux dépendances qui seront installées. Le paquet libc6 et l'élément central du système et détermine en quelque sorte la version Debian que l'on utilise. En d'autres termes si l'on installe la libc6 provenant des dépôts Unstable/Sid ou Experimental on ne sera plus en Testing !

Ajouter un dépôt extérieur à Debian

Il est courant d'utiliser un ou plusieurs dépôts extérieurs aux dépôts Debian. Pour les dépôts proposant uniquement des paquets qui ne sont pas disponibles dans les dépôts Debian il suffit des les rajouter au sources.list, ils auront généralement une priorité fixée par défaut à 500 (vérifiable avec la commande *apt-cache policy*). Pour les dépôts proposant des paquets qui sont déjà disponibles dans les dépôts Debian il est indispensable de leur fixer une priorité.

Dans le doute il est préférable de fixer une priorité à tout les dépôts, c'est ce que nous allons voir avec l'un des dépôts d'[Xcfa](#).

On commence par ajouter le dépôt au sources.list :

```
## Xcfa  
deb http://download.tuxfamily.org/xcfaudio/dev/ lenny contrib  
# deb-src http://download.tuxfamily.org/xcfaudio/dev/ lenny contrib
```

Puis on recharge la listes des paquets :

```
# apt-get update
```

Pour récupérer les identifiants du dépôt on lance cette commande :

```
$ apt-cache policy
```

On repère la ligne correspondant au dépôt :

```
500 http://download.tuxfamily.org lenny/contrib Packages
release o=Depot Tuxfamily pour Debian Ubuntu,a=testing,l=Depot
Tuxfamily pour Debian Ubuntu,c=contrib origin
download.tuxfamily.org
```

Nous avons donc maintenant toutes les informations pour pouvoir attribuer une priorité au dépôt dans le fichier /etc/apt/preferences :

```
Package: *
Pin: release o=Depot Tuxfamily pour Debian Ubuntu,a=testing,l=Depot
Tuxfamily pour Debian Ubuntu
Pin-Priority: 900
```

Et on fini par vérifier que la nouvelle priorité ait bien été pris en compte :

```
$ apt-cache policy

900 http://download.tuxfamily.org lenny/contrib Packages
release o=Depot Tuxfamily pour Debian Ubuntu,a=testing,l=Depot
Tuxfamily pour Debian Ubuntu,c=contrib origin
download.tuxfamily.org
```

Pour les dépôts proposant des paquets qui sont déjà disponibles dans les dépôts Debian la priorité dépendra de vos désirs et se situera entre ces deux extrêmes :

- ▶ Si vous voulez que les paquets provenant de ces dépôts soient toujours prioritaires quelque soit leur version vous pouvez leur attribuer une priorité de 901 voire supérieure ou égale à 990.
- ▶ Si vous voulez que les paquets provenant de ces dépôts ne soient uniquement installables et mis à jours volontairement vous devez leur attribuer une priorité comprise entre 1 et 89 (donc inférieure aux dépôts d'Unstable).

Pour plus d'informations sur les priorités :

```
$ man apt_preferences
```

Attribuer une priorité particulière à un paquet

À l'instar des dépôts, le fichier /etc/apt/preferences permet aussi d'attribuer une priorité à un paquet. Cela peut être intéressant par exemple pour ne pas voir un paquet que l'on a compilé avec amour être remplacé à la suite d'une mise à jour, suivre un paquet provenant d'une version ou d'un dépôt particulier, etc.

Voici donc comment procéder pour attribuer une priorité particulière à un paquet. Dans notre exemple nous donnerons la priorité au paquet Iceweasel

provenant des dépôts d'Unstable/Sid.

Dans le fichier `/etc/apt/preferences` nous rajoutons ceci :

```
Package: iceweasel
Pin: release o=Debian,a=unstable,l=Debian
Pin-Priority: 901
```

ATTENTION

Il faut garder à l'esprit que seul Iceweasel est concerné par cette entrée, les paquets dont il dépend seront eux pris dans les dépôts Testing ce qui peut réserver quelques mauvaises surprises. Par exemple, les utilisateurs de Gnome qui ont installés le paquet « *iceweasel-gnome-support* » verront ce paquet être désinstallé à chaque mise à jour d'Icweasel car le paquet *iceweasel-gnome-support* à une dépendance strict sur la version d'Icweasel. Une possibilité serait de rajouter une entrée similaire pour « *iceweasel-gnome-support* » mais lors d'un changement de version majeur d'Icweasel (passage de la version 2 à la version 3) le problème risque de se représenter avec d'autres dépendances : il faut donc être très prudent avec ce genre de manipulation.

D'autres détails peuvent être ajouté pour suivre une version particulière d'un paquet (Pin : version x.x.x), etc.

Pour plus d'informations sur les priorités :

```
$ man apt_preferences
```

L'utilisation des outils Apt dans le cadre d'un système utilisant les préférences

Les informations

Nous avons déjà vu la commande « *apt-cache policy* » qui nous permet d'avoir une vue sur les priorités assignées aux dépôts renseignés dans le `sources.list`. Cette commande peut aussi être utilisé avec le nom d'un paquet en argument ce qui nous renseignera sur toutes les versions disponibles pour ce paquet, leurs provenances et leurs priorités :

```
$ apt-cache policy un_paquet
```

Les mises à jour

Debian Testing étant une version en développement certains paquets mis à jour requièrent de nouvelles dépendances. Il est donc préférable d'utiliser :

```
# apt-get dist-upgrade
```

plutôt que « *apt-get upgrade* » ou

```
# aptitude safe-upgrade
```

plutôt que « *aptitude upgrade* ».

Ces commandes font une mise à jour « intelligente » du système en installant si besoin les nouvelles dépendances requises.

Elles ne feront une mise à niveau du système vers une version supérieure (ex : de Testing à Sid) que si l'on définit au préalable la nouvelle version à suivre dans le fichier de préférence des priorités, ou, si l'on utilise pas ce fichier, que si l'on remplace dans le `sources.list` les dépôts de la version utilisée par ceux de la version vers laquelle on veut faire la mise à niveau.

Installer un paquet provenant d'une version supérieure à Testing (soit Unstable/Sid ou Experimental)

Il y a plusieurs possibilités :

```
# apt-get install iceweasel/unstable
```

ou

```
# aptitude install iceweasel/unstable
```

Installer le paquet `iceweasel` disponible dans les dépôts `Unstable/Sid`. Ces dépendances devront elles provenir des dépôts `Testing` ou `Stable` (Subtilité concernant surtout `Apt-get`).

```
# apt-get install -t unstable iceweasel
```

ou

```
# aptitude install -t unstable iceweasel
```

Installer le paquet `iceweasel` disponible dans les dépôts `Unstable/Sid` en donnant à ces dépôts une priorité temporaire de 990. Les dépendances de ce paquet seront, elles, installés en fonction des priorités définies dans le fichier `/etc/apt/preferences` (soit `Testing > Stable > Unstable > Experimental`). Il est donc préférable d'utiliser cette commande.

Aptitude dispose d'un système proposant plusieurs solutions pour régler les problèmes de dépendances. Un score est attribué à chacune de ces solutions, la solution ayant le plus grand score étant supposée la meilleure. Étrangement la première solution qu'il propose n'est pas toujours la solution ayant le plus grand score ! Il faut donc faire très attention avec Aptitude dans le choix de la solution qui sera utilisée.

Généralement la solution proposée par Apt-get correspond à la solution ayant le meilleur score pour Aptitude, et quand Apt-get refuse d'installer le paquet demandé les solutions proposées par Aptitude ont un score négatif : elles sont donc à fuir...

ATTENTION

L'installation de paquets provenant des dépôts Unstable/Sid ou Experimental demande une attention particulière quand aux dépendances qui seront installées. Le paquet libc6 et l'élément central du système et détermine en quelque sorte la version Debian que l'on utilise. En d'autres termes si l'on installe la libc6 provenant des dépôts Unstable/Sid ou Experimental on ne sera plus en Testing !

Revenir à une version inférieure d'un paquet

Vous avez fait une bêtise ? Vous avez installé un paquet Sid bogué et voulez revenir à la version Testing ? Outre la désinstallation du paquet et sa simple réinstallation vous pouvez aussi réinstaller le paquet avec Apt-get ou Aptitude :

```
# apt-get install iceweasel/testing
```

ou

```
# aptitude install iceweasel/testing
```

Même si vous préférez Apt-get il est très avantageux dans ce cas d'utiliser Aptitude qui avec son système de solutions multiples pour la gestion des dépendances s'en sort beaucoup mieux.

Si la version du paquet que vous voulez réinstaller n'est plus présente dans les dépôts deux solutions s'offrent à vous :

1. Regarder si le paquet se trouve dans le dossier `/var/cache/apt/archives/`
2. Aller sur le site <http://snapshot.debian.net/> pour récupérer le paquet.

Il faudra ensuite installer ce paquet avec la commande « `dpkg` » :

```
# dpkg -i le_paquet
```

La simulation

Elle se caractérise par l'option « -s » et permet de simuler une action demandée à Apt-get ou Aptitude. C'est très pratique pour tester sans risque une mise à jour, l'installation d'un paquet, etc.

Il est sage de l'utiliser avant toute manipulation sur les paquets.

Exemples :

```
# apt-get -s dist-upgrade
# apt-get -s install -t unstable iceweasel
# aptitude -s safe-upgrade
```

Astuces

Faire une mise à niveau du système vers le bas

Vous avez trop joué les explorateurs avec les paquets d'Unstable/Sid ou Experimental et voulez revenir dans la branche Testing ? Avant de réinstaller essayer de faire une mise à niveau du système vers le bas !

Pour cela il vous suffit de mettre temporairement les priorités des dépôts Testing à 1001 et celles des dépôts Unstable/Sid et Experimental à -1 dans le fichier /etc/apt/preferences puis :

```
# apt-get update
# apt-get dist-upgrade
```

ou

```
# aptitude update
# aptitude full-upgrade
```

(Pensez à faire une simulation avant ;))

Apt-get et les paquets recommandés

Depuis quelques temps Apt-get, tout comme Aptitude, installe par défaut les paquets marqués comme recommandés. Si vous voulez passer outre il vous faudra utiliser l'option « --no-install-recommends » :

```
# apt-get --no-install-recommends install un_paquet
```

Si vous voulez que ce comportement soit celui par défaut pour ne pas avoir à utiliser cette option à chaque fois il faut rajouter au fichier `/etc/apt/apt.conf.d/00configperso` l'option :

```
APT::Install-Recommends "false";
```

Après cela, si vous voulez installer un paquet plus les paquets marqués comme recommandés il vous faudra utiliser l'option « `--install-recommends` » :

```
# apt-get --install-recommends install un_paquet
```



debian

Ce document est placé sous la licence libre [GNU FDL](#).

Copyright (c) 2008 Shankarius.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".