# Debian Linux: dump and migrate identical packages with (dpkg) from server 1 to server 2 /A common sysadmin dpkg package dump mistake

**Author :** admin

Over the last years it happened multiple times to me to migrate identical *Debian installation (with identical services)* hosts, running *identical Debian version* and *identical installed packages and configs* in order to move Old (hardware) servers to newer (harware) hosts. I will call for simplicity first system from which migrating "copy from host" and second "copy to host". Moving exact number of installed packages between "copy host" and "copy to host" systems can probably be done in many ways but I personally prefer using a single method - using *dpkg* to dump all deb packages list on the system in a file; move this file to "copy to host" and there use a tiny *for loop bash (cycle)* + *dpkg* to install all listed packages. Last time I've done this is just 2 days ago while I was "Resurrecting" Pc-Freak machine using my *l337 h4x0r zk!1lZ* and same good old well tested logic :)

I used following to dump all packages;

```
# dpkg -l | awk '{ print $2 }' >> /root/packages_list.txt
```

This though dumps all deb packages, along with all current installed ones dumps also, package names of debs, which used to some point in time be existent on the system - removed and the belonging package configs were kept on the system (in other words a tiny part of the package left installed on the system, just in case if one needs to install and use package some time in lets say short future).

This keeping of package name configs and skele files in Debian is called in "*dpkg* language" (rc - Remove Candidate). While doing operations *dpkg* package manager marks different packages with different flags, so *rc* flags are set once the package is *apt-get remove-d* or *dpkg -r packagename* is done over a pack.

For unfamiliar with Debian's dpkg, package system flags, check out *man dpkg*. Just to give example of *rc*, here are few packages marked as RC (Remove Candidates):

```
# dpkg -l |grep -i ^rc|head -n 3
rc acidrip 0.14-0.3 ripping and encoding DVD tool using mplayer and mencoder
rc airsnort 0.2.7e-2 WLAN sniffer
rc airstrike 0.99+1.0pre6a-4 2d dogfight game in the tradition of 'Biplanes' and 'BIP'
```

The reason, why this package are still "remembered" by *dpkg* is they were not *purged* after install- i.e. (**dpkg --purge whatever-packagename**) was not issued over 'em.

With this said in mind, it is common mistake I make while making a dump of all packages to also dump inside list names of packages mared as *RC*, e.g.:

```
# dpkg -l | awk '{ print $2 }' >> /root/packages_list.txt
```

Later I install often install every packages inside */root/packages_list.txt as for exmp., pointed out in my previous article* Debian Linux Squeeze 32 bit i386 to amd64 hell *just to later find out I have numerous (daemons), on the old "copy from host" but are installed and ran by dpkg (config scripts) on the 2nd "copy to host ....*

Thus to prevent this I recommend people, always think well before doing something (something I often miss).

Thus it is much better to dump only packages obtaining, the **ii** (dpkg flags).
Here is example of few packages which have *ii dpkg package flags*:

```
# dpkg -l | grep -i '^ii' | tail -n 3
ii zip 3.0-3 Archiver for .zip files
ii zlib1g 1:1.2.3.4.dfsg-3 compression library - runtime
ii zlib1g-dev 1:1.2.3.4.dfsg-3 compression library - development
```

Probably other people just like me, did same mistake as me to dump all ever available package names on the system and later ended up in same situation, where have to remove packages and stop services from running on system boot ...

Thus the "correct" way to dump only installed and configured ones debs having the *II* system flags is by:

```
# dpkg -l | grep -i '^ii' | awk '{ print $2 }' >> /root/only_installed_deb_packages_list.txt
```

Then the rest of package copy from *"copy from host" machine 1* to *"copy to host" 2-nd machine* is to be done by uploading */root/only_installed_deb_packages_list.txt* to 2nd host with *ftp, sftp, scp* whatever transfer proto and running on *copy to host*:

```
for i in $(cat /root/only_installed_deb_packages_list.txt); do
```

apt-get install --reinstall $i; done
.

Generally this will make programs on *copy host*, be on *copy to host*.
Enjoy :)