

P1 Tower System Module Linux Target Image Builder (LTIB) Quick Start Guide

1. Required Linux Host Environment:

Install one of the Linux distributions from the below choices.

i. Install Linux:

Redhat: 7.3, 8.0, 9.0

Fedora Core: FC1/2/3/4/5/8/10

Debian: 3.1r0, 4.0

OpenSuse: 8.2,9.1, 9.2, 10.0, 10.2, 10.3

Redhat Enterprise: 4, 5.3

Ubuntun Desktop: 6.10, 7.04, 7.10, 8.04, 8.10, 9.04, 9.10, 10.04(64bit)

ii. LTIB relies on these packages to be installed so ensure your host machine has these installed:

Rsync	(any version)	Installer: to copy the packages
Perl	(>=5.6.1)	Ltib script
Sudo	(any verision)	To run "rpm install"
Wget	(any version)	To download package/paches
Rpm-build	(any version)	Need by rpm to do actual building
Rpm	(any version)	To build initial rpm-fs host package
Glibc	(>=2.2.x)	To build/run host package
Libstdc++-devel	(any version)	To build rpm-fs host package
Binutils	(>=2.11.93)	To build host package
Gcc	(>=2.96)	To build host package
Gcc-c++	(>=2.26)	To build rpm-fs host package
Zlib-devel	(any version)	To build rpm-fs and mtd-utils host package
Ncurses	(>=5.1)	To build lkc host package
Ncurses-devel	(>=5.1)	To build lkc host package
M4	(any version)	Needed by bison
Bison	(any version)	To build lkc host package
Flex	(any version)	To build lkc host package
Texinfo	(any version)	To build genext2fs host package
Autoconf	(>=2.54)	To build automake host package
Libtool	(>=1.4.2)	To build libusb host package

2. LTIB Environment Set Up

On the Linux Hosted PC, download the ltib-p1025twr-2.6.35.14-20120307.iso to the directory that holds all downloads, e.g. /home/freescale/download where freescale is the user name and download is the directory created to hold all internet downloads. Also Create a directory named “p1025twr” in /mnt directory. Then start a terminal and type

i. `mount -o loop /home/freescale/download/ltib-p1025twr-2.6.35.14-20120307.iso /mnt/p1025twr`

Note: the above command needs to be run in root terminal.

ii. `cd /mnt/p1025twr`

iii. `./install`

After the three steps, the ltib package should be installed. Switch the path to the install directory which will be shown by LTIB install script in the very end.

Note: For p1025TWR ISO, you may need to manually extract the “pkgs” directory from the ISO to replace the directory of /opt/freescale/pkgs after the install.

3. Install Tool Chain

After switching to the ltib-p1025twr-2.6.35.14-20120307 ltib directory, if you run “./ltib” command for the first time, it will install tool chain automatically, subsequently this command will compile u-boot, kernel and application packages.

i. `./ltib`

Then the tool chain is installed in below folder:

`/opt/freescale/usr/local/gcc-4.1.78-eglibc-2.5.78-1/powerpc-e300c3-linux-gnu`

4. U-boot source code & Compilation Steps

To get the u-boot source code:

i. `./ltib -m prep -p u-boot`

The source code is at the <P1025TWR ltib directory>/rpm/BUILD/<u-boot>

To compile the u-boot:

i. `./ltib -m scbuild -p u-boot`

u-boot image (u-boot.bin) is at the <P1025TWR ltib directory>/rpm/BUILD/<u-boot>

To upgrade the image into the board:

a. Update u-boot

i. `tftp 1000000 u-boot.bin`

ii. `protect off all`

iii. `erase 0xeff80000 +0x80000`

- iv. cp.b 0x1000000 0xeff80000 0x80000

- b. Update dtb:
 - v. tftp 0xc00000 twr-p1025_32b.dtb
 - vi. erase 0xefe80000 +0x40000
 - vii. cp.b 0xc00000 0xefe80000 0x40000

- c. Update uImage:
 - viii. tftp 1000000 uImage
 - ix. erase 0xefa80000 +0x400000
 - x. cp.b 0x1000000 0xefa80000 0x400000

- d. Update rootfs:
 - xi. tftp 0x2000000 root.squashfs
 - xii. erase 0xec000000 +0x3a80000
 - xiii. cp.b 0x2000000 0xec000000 0x01c00000

- e. Update FSL_QE
 - xiv. tftp 0x1000000 fs1_qe_ucose_1021_10_A.bin
 - xv. erase 0xefec0000 +0x40000
 - xvi. cp.b 0x1000000 0xefec0000 0x40000

- f. Modify u-boot environment:
 - xvii. setenv bootargs "root=/dev/mtdblock0 rw console=ttyS0,115200 init=/etc/preinit
cache-sram-size=0x10000 cache-sram-offset=0xffff00000" <= this line should be the
continuous segment of the above line
 - xviii. setenv bootcmd 'bootm 0xefa80000 - 0xefe80000'

- g. saveenv

5. Application Code Compilation Steps

Prepare the application source code and the rootfs for the board.

- i. Enter the cross compile environment: ./ltib -m shell
- ii. Switch to the application source code
- iii. Configure e.g. ./configure --host=powerpc
- iv. Compiling
- v. Install the application binary file and configuration to the rootfs
- vi. Package the rootfs

6. Quick way to compile and release an application:

- i. Configure compile tool chain environment:

```
export PATH=$PATH; /opt/freescale/usr/local/gcc-4.1.78-eglibc-2.5.78-1/powerpc-e300c3-linux-gnu/bin
export CROSS_COMPILE= powerpc-e300c3-linux-gnu-
export ARCH=powerpc
```

Or enter ltib shell mode:
`./ltib -m shell`
- ii. Enter application folder, and do standard linux compile instruction:

```
./configure --host=powerpc
make
```
- iii. Copy application image and configuration into rootfs directory, which will be used to create rootfs.ext2.uboot or other rootfs image.
- iv. Create rootfs image, different rootfs format has different command line. [In below example, rootfs.tmp is the rootfs directory to make rootfs image.]
if want to create ramdisk filesystem , please use below command:

```
genext2fs -U -b 118912 -i 7114 -d /rootfs.tmp ./rootfs.ext2
gzip ./rootfs.ext2
mkimage -n 'uboot ext2 ramdisk rootfs' -A ppc -O linux -T ramdisk -C gzip
-d ./rootfs.ext2.gz ./rootfs.ext2.gz.uboot
```

If wanting to create squashfs filesystem, please follow below command:

```
/opt/freescale/ltib/usr/bin/mksquashfs4 ./rootfs.tmp ./release/root.squashfs
-nopad -noappend -root-owned -comp lzma -processors 1
```

7. Other Information:

In the Freescale P1025TWR we implement Web server based on OpenWRT webif package, all the GUI pages are implemented with shell script files under `/www/cgi-bin/webif`. Which can be customized.

Appendix

Steps to setup:

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 privileges 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.

```
#!/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
```

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](#)

