Create dummy packages on Debian, Ubuntu, Mint Linux how to

Author : admin

Creating dummy packages on Debian, Ubuntu, Mint Linux in order to fulfill broken or missing package dependencies is a very useful thing to do as many third party vendor packages such as **skype** does p*rovide software that is compiled against a certain set of libraries to a specific Linux distribution release and as the distribution versions evolve it is impossible to install because of missing dependencies.*

Sadly the third party maintainer almost always did not compile / provide new .deb package of their software specific for the new or sometimes older Linux distribution release and did not consider that Linux versions change frequently every 6 months or a year.

Hence The precompiled proprietary version of program and deb package provided by them depends on libraries or tools (contained within packages) that are either obsolete or does not match the package name and versions of installed Linux / Ubuntu / Mint or other deb based distribution.

When this occurs once you try to install the third party software, you can't because of the missing or unavailable packages within the current installed Linux version.

Commonly the problem with the missing and unavailable packages are not due to the inability of software to run with the older or newer (or in other words different) version of a library or tool from the ones provided by apt repositories but just because the package was compiled by third party vendor to depend on a very specfic versions of libraries and tools from the one provided by the respective Linux repositories.instead of making it cross platform compatible.

Thus often the ugly work around to that is to simply install the package in question without regard for dependencies with **dpkg**, i.e run:

```
dpkg -i --force-all skype-install.deb
```

...

Though the package gets installed, that way the broken dependencies of it are still pending and the apt package (on top of dpkg) is aware of that and once you try to do to lets say patch server or laptop against newest security flaws the do a package update you will end in a broken package state with a bunch of dependency errors of the missing libraries / packages.

You might end up in such situation if you're using lets say unstable verson of Debian or just if you're mixing /etc/apt/sources.list from Stable version with unstable one

Here is few example of ***what you might be getting once you're trying to install a package and there is / are broken unmet (missing) (uninstallable dependencies) caused by a forcably installed 3rd party software:***

```
# apt-get install libmagickwand-dev
libmagickwand-dev : Depends: libmagickwand5 (= 8:6.7.7.10-6ubuntu3.4) but it is not going to be installed
Depends: libmagickcore5-extra (= 8:6.7.7.10-6ubuntu3.4) but it is not going to be installed

Depends: libmagickcore-dev (= 8:6.7.7.10-6ubuntu3.4) but it is not going to be installed E: Unable to correct problems, you have held broken packages.
```

Above example indicates **libmagickcore5-extra and libmagickcre-dev** required version can't be installed because the provided packages from Ubuntu **/etc/apt/sources.list** defined package download location for (current installed Ubuntu Linux) is different from the one the libmagickwand-dev required, this is because I have extra package sources defined to **/etc/apt/sources.list**

So what is the solution workaround to a missing and unavailable packages due to missing dependency package requirements ?

The easiest and least "painful" way is to fool the package system by installing a dummy package with the name of the missing package requirement, so the next time, some installed package on your system depends on a missing :i386 / amd64 or i586 architecture package or some other weird older or newer package missing on the system you can emulate the package is present by:

Creating Dummy Package with equivs (*equivs-build, equivs-control*)

Lets say you want to install **GXMame frontend package manually with dpkg which is now obsolete and no longer available for install across Linux distributions the package depends on *xmame but*** xmame is not available in latest Linux deb distributions, so the only work around is to create the **xmame package as a dummy package and install it so you fool deb package management system to thing it is there and not explode with errors on apt operations.**

To create a dummy package use the *equivs which is a tool used by Debian / Ubuntu developers to do the package packaging, here is how to do it:*

```
root@jericho:/home/hipo# apt-get install --yes equivs
root@jericho:/home/hipo# equivs-control xmame
```

```
root@jericho:/home/hipo# cat xmame
```

```
Section: games
Priority: optional

Standards-Version: 2.5.5
```

```
Package: xmame
Version: 0.182-1
Section: games
Maintainer: Georgi Georgiev
Provides: mame
Architecture: all

Description: Dummy Xmame Arcade Emulator package
```

Edit xmame or whatever dummy package you have chosen to build and change the ***Priority / Section / Maintainer / Version / Provides / Description***

Architecture: all is suitable if you want to build the dummy package be compatible with all architectures, **if you need a specific *CPU architecture i386, i586, amd64 armpc etc. just set the proper one***

```
root@jericho:/home/hipo# vim xmame
```

```
root@jericho:/home/hipo# equivs-build xmame
dh_testdir
dh_testroot
dh_prep
dh_testdir
dh_testroot
dh_install
dh_install: Compatibility levels before 9 are deprecated (level 7 in use)
dh_installdocs
dh_installdocs: Compatibility levels before 9 are deprecated (level 7 in use)
dh_installchangelogs
dpkg-parsechangelog: warning:     debian/changelog(l5): badly formatted trailer line
LINE:  -- Georgi Georgiev  Mon, 25 Sep 2017 12:55:27 +0300
dpkg-parsechangelog: warning:     debian/changelog(l5): found end of file where expected more change data or trailer
dh_installchangelogs: Compatibility levels before 9 are deprecated (level 7 in use)
dh_compress
dh_fixperms
dh_installdeb
dh_installdeb: Compatibility levels before 9 are deprecated (level 7 in use)
dh_gencontrol
dpkg-gencontrol: warning:     debian/changelog(l5): badly formatted trailer line
LINE:  -- Georgi Georgiev  Mon, 25 Sep 2017 12:55:27 +0300
dpkg-gencontrol: warning:     debian/changelog(l5): found end of file where expected more change data or trailer
dh_md5sums
dh_builddeb

dpkg-deb: building package 'xmame' in '../xmame_0.182-1_all.deb'.
```

The package has been created.
Attention, the package has been created in the current directory, not in ".." as indicated by the message above!

Historically a very common use of Dummy debian package creation was when installing Qmail Mail server instead of Postfix (because some of the binary packages such as **bsd-mailx** used to be dependent on **Postfix**

Here is how the dummy postfix package was generated once the real one was already removed (because its presence on system was intereferring with already installed Qmail mail agent)

```
root@jericho:/home/hipo# equivs-control postfix
```

Command would generate a default dummy package skele template such as below:

```
### Commented entries have reasonable defaults.
### Uncomment to edit them.
# Source:
Section: misc
Priority: optional
# Homepage:

Standards-Version: 3.9.2
```

```
Package:
# Version:
```

```
# Maintainer: Your Name
# Pre-Depends:
# Depends:
# Recommends:
# Suggests:
# Provides:
# Replaces:
# Architecture: all
# Multi-Arch:
# Copyright:
# Changelog:
# Readme:
# Extra-Files:
# Files:
#
Description:
 long description and info
 .

 second paragraph
```

To make the postfix named package you can modify the equivs-control generated file postfix to look like so:

```
    Section: misc
Priority: optional

Standards-Version: 2.3.3
```

```
    Package: postfix-dummy
Version: 2.7.0
Section: mail
Maintainer: Eric Lubow
Provides: mail-transport-agent
Architecture: all

Description: Dummy Postfix package
```

```
    root@jericho:/home/hipo# equivs-build postfix
dh_testdir
dh_testroot
dh_prep
```

```
dh_testdir
dh_testroot
dh_install
dh_install: Compatibility levels before 9 are deprecated (level 7 in use)
dh_installdocs
dh_installdocs: Compatibility levels before 9 are deprecated (level 7 in use)
dh_installchangelogs
dpkg-parsechangelog: warning:     debian/changelog(l5): badly formatted trailer line
LINE:  -- Eric Lubow  Mon, 25 Sep 2017 12:52:46 +0300
dpkg-parsechangelog: warning:     debian/changelog(l5): found end of file where expected more change data or trailer
dh_compress
dh_fixperms
dh_installdeb
dh_installdeb: Compatibility levels before 9 are deprecated (level 7 in use)
dh_gencontrol
dpkg-gencontrol: warning:     debian/changelog(l5): badly formatted trailer line
LINE:  -- Eric Lubow  Mon, 25 Sep 2017 12:52:46 +0300
dpkg-gencontrol: warning:     debian/changelog(l5): found end of file where expected more change data or trailer
dh_md5sums
dh_builddeb

dpkg-deb: building package 'postfix-dummy' in '../postfix-dummy_2.7.0_all.deb'.
```

The package has been created.
Attention, the package has been created in the current directory,
not in ".." as indicated by the message above!

Once you have all your dummy packages built, just install the packages in a standard way with **dpkg** to install below 2 generated packages ***xmame_0.182-1_all.deb*** and ***postfix-dummy_2.7.0_all.deb*****:**

```
root@jericho:/home/hipo#  dpkg -i  xmame_0.182-1_all.deb; dpkg -i postfix-dummy_2.7.0_all.deb

...
```