# LTIB Build Host Setup

## Setting up a Linux host for LTIB builds on Ubuntu 9.04

## Important note: Please be methodical and Do Not Skip Any Steps.

LTIB is very picky and will likely not work unless this build environment is correct.

- **Ubuntu 9.04 (Jaunty) Desktop** is the **only supported Linux distribution** for LTIB build hosts.
    - Both the 32 Bit and the 64 Bit install from the Desktop CD are supported.
    - Other versions of Ubuntu are not supported and may have build issues.
- The install ISO image is available from:
    - http://old-releases.ubuntu.com/releases/9.04/

### Proxy

- The Ubuntu package mangager requires network access to download packages. If the Ubuntu host is on a network that requires a network proxy, this can be configured in the Gnome menu option "System -> Preferences -> Network Proxy"
- Once the settings are filled in, ensure that you click on the "Apply System Wide" button.
- If you had a terminal window open before setting the proxy, it will have the old settings.  Open a new terminal window to get the new values.
- To check the settings run the following command:

```
user@ubuntu:~$ env | grep proxy
http_proxy=http://example:8080/
ftp_proxy=ftp://example:8080/
https_proxy=https://example:8080
```

### Update sources.list to point to old-releases repository

- Ubuntu 9.04 is considered an old release, so the package manager can't get packages from Canonical's normal repository.
- A 'sources.list' is included in the documentation package.  This sources.list adjusts the package manager's source paths to point to 'old-releases.ubuntu.com'.
- Copy the included '**sources.list**' file to **/etc/apt/sources.list**
- If the '**sources.list**' is not included, copy the following to a file and copy it to **/etc/apt/sources.list**

```
# deb cdrom:[Ubuntu 9.04 _Jaunty Jackalope_ - Release i386 (20090420.1)]/ jaunty main restricted
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.

deb http://old-releases.ubuntu.com/ubuntu/ jaunty main restricted
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://old-releases.ubuntu.com/ubuntu/ jaunty-updates main restricted
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://old-releases.ubuntu.com/ubuntu/ jaunty universe
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty universe
deb http://old-releases.ubuntu.com/ubuntu/ jaunty-updates universe
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://old-releases.ubuntu.com/ubuntu/ jaunty multiverse
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty multiverse
deb http://old-releases.ubuntu.com/ubuntu/ jaunty-updates multiverse
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty-updates multiverse

## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
# deb http://us.old-releases.ubuntu.com/ubuntu/ jaunty-backports main restricted universe multiverse
# deb-src http://us.old-releases.ubuntu.com/ubuntu/ jaunty-backports main restricted universe
multiverse

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu jaunty partner
# deb-src http://archive.canonical.com/ubuntu jaunty partner

deb http://old-releases.ubuntu.com/ubuntu/ jaunty-security main restricted
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty-security main restricted
deb http://old-releases.ubuntu.com/ubuntu/ jaunty-security universe
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty-security universe
deb http://old-releases.ubuntu.com/ubuntu/ jaunty-security multiverse
deb-src http://old-releases.ubuntu.com/ubuntu/ jaunty-security multiverse
```

## Upgrade to the latest packages

- Open up **System -> Administration -> Update Manager**
- Click on **Settings**
- Open the **Updates** Tab
- Set '**Release upgrade**' to '**Neve**r'. That makes the option to upgrade to Karmic go away.
- Close the settings dialog box.
- Click on '**Check**' to check for upgraded packages. It will look for packages that are upgraded from the version that is installed on your box.
- Choose to **install the upgrades.** This will take a while on a freshly installed box.

## Sudoers

- LTIB is executed as a regular user but requires root privileges during execution.  To accomodate this, the sudoers file needs to be modified.
- Run this command to edit the sudoers file:

```
sudo visudo
```

- Add the following line to the end of the sudoers file.

```
%admin ALL = NOPASSWD: /usr/bin/rpm, /opt/freescale/ltib/usr/bin/rpm
```

- The %admin specifies any user in the group ‘admin’ thus your user name must be in this group. If that is not the case, a similar line can be added for each user. Please refer to the man page on sudoers for additional information:

```
man sudoers
```

- Note: If the sudoer’s file is not updated before running ltib, the recommended sudoer’s line will be shown when ltib exits.

## Install host packages needed by LTIB

- The default Ubuntu Desktop installation requires additional packages installed for LTIB.
- The following script is included in the documentation package as the "host-setup.sh" script.   Running the script will install the required packages.
- If you have this document, but don't have the script, you can past the following to a text file and set the permissions as executable, and run it to install the packages.

```
#!/bin/bash

# Install packages needed by LTIB
sudo aptitude -y install gettext libgtk2.0-dev rpm bison m4 libfreetype6-dev
sudo aptitude -y install libdbus-glib-1-dev liborbit2-dev intltool
sudo aptitude -y install ccache ncurses-dev zlib1g zlib1g-dev gcc g++ libtool
sudo aptitude -y install uuid-dev liblzo2-dev
sudo aptitude -y install tcl dpkg

# Packages required for 64-bit Ubuntu
# Do "uname -a" and see if the word "x86_64" shows up.
if uname -a|grep -sq 'x86_64'; then
    sudo aptitude -y install ia32-libs libc6-dev-i386 lib32z1
fi

# The following recommended for Linux development.
# They are not required by LTIB.
sudo aptitude -y install gparted emacs22-nox openssh-server
sudo aptitude -y install nfs-common nfs-kernel-server lintian
sudo aptitude -y install git-core git-doc git-email git-gui gitk
sudo aptitude -y install diffstat indent tofrodos fakeroot doxygen uboot-mkimage
sudo aptitude -y install sendmail mailutils meld atftpd sharutils
sudo aptitude -y install manpages-dev manpages-posix manpages-posix-dev linux-doc
sudo aptitude -y install vnc4server xvnc4viewer
```

## Set an upper limit on ccache

- LTIB uses ccache on a per-user basis to speed up compilation. The cache that LTIB uses exists as a .ccache directory in each user's home directory. This directory can grow to be quite large if no upper limit is set.
- To set the upper limit of your ccache do these two steps. The first step sets the upper limit, for example, 50 Meg.  The second step clears the cache to meet the limit.

```
ccache -M 50M
ccache -c
```

## Configure tftp server

- Install atftpd by running:

```
sudo aptitude -y install atftpd
```

- Configure **atftpd** by editing **/etc/inetd.conf** and **/etc/default/atftpd**.
- In both files, change the default export path (it is either /usr/var/tftpboot or /var/lib/tftpboot) to / or whatever other directory you want to be able to download from.
- Then reboot your box.

## Configure nfs server

- Install nfs server by running:

```
sudo aptitude -y install nfs-common nfs-kernel-server
```

- To configure nfs server, add lines to **/etc/exports** like this (below example exports /home and everything under it.

```
/home *(rw,no_root_squash)
```

- Then restart the nfs server:

```
sudo /etc/init.d/nfs-kernel-server restart
```

## Setting up samba shares

- Use the GUI to share folders. For instance:
    - Click on "**Places --> Home Folder**"
    - Right click on "**Public**" folder and select "**Sharing Options**"
    - Click on "**Share Folder**" and then give this share a unique name (i.e. it will be confusing if multiple people all have a 'public' share on the same machine).
    - If this is the first samba share you've done, it will ask for your password and go ahead and install samba for you.
- Alternatively, samba can be configured from the command line.
    - Install samba

      ```
      sudo aptitude -y install samba
      ```

    - Give yourself a samba password. Can be blank.

      ```
      sudo smbpasswd -a YOURUSERNAME
      ```

    - Edit the /etc/samba/smb.conf file and add a section such as this example:

      ```
      [wilma]
       comment = wilma
       read only = no
       locking = no
       path = /home/wilma/Public
       create mask = 0644
       directory mask = 0770
      ```

    - Restart samba:

      ```
      sudo /etc/init.d/samba restart
      ```

## Remote desktop

- Ubuntu comes with a remote desktop installed by default.
    - Enable it on the remote machine in **System -> Preferences -> Remote Desktop**
    - Connect using **Applications -> Internet -> Remote Desktop Viewer**
- Another option is VNC
    - On remote machine:

        ```
        sudo aptitude -y install vnc4server
        ```

        - See what vnc displays are already running for other users by doing "**ps ax | grep vnc**". The display numbers will have a colon like ':4'. Choose a number that is not already being used.

            ```
            # vncserver :1
            # vncserver -kill :1
            ```

        - edit your **~/.vnc/xserver** to be like this:

            ```
            #!/bin/sh

            # Uncomment the following two lines for normal desktop:
            # unset SESSION_MANAGER
            # exec /etc/X11/xinit/xinitrc

            [ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
            #[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
            #xsetroot -solid grey
            vncconfig -iconic &
            # xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
            #twm &
            gnome-session &
            ```

        - Now start it again:

            ```
            vncserver :$mysession -geometry 1024x768 -depth 16
            ```

    - On local machine:
        - Install the viewer.

            ```
            sudo aptitude -y install xvnc4viewer
            ```

        - Start vncviewer.

            ```
            vncviewer HOSTNAME:DISPLAY
            ```

            - where HOSTNAME is the remote hostname and DISPLAY is the display number you started above