

Fix broken package dependencies on Debian GNU / Linux, Ubuntu, Mint with apt commands how to

Author : admin



Deb based distributions are famous for their good binary package management system and great ability to fulfill package dependencies elegantly with the apt tool.

I personally have a long time in choosing what Linux distribution best fits me and the reason I preferred Debian over others was the simplicity in applying updates and the great number of provided prepackaged software and the easiness to manage.

However even though **apt**, **apt-get** and **aptitude** package management wrapper scripts work great most of the time - note that these 3 tools are just intelligent wrappers over the basis **dpkg** (the main install / remove / update etc. tool behind the surface), sometimes due to user stupidity or 3rd package installations with **dpkg -i** or a combined packages installed from various non-debian repositories or just user errors you might end up with **perfectly working installed packages** that depend on other packages installed from source or provided by already installed package under a different name than the installed package has a predefined dependencies, and in that case you end up with the so called **Broken package Dependencies** .

The problems with upgrading due to packages broken dependencies sometimes can be fixed with running below bunch of apt commands or - how to fix Debian / Ubuntu broken dependencies and missing dependencies preventing you from Updating / Upgrading the system:

Update the system installed package list

```
apt-get update
```

That would delete old packages from /var/lib/cache/ so sometimes if apt-get install --reinstall packagename used to be failing due to the inconsistency of previous packagename.deb which is fetched from .deb package cache, on next apt-get install --reinstall a new copy of package will be downloaded either from internet or recopied from CD / DVD repository

```
apt-get autoclean
```

to clean up any partial files

```
apt-get clean
```

to clean up apt cache

```
apt-get autoremove
```

Sometimes if all else fails and you're forced to remove the broken package that is preventing you from normal package management operations to remove it in a forceful way you can run

```
dpkg --remove -force --force-remove-reinstreq package name
```

that command is saying apt to try its best to solve the broken / missing package dependencies problems sometimes it just removes the problematic package as a solution or tries to refetch the missing package from the internet but often when the dependency system is messed up due to 5 / 10 missing or broken packages this won't help and you can't afford to just remove MySQL server that has missing dependencies or LiGHTTPD with reporting missing packages in package management system

```
apt-get update --fix-missing
```

This might help you to completely the scripts being executed while a certain package was being installed with dpkg tool, for example sometimes it happens you start installing a package with apt-get

install whatever-package-name and some extra package is being installed or to be remove which you don't want and you interrupt the process with CTRL+C and then the package that was being installed at the moment of interruption becomes in a bad broken state because of incompleted scripts on dpkg -i whatever-package-name to solve that below often helps

```
dpkg --configure -a
```

In some circumstances below command that tries to install a package being interrupted with fix-missing helps solve dependency issues

```
apt-get install -f
```

To get a list of all packages that are little known to dpkg and not perfectly installed or have been removed but configuration left unpurged:

```
dpkg -l | grep -v '^ii'
```

```
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                               Version                               Architecture Description
+++-=====
=====
rc gufw                                17.04.1-1.1                          all          graphical user
interface for ufw
```

rc laptop-mode-tools	1.71-2	all	Tools for Power
Savings based on battery/AC status			
rc mgetty	1.1.36-3+b1	amd64	Smart Modem
getty replacement			
rc ntopng	2.4+dfsg1-3	amd64	High-Speed Web-
based Traffic Analysis and Flow Collection Tool			
rc opencubicplayer	1:0.1.21-1.1	amd64	UNIX port of
Open Cubic Player			
rc snort	2.9.7.0-5	amd64	flexible Network
Intrusion Detection System			
rc speechd-up	0.5~20110719-6+b1	amd64	Interface
between Speech Dispatcher and SpeakUp			
rc thinkfan	0.9.3-2	amd64	simple and
lightweight fan control program			
rc ufw	0.35-4	all	program for managing a
Netfilter firewall			
rc wicd-gtk	1.7.4+tb2-4	all	wired and wireless
network manager - GTK+ client			

To get all packages, that are half installed and half-configured run:

```
dpkg-query -f '${status} ${package}\n' -W | awk '$3 != "installed" {print $4}'  
dpkg-query -f '${status} ${package}\n' -W | awk '$3 == "half-configured" {print $4}'
```

In some very badly messed packages that are in unrecoverably state, another useful trick is to completely wipe out information from the dpkg file with package installed statuses

```
/var/lib/dpkg/status
```

Open the file with favourite text editor (vim, nano etc.)

```
vim /var/lib/dpkg/status
```

Find the package name inside that is problematic by either scrolling or using search and delete all block records from file regarding the package

/var/cache/apt/lists/lock

Next delete

```
rm /var/cache/apt/lists/lock
```

A note to make here is in previous Debian 6 / 7, and Ubuntu version 12 and lower the location of lock file is in

```
/var/cache/apt/archives/lock
```

Talking about broken packages and how to fix them perhaps it is worthy to mention about **one great feature of apt package manager (putting the package on hold)**.

Marking a package with a hold flag is useful feature especially if you have installed on a server or Desktop with Linux a package that you want to be prevented from updating during the usual:

apt-get update && apt-get upgrade && apt-get dist-upgrade

Sometimes if you want a specific version of software to stay at the current installed version even though of if rest of software gets updated, it is useful to put this specific package on hold with dpkg, apt or aptitude.

That would prevent the set installed package program to update in future if you issue update - note that this includes preventing the programs set on hold from future security updates do issue:

Setting a package to hold / unhold with dpkg, apt, aptitude to prevent package from being updated on apt operations

Set Package Hold with dpkg:

```
echo "
hold" | sudo dpkg --set-selections
```

Set package Hold with apt

```
apt-mark hold
```

Set package Hold with aptitude

```
aptitude hold
```

Keep in mind though that If you however do a Linux version major version upgrades, putting some packages on hold could cause you problems so if you do that make sure you **unhold** the packages .

Unhold a previously holded package with dpkg / apt / aptitude:

Set Package UnHold with dpkg:

```
echo "  
unhold" | sudo dpkg --set-selections
```

Set package UnHold with apt

```
apt-mark unhold
```


Set package UnHold with aptitude

```
aptitude unhold
```