

## Debian packages

- source packages
- binary packages

## Debian package manager

- install/remove debian packages
  - `apt-get`

```
$ apt-get install halma
$ apt-get remove halma
$ apt-get purge halma
```

```
$ apt-get dist-upgrade
```

## Debian package manager

- ```
$ apt-get source halma
```
- results in a directory of source code which was used to build the binary package

## Rebuilding a debian package

- see [excellent tutorial](http://people.connexer.com/~roberto/howtos/debcustomize) (<http://people.connexer.com/~roberto/howtos/debcustomize>)
- first we need to install the `pbuilder` environment and tools
  - allows us to build a package against a minimal pristine set of packages
  - keeps the possibly tainted host machine away from the build environment
  - the build environment is `chrooted` safely out of harms way

## Rebuilding a debian package

- ```
$ sudo apt-get install devscripts debhelper cdbs dpkg-dev
$ sudo apt-get install pbuilder gnupg
```

- run visudo and replace bob by your name

- ```
bob ALL = NOPASSWD: /usr/sbin/pbuilder
bob ALL = NOPASSWD: /usr/lib/pbuilder/pbuilder-satisfydep
```

## Edit ~/.pbuilderrc

- include these lines:

- ```
MIRRORSITE=http://mirror.ox.ac.uk/debian
OTHERMIRROR="deb http://floppsie.comp.glam.ac.uk/debian/"
DISTRIBUTION=squeeze
DEBEMAIL="Your Name <your@email.address>"
```

- and replace any contents as appropriate

## Create pbuilder chroot environment

- ```
sudo pbuilder
```

- creates an up to date chroot environment based on latest packages

## mybuild

- create a tiny shell script

- ```
#!/bin/bash
pdebuild --configfile ~/.pbuilderrc --buildsourceroot fak
--pbuilderroot sudo --buildresult /var/cache/pbuilder/re
--auto-deb sign $*
```

- ```
$ chmod 755 mybuild
```

## myrevdeb

- create another tiny shell script called, myrevdeb

```
#!/bin/bash
export EDITOR=emacs
export DEBEMAIL=you@example.com
debchange --nmu
```

- and make it executable via:

```
$ chmod 755 myrevdeb
```

## Download the source package

```
$ apt-get source halma
$ cd halma-1.0
```

- make changes to the source code
  - ensure that your command line terminal is still in the top level directory of the package

- now type:

```
$ myrevdeb
```

- add a simple statement indicating the nature of the modification

## Building the package

- from the same command line terminal which ran myrevdeb now execute

```
$ mybuild
```

- which should download all necessary dependencies for the package
  - chroot the environment safely away from the host system and build the package
  - the result will be placed in a directory specified by /etc/pbuilderrc