LTIB Build Host Setup

Setting up a Linux host for LTIB builds

- We support building using Ubuntu 9.04 (Jaunty) installed from the 32 or 64 bit Desktop Ubuntu install cd.
- Other versions of Ubuntu are not currently supported and may have build issues.

Sudoers

- Run 'sudo visudo' so you can edit the sudoer's file. Add the following line to the end of the sudoers file. This is needed for people to be able to use LTIB. This assumes that all your developers have administrator priviledges on this host. If that is not the case, a similar line can be added for each user.

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

Update to the latest packages

- Open up System -> Administration -> Update Manager
- Click on Settings
- Open the Updates Tab
- Set 'Release upgrade' to 'Never'. 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.

Install host packages needed by LTIB

This document assumes you are using Ubuntu. Not a requirement, but the packages may be named differently and the method of installing them may be different.

```
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
```

Packages required for 64-bit Ubuntu

If you don't know whether you have 64-bit Ubuntu installed, do "uname -a" and see if the word "x86_64" shows up.

```
sudo aptitude -y install ia32-libs libc6-dev-i386 lib32z1
```

The following are optional They are not required by LTIB but are recommended for Linux development:

```
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
```

Configure tftp server

After installing atftpd, configure it 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 change the default export path to whatever directory you want to be able to download from.
Actually it might be the case that only configuring /etc/default/atftpd will do it, I'm not sure.
Then reboot your box.

Configure nfs 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.

You may have to enable some wide open permissions to be able to get to folder, not sure.

Setting up ccache

LTIB uses ccache 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, you can specify anything, my example is 50 Meg. I'm not sure what an optimal size would be here. The second step clears the cache to meet the limit.

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

If you are not able to run these commands you can do 'sudo apt-get install ccache'. Or use the version that LTIB intalled under /opt which isn't in your PATH unless LTIB is doing a build (/opt/freescale/ltib/usr/bin/ccache). If you want to see how ccache is set up currently, do "ccache -s". For more info on ccache, read the manpage.

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

See Also

Setting Up a Development System, a Step-by-step guide