Development environment with debootstrap

This page contains a development machine setup with Wheezy (with all the development stuff) installed in chroot.

Prerequisites

Get a (Debian-based) system up and running, the idea of this setup is to have a more recent system for hardware compatibility, so at the time of writing Debian Jessie or Debian Stretch are the obvious choices. Ubuntu, Mint, etc. should work too.

More Debian based systems:

http://distrowatch.com/search.php?ostype=All&category=All&origin=All&basedon=Debian&notbasedon=None&desktop=All&architecture=All&package=All&rolling=All&status=Active

Debootstrap

Create a directory which we will use for the chroot environment.

e.g.:

```
/> mkdir ~/wheezy_chroot
```

Next, we run debootstrap to get a minimal Wheezy system:

```
/> sudo apt-get install debootstrap
/> sudo debootstrap wheezy ~/wheezy_chroot http://http.debian.net/debian/
```

(Debian FTW!)

References

- https://wiki.debian.org/Debootstrap

Chroot

Next step is to enter our new root, you can do this manually, or with a nice tool called schroot.

Manual

First step is to mount some directories from your system, this depends on the things you want to do inside chroot, here are the basic mounts:

```
/> sudo mount -t proc /proc ~/wheezy_chroot/proc
/> sudo mount -t devpts /dev/pts ~/wheezy_chroot/dev/pts
```

Next step, is to chroot:

```
/> sudo chroot ~/wheezy_chroot /bin/bash --login
```

If you do it manually, best is to make sure your prompt changes once inside chroot, to avoid troubles.

e.g.:

in chroot

```
/> echo 'PS1="MY_WHEEZY_CHROOT:\w# "' >> ~/.bashrc
```

Schroot

Or you can use this tool to do all this stuff for you, especially when you have more than one chroot directory.

on system

```
/> sudo apt-get install schroot
```

edit the configuration at:

`/etc/schroot/schroot.conf`

e.g.

```
[pfe]
type=directory
description=Wheezy for pfe development
aliases=default
directory=/home/jo/wheezy_chroot
## A comma-separated list of users which are allowed access to the chroot.
users=jo
## A comma-separated list of groups which are allowed access to the chroot.
# groups=
## A  comma-separated  list of users which are allowed password-less root
access to the chroot.
# root-users=
## A comma-separated list of groups which are allowed password-less root
access to the chroot.
root-groups=root
## The behaviour of the chroot setup scripts may be customised on a
per-chroot basis by setting a specific configuration profile.
## This is a custom profile, so will not work out of the box, see below for
more details, change to 'minimal' or 'default' to start.
profile=pfe
```

(In the config above 'default' is added to the aliases so running schroot without arguments will enter that chroot instance.)

Enter chroot 'pfe':

in system

```
/> schroot -c pfe
```

Execute command 'cmd' in chroot 'pfe':

in system

```
/> schroot -c pfe cmd
```

The setup behavior of the chroot is determined by the 'profile' setting in your 'schroot.conf'.

> *profile=profile-name*
>
> *The behaviour of the chroot setup scripts may be customised on a per-chroot basis by setting a specific configuration profile.*
> *The directory is relative to /etc/schroot. The default is ‘default’. The files in this*
> *directory are sourced by the setup scripts, and so their behaviour may be customised by selecting the appropriate profile.*
> *Alternatives are ‘minimal’ (minimal configuration), ‘desktop’ (for running desktop applica*
> *tions in the chroot, making more functionality from the host system available in the chroot) and ‘sbuild’ (for using the chroot for*

> *Debian package building). Other packages may provide additional profiles. The default values of the keys setup.config, setup.copyfiles, setup.fstab and setup.nssdatabases are set based upon the profile setting.*
>
> *...*

Check out the files in the profile directory ('/etc/schroot/<profile-name>').

Default the 'home' directory is mounted into the chroot environment, resolv.conf is copied, nssdatabases from the system are used (passwd, shadow, group, gshadow, services, protocols, networks,hosts, etc.).

My current setting is set to a custom profile 'pfe' (modified 'default').

Note that schroot will try to go to the same directory as the directory you executed schroot in (override this by using --directory).

References:

- https://wiki.debian.org/Schroot

Inside new root

Once inside you can setup your system, for me:

in chroot

```
/> adduser jo
/> apt-get install vim sudo
/> adduser jo sudo
/> su - jo
```

Next (for me):

Setup FE Environment

Accessing graphical applications inside the chroot

For me it's necessary to run my IDE (PyCharm) from chroot, to run X applications from chroot:

in chroot

```
/> export DISPLAY=:0.0
```

in system

```
/> xhost +
```

e.g. starting Pycharm from main system to run in a chroot environment:

chroot_pycharm.sh

```
#!/bin/bash
xhost +
schroot -c pfe /home/jo/Apps/pycharm/bin/pycharm.sh
```

pycharm.sh

```
#!/bin/sh
#
# ---------------------------------------------------------------------
# PyCharm startup script.
# ---------------------------------------------------------------------
#
export DISPLAY=:0.0
. /etc/profile
export JAVA_HOME=/home/jo/Apps/jdk8

...
```

References

- https://help.ubuntu.com/community/BasicChroot#Accessing_graphical_applications_inside_the_chroot

Troubleshooting

- **problem:** CLTB fails to start with a multiprocessing error.
  solution: add following to /etc/fstab (and mount):

  ```
  none /dev/shm tmpfs rw,nosuid,nodev,noexec 0 0
  ```

- **problem**: problems with locale (e.g. when starting pfe apps)
  solution:

  ```
  /> sudo apt-get install locales
  /> sudo dpkg-reconfigure locales # select en_US with UTF8
  ```

- **problem**: when using schroot, and you get a lot (*really* **a lot**) of mounts
  solution: edit the **/etc/default/schroot** file and modify the setting of **SESSIONS_RECOVER** to be "**end**" instead of "recover".

More

- If you need the Wheezy kernel for development, it's also possible to install a kernel, setup boot, adjust grub and boot into the chroot system.
- Might be useful when upgrading critical systems, test upgrade in chroot, adjust grub only if successful.