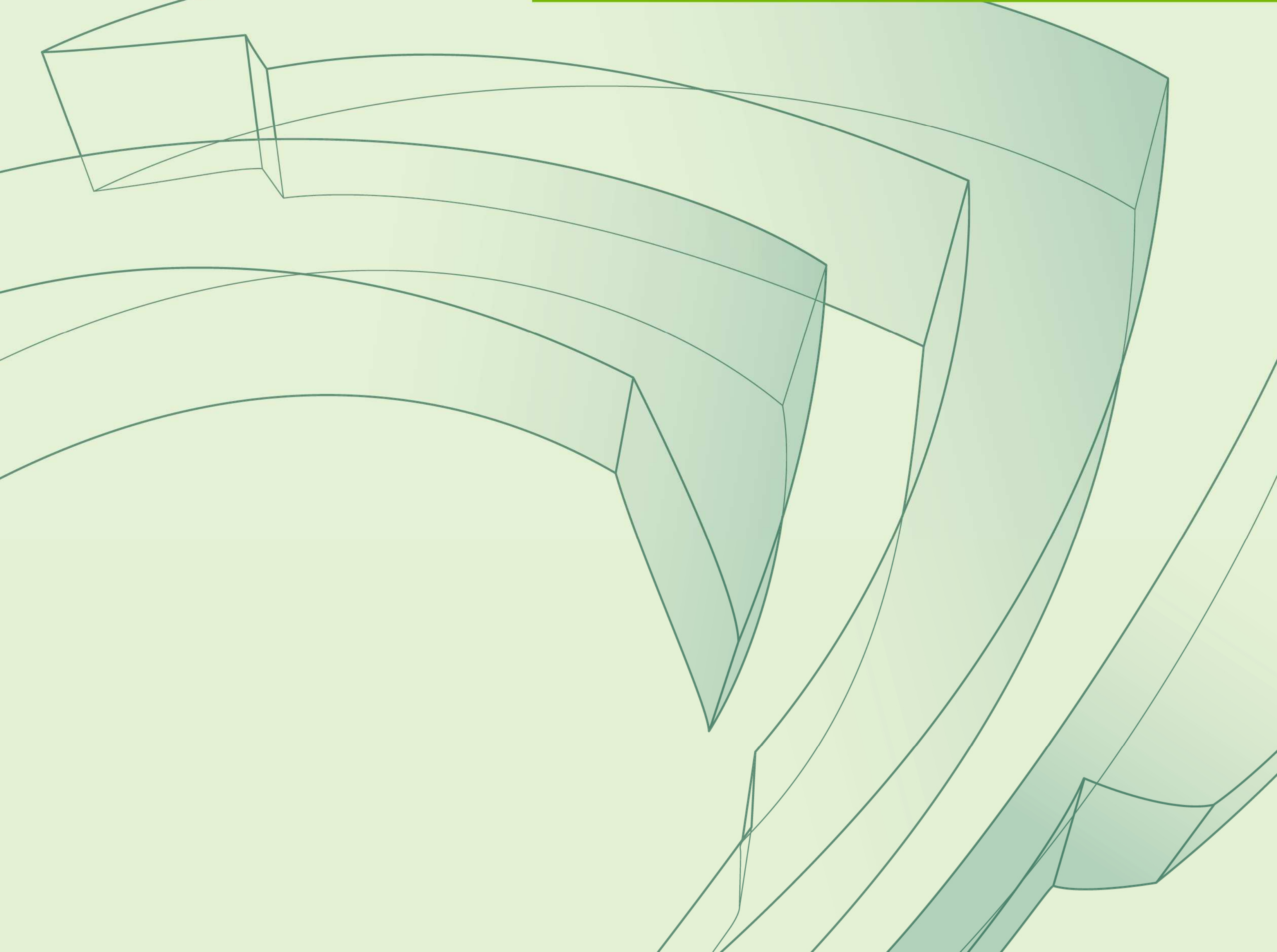
NVIDIA TEGRA LINUX DRIVER PACKAGE

PG_06076-R16 | September 4, 2012 | R16 Release

Developers' Guide



Note: Apparent hyperlinks in this document are a legacy of the HTML version and may not operate as expected in the PDF version.

Overview

Welcome to *NVIDIA Tegra Linux Driver Package Developers' Guide*. Engineers can use this document to learn about working with NVIDIA® Tegra® Linux Driver Package, sometimes referred to as Linux for Tegra (L4T).

> **Important**: This documentation is preliminary and subject to change. Please see your NVIDIA representative for additional information and to request documentation updates.

Read the following sections to get started using Tegra Linux Driver Package.

- Package Manifest—describes the top level directories and files installed when expanding the release TAR file.
- Getting Started—provides requirements and set up information to help you get started using the package.
- U-Boot Guide—describes the U-Boot implementation for L4T.
- Building Crosstool-ng Toolchain and glibc—provides instructions to build the Crosstool-ng supplied cross toolchain suite version 4.5.3 and the glibc suite with an Ubuntu host machine
- Software Features—describes the software features supported by the release.
- Licenses—provides license information for Tegra and 3rd-party software.
- Appendix—provides an example .config file for the Crosstool-ng toolchain.
- FAQs—provides answers to frequently asked questions.
- Glossary—provides definitions of key terms.

Package Manifest

The NVIDIA® Tegra® Linux Driver Package is provided in the following tar file:

```
<platform>_Tegra-Linux-
<release_num>.<version_num>_<release_type>.tbz2
```

where

- <release_num> is the branch number of the release, such as R16.
- <version num> is the version number of the build, such as 1.0 for the first build.
- <release_type> is `armel` (for softfp ABI) or `armhf` (for hard-float ABI).

The following table lists the top level directories and files that are created when you expand the tar file.

Filename	Description
./rootfs	Directory used as a staging directory for the root filesystem
./rootfs/README.txt	This file explains the need to copy the sample file system here.
./kernel	Directory containing the kernel images and kernel modules
./bootloader	Directory containing the boot loader and related components.
./bootloader/<platform>	Directory containing platform-specific files
./bootloader/<platform>/BCT	Directory containing the platform-specific BCT files
./bootloader/<platform>/cfg	Directory containing the appropriate cfg files.
./nv_tegra	Directory containing the NVIDIA drivers and sample applications.
./nv_tegra/nv_sample_apps	Directory containing the NVIDIA sample applications.
./source_sync.sh	A script that downloads kernel and uboot source
./apply_binaries.sh	A script to apply `nv_tegra` components.
./flash.sh	A script that flashes the boot loader and kernel from the package.
./zImage_to_uimg.sh	A script that creates the `vmlinux.uimg` with

	mkimage for use as the kernel image for u-boot

Note: The `<platform>` in `/bootloader/<platform>` specifies the development system. For Tegra 3 series, the code-name is "cardhu", for Tegra 2 series, the code-name is "ventana".

Documentation

Tegra Linux Driver Package (L4T) also includes the following documentation:

- Tegra_Linux_Driver_Package_Release_Notes_<ver>.pdf
- Tegra_Linux_Driver_Package_Documents_<ver>.tar

Where <ver> is the version of the release, such as R15. Both the "armel" and "armhf" software versions are documented.

Section Overview

This section provides information about the contents of the L4T tar file and includes the following topics:

- Kernel
- Boot Loader
- NV Tegra
- Nvgstapps TBZ2
- Config TBZ2
- NVIDIA Drivers TBZ2

Kernel

This section describes the major components included in the `./kernel` directory.

Filename	Description
./zImage	A kernel binary image.
./LICENSE	A license file for "GNU GENERAL PUBLIC LICENSE".
./kernel_supplements.tbz2	Loadable kernel modules specific to the included kernel zImage built with the defconfig enabled for the device.
./kernel/vmlinux.uimg	A u-boot kernel binary image.

Boot Loader

This section describes the files provided in the `./bootloader` directory.

Filename	Description
./mkbootimg	A tool used for img creation.
./nvflash	The NVIDIA flashing tool.
./LICENSE.mkbootimg_and_mkubootscript	A license file for the mkbootimg and mkbootscript tools.
LICENSE.u-boot_and_mkimage	A license file for u-boot and the mkimage tool.
./mkimage	A u-boot tool for vmlinux.uimg creation.
./mkubootscript	A tool for flashing u-boot
./<platform>	<platform> specifies the development system, cardhu for Tegra3, or ventana for Tegra 2 devices.
./<platform>/fastboot.bin	The Fastboot-versioned boot loader binary file.
./<platform>/<platform>_emmc.hush	The local storage device u-boot hush file.
./<platform>/<platform>_net.hush	The network u-boot hush file.
./<platform>/u-boot.bin	The u-boot binary image.
./<platform>/BCT	Platform-specific BCT directory.
./<platform>/BCT/ cardhu_12Mhz_H5TC2G83BFR_333Mhz_1GB_emmc_SDIN5C2-16G_x8.bct	BCT for Cardhu.
./<platform>/BCT/ E1186_Hynix_1GB_H5TC2G83BFR-PBA_375MHz_110622_sdmmc4_x8.bct	BCT for Cardhu.
./<platform>/BCT/ ventana_A03_12MHz_EDB8132B1PB6DF_300Mhz_1GB_emmc_THGBM1G6D4EBAI4.bct	BCT for Ventana.
./<platform>/cfg	Platform-specific CFG directory.
./<platform>/cfg/gnu_linux_fastboot_emmc_full.cfg	Platform-specific CFG file.
./<platform>/cfg/gnu_linux_fastboot_emmc.cfg	Platform-specific CFG file.

NV Tegra

This section describes the major components included in the `./nv_tegra` directory.

Filename	Description

./config.tbz2	Configuration files specific to the sample filesystem.
./LICENSE	Tegra software license.
./nvidia_drivers.tbz2	NVIDIA driver components
./nv_sample_apps/LICENSE.gst-openmax	License file for libgstomx.so and libgstnvxvimagesink.so included in nvgstapps.tbz2.
./nv_sample_apps/nvgstapps.tbz2	The nvgstplayer and nvgstcapture multimedia test applications.
./nv_sample_apps/nvgstcapture_README .txt	Instructions on how to run the nvgstcapture application.
./nv_sample_apps/nvgstplayer_README. txt	Instructions on how to run the nvgstplayer application.

Nvgstapps TBZ2

This section describes the files included in the following file:

```
./nv_tegra/nv_sample_apps/nvgstapps.tbz2
```

Filename	Description
./usr	-
./usr/bin	-
./usr/bin/nvgstcapture	The multimedia capture camera application
./usr/bin/nvgstplayer	The multimedia video player application.
./usr/lib	-
./usr/lib/gstreamer-0.10	-
./usr/lib/gstreamer-0.10/libgstnvxvimagesink.so	Video sink.
./usr/lib/gstreamer-0.10/libgstomx.so	OpenMax driver.

Config TBZ2

This section describes the files included in the following file:

```
./nv_tegra/config.tbz2
```

Filename	Description
./etc	-
./etc/init	-
./etc/init/nv.conf	An NVIDIA-specific initialization script.

./etc/init/ttyS0.conf	An initialization script for getty on ttyS0.
./etc/udev	-
./etc/udev/rules.d	-
./etc/udev/rules.d/99-tegra-devices.rules	Sets permissions for Tegra devices.
./etc/X11	-
./etc/X11/xorg.conf	The xorg configuration file.

NVIDIA Drivers TBZ2

This section describes the files included in the following file:

```
./nv_tegra/nvidia_drivers.tbz2
```

Filename	Description
./etc	-
./etc/wpa_supplicant.conf	A sample WPA supplicant
./etc/nv_tegra_release	The tegra driver versioning file.
./lib/firmware/	-
./lib/firmware/nvavp_os_*.bin	NVIDIA AVP Kernel firmware.
Restricted codec: ./lib/firmware/nvavp_vid_ucode_alt.bin	NVIDIA video decoders.
Restricted codec: ./lib/firmware/nvmm_aacdec.axf	AAC decoder. Not included in the NVIDIA driver release; available through a separate software license agreement.
Restricted codec: ./lib/firmware/nvmm_adtsdec.axf	ADTS decoder running on AVP. Not included in the NVIDIA driver release; available through a separate software license agreement.
Restricted codec: ./lib/firmware/nvmm_h264dec.axf	H.264 video decoder. Not included in the NVIDIA driver release; available through a separate software license agreement.
Restricted codec: ./lib/firmware/nvmm_h264dec2x.axf	H.264 video decoder driver. Not included in the NVIDIA driver release; available through a separate software license agreement.
Restricted codec: ./lib/firmware/nvmm_jpegdec.axf	JPEG image decoder. Not included in the NVIDIA driver release; available through a separate software license agreement.
Restricted codec: ./lib/firmware/nvmm_jpegenc.axf	JPEG Encoder. Not included in the NVIDIA driver release; available through a separate software license agreement.
Restricted codec: ./lib/firmware/nvmm_manager.axf	Multimedia Manager Kernel driver. Not included in the NVIDIA driver release; available through a separate software

	license agreement.
Restricted codec: ./lib/firmware/nvmm_mp3dec.axf	MP3 decoder. Not included in the NVIDIA driver release; available through a separate software license agreement.
Restricted codec: ./lib/firmware/nvmm_mpeg4dec.axf	MPEG-4 video decoder. Not included in the NVIDIA driver release; available through a separate software license agreement.
Restricted codec: ./lib/firmware/nvmm_service.axf	NVIDIA multimedia services firmware. Not included in the NVIDIA driver release; available through a separate software license agreement.
./lib/firmware/nvrm_avp.bin	**Applies to:** Ventana: AVP kernel firmware.
./lib/firmware/nvrm_avp*.bin	**Applies to:** Cardhu: AVP kernel firmware.
./lib/modules/	-
./usr	-
./usr/lib	-
./usr/lib/libardv_dynamic.so	GLES2 graphics support library
./usr/lib/libcgdrv.so	NVIDIA Tegra OpenGL ES 2.0 shader compiler.
./usr/lib/libEGL.so	OpenGL ES driver file.
./usr/lib/libGLESv1_CM.so	OpenGL ES driver file.
./usr/lib/libGLESv2.so	OpenGL ES driver file.
./usr/lib/libjpeg.so	Accelerated libjepeg library for Tegra.
./usr/lib/libKD.so	OpenKODE driver.
./usr/lib/libnvapputil.so	Host (x86) shared object for application utilities.
./usr/lib/libnvavp.so	User-space interface to the AVP for audio/video acceleration via the nvavp kernel driver.
./usr/lib/libnvcwm.so	Compositing Window Manager library.
./usr/lib/libnvdc.so	DC driver file.
./usr/lib/libnvddk_2d.so	DDK 2D.
./usr/lib/libnvddk_2d_v2.so	DDK 2D.
./usr/lib/libnvddk_disp.so	Display abstraction driver file.
./usr/lib/libnvddk_kbc.so	NVIDIA keyboard controller driver.
./usr/lib/libnvddk_mipihsi.so	High-speed interface for MIPI.
./usr/lib/libnvddk_nand.so	NVIDIA NAND driver.
./usr/lib/libnvddk_se.so	NVIDIA Security Engine driver.
./usr/lib/libnvddk_snor.so	NVIDIA SNOR driver.
./usr/lib/libnvddk_spif.so	NVIDIA SPIF driver.
./usr/lib/libnvddk_usbphy.so	NVIDIA USB phy driver.
./usr/lib/libnvdispatch_helper.so	NvRM daemon dispatch helper.
./usr/lib/libnvglsi.so	GLES2 graphics support library

./usr/lib/libnvmedia_audio.so	Interface for audio decoder.
./usr/lib/libnvmm_audio.so	Audio codecs and components.
./usr/lib/libnvmm_camera.so	Multimedia camera driver file.
./usr/lib/libnvmm_contentpipe.so	Content pipe implementation (file source abstraction).
./usr/lib/libnvmm_image.so	Image codecs.
./usr/lib/libnvmmlite_audio.so	NVIDIA Multimedia audio driver.
./usr/lib/libnvmmlite_image.so	NVIDIA Multimedia image driver.
./usr/lib/libnvmmlite.so	NVIDIA Multimedia driver.
./usr/lib/libnvmmlite_utils.so	NVIDIA Multimedia utilities.
./usr/lib/libnvmmlite_video.so	NVIDIA Multimedia video driver.
./usr/lib/libnvmm_manager.so	Multimedia Manager kernel driver.
./usr/lib/libnvmm_parser.so	Parser.
./usr/lib/libnvmm_service.so	Multimedia Framework kernel services HAL.
./usr/lib/libnvmm.so	NVIDIA Multimedia Framework.
./usr/lib/libnvmm_utils.so	Multimedia Framework utilities.
./usr/lib/libnvmm_video.so	NVIDIA Multimedia Framework.
./usr/lib/libnvmm_writer.so	3GP writer block on CPU.
./usr/lib/libnvodm_disp.so	ODM kit display driver.
./usr/lib/libnvodm_dtvtuner.so	Applies to: Ventana and Cardhu releases: Tegra development platform ODM adaptation for digital TV tuner.
./usr/lib/libnvodm_imager.so	Tegra development platform ODM adaptation for imager.
./usr/lib/libnvodm_misc.so	ODM kit.
./usr/lib/libnvodm_query.so	ODM Query interface.
./usr/lib/libnvomxilclient.so	OpenMAX IL client.
./usr/lib/libnvomx.so	OpenMAX IL implementation.
./usr/lib/libnvos.so	NVIDIA OS abstraction library.
/usr/lib/libnvparser.so	Parser used for NVIDIA NvMMlite.
./usr/lib/libnvrm_graphics.so	Resource Manager (NvRM) graphics host, AVP communication library, and graphics drivers.
./usr/lib/libnvrm.so	Resource Manager kernel interface.
./usr/lib/libnvsm.so	NVIDIA shader manager library.
./usr/lib/libnvtestio.so	Target (ARM) shared object for test I/O utilities.
./usr/lib/libnvtestresults.so	Test results shared object.
./usr/lib/libnvtvmr.so	Multimedia Tegra video mixer/renderer.
./usr/lib/libnvwinsys.so	Winsys library.
./usr/lib/libnvwsi.so	NVIDIA windowing system integration for

	EGL.
./usr/lib/xorg	-
./usr/lib/xorg/modules	-
./usr/lib/xorg/modules/drivers	-
./usr/lib/xorg/modules/drivers/tegra_drv.abi*.so	Tegra X ABI drivers.

Getting Started

This section contains information to help you get started using this release of NVIDIA® Tegra® Linux Driver Package. It covers the following topics:

- Requirements
- Boot Options
- Setting Up Your Environment
- Setting Up Your File System
- Configuring NFS Root on the Linux Host
- Setting Power Saving Options
- Flashing the Boot Loader and Kernel
- Synchronizing the Kernel Sources
- Building the NVIDIA Kernel
- OpenGL/EGL Gears Test Application
- GStreamer-based Multimedia Playback
- GStreamer-based Camera Capture

Requirements

The following lists the requirements to use this Tegra Linux Driver Package (L4T) release:

- Host PC running Linux. Ubuntu 10.04 is used in examples in this document, but other distributions should also work.
- A kernel image (zImage). L4T does contain a kernel image (zImage), and you can also download and rebuild from source.
- Boot loader. This can be Fastboot or U-Boot. Flashing on Tegra 3 series (code-named Cardhu) and Tegra 2 series (code-named Ventana) developer boards requires a boot loader. This boot loader can be either the Fastboot utility or U-Boot, both of which are included in this release.
- A rootfs device which can be an SD card, a USB hard disk, or USB stick formatted to EXT3. It is also possible to use the target device's internal memory, or your Linux Host PC hard-drive through NFS.
- (Cardhu) A USB Micro-B plug to USB Std A female cable to plug into the board's recovery port.

- (Ventana) A USB Micro-B plug to Std A female cable to plug into the board's recovery port [J10].

  Note: The NVIDIA binaries provided for Tegra 2 devices code-named "Ventana" may be able to support the Tegra 2 devices code-named "Harmony".

Boot Options

It is currently possible to boot L4T on the Tegra 2 series Ventana developer board, as well as Tegra 3 series Cardhu with a root file system from:

- USB stick or USB hard disk
- SD card
- Internal eMMC
- Network file system (NFS)

Setting Up Your Environment

The following subsections contain information to help you get started using this pre-release of L4T. They cover the following topics:

- Extracting Tegra Linux Driver Package
- Setting Up Your Board

Extracting Tegra Linux Driver Package

Note: The procedures in this document assume you extract the release package in ~/.

To extract Tegra Linux Driver Package

- Extract the package manually by executing the following command:

  ```
  $ sudo tar -vxjf <platform>_Tegra-Linux-
  <release_num>.<version_num>_<release_type>.tbz2
  ```

  Where:

  - <release_num> is the branch number of the release, such as R16.
  - <version_num> is the revision number of the build such as 1.0 for the first build.
  - <release_type> is `armel` (for softfp ABI) or `armhf` (for hardfp ABI).

Setting Up Your Board

L4T requires a Tegra 2 series developer board (Ventana), or a Tegra 3 series (Cardhu) as well as a host PC running Linux. Please consult your board documentation for steps on how to setup and configure your board.

Prerequisites

- You have a device specified above in the "Boot Options" topic (formatted to EXT3). (It can also be a memory card with a USB adapter.)
- You have a micro-B USB male-to-USB Std A female cable to plug into the board's recovery port.

Setting Up Your File System

This section describes the steps for setting up your file system. You must set up the root file system and copy the file system to your boot device.

About the Root File System

The provided sample root file system was created with Rootstock 0.1.99.4 using the following command:

```
$ sudo rootstock --fqdn tegra-ubuntu --login ubuntu --password
ubuntu --imagesize 1G -d precise --seed ubuntu-
minimal,xserver-xorg-core,xinit,xterm,alsa-utils,wireless-
tools,wpasupplicant,x11-xserver-utils,openssh-client,openssh-
server
```

This creates a file system with the hostname `tegra-ubuntu`, the username `ubuntu`, and the password `ubuntu`.

> **Note**: The provided sample target file system does not come with pre-generated ssh host keys. These host keys can be re-generated with the following command:
>
> ```
> $ ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key
> ```
>
> View the `ssh-keygen` man page for other `-t` options.

We have also made the following changes:

- Modified: `/etc/{init/ssh.conf, X11/Xwrapper.conf, resolve.conf}`
- Deleted: `/etc/ssh/ssh_host_*`

To replicate these changes in your own rootstock file-system, copy or make similar modifications from those files to your own rootstock file system. Please also view the files included in `~/Linux_for_Tegra/nv_tegra/config.tbz2 file` and make appropriate adjustments as made in those files for your root filesystem.

The following packages are installed by default:

- ubuntu-minimal
- xserver-xorg-core
- x11-xserver-utils
- xinit
- xterm
- alsa-utils
- wireless-tools
- wpasupplicant
- opens-client
- openssh-server

Setting Up the Root File System

The next step in booting the target board is to configure the root file system. Follow the procedures in this section to set up the rootfs and to copy the file system to the rootfs device.

> **Note**: The instructions below use the sample file system that is provided by NVIDIA as the base. If you would like to use your own, set the LDK_ROOTFS_DIR environment variable to point to where your rootfs is located and skip Steps 1 and 2.

To set up the rootfs

1. Download the following file to your home directory:

   `Tegra-Linux-Sample-Root-Filesystem_<release_type>.tbz2`

   where <release_type> is `armel` (for softfp ABI) or `armhf` (for hardfp ABI).

   This file contains the NVIDIA-provided sample root file system.

2. Extract the compressed file as follows:

- Navigate to the rootfs directory of the extracted NVIDIA driver package with this command:

  ```
  $ cd <your_L4T_root>/Linux_for_Tegra/rootfs
  ```

  where `<your_L4T_root>` is your L4T root directory, which is assumed to be your home directory (~).

  For more information, see Extracting Tegra Linux Driver Package in this section.

- Extract the sample file system to the rootfs directory with this command:

  ```
  $ sudo tar zxpf ../../Tegra-Linux-Sample-Root-
  Filesystem_<release_type>.tbz2
  ```

3. Run the `apply_binaries.sh` script to copy the NVIDIA user space libraries into the target file system:

   ```
   $ cd ..
   $ sudo ./apply_binaries.sh
   ```

   Note: The `apply_binaries.sh` installs the appropriate default version, for the provided sample file system, of the Tegra X driver by creating a sym-link to point to the proper ABI. The script installs the sym-link to the following location in the rootfs:

   ```
   ./usr/lib/xorg/modules/drivers/tegra_drv.so
   ```

   If you are using a different rootfs, or if you already have configured your rootfs, you can apply the NVIDIA user space libraries by setting the LDK_ROOTFS_DIR environment variable to point to your rootfs. Then run the script, as shown above, to copy the binaries into your target file system.

   To install a different ABI:

   - Use the `apply_binaries.sh` script and pass the `--xabi <number>` flag by executing:

     ```
     $ sudo ./apply_binaries.sh --xabi <number>
     ```

     where <number> is the ABI version to install, either 5, 6, 7, 8, 10, 11, or 12. The ABI default value for the X driver is 'X ABI version 11' and it is the version that is compatible with the provided sample file system. You can find the version to use in the sample file system by following the directions listed below.

   To determine the X driver ABI of the X server used in the root file system:

   - Start X once on the Tegra device.
   - Examine the resulting file `/var/log/Xorg.0.log`, which will contain something like the following:

     ```
     (II) Module ABI versions:
     X.Org ANSI C Emulation: 0.4
     X.Org Video Driver: 8.0
     ```

The X.Org Video Driver line reports the ABI version. The sample Ubuntu 12.04-based root file system uses X driver ABI 11.

If the `apply_binaries.sh` script installs the binaries correctly, the last message output from the script is “Success!”.

4. Optionally load any additional packages as shown in the Installing Additional Packages topic in this guide.
5. Load the target file system that you have generated onto the first partition of a device (either a USB stick, an SD card, or a USB hard drive) and attach that device to the target board. Alternatively, you can use the `flash.sh` script to flash the root filesystem to the internal eMMC. In this case proceed with the following steps, and then and follow the internal eMMC instructions.
6. Follow the steps in the Flashing the Boot Loader and Kernel section of this guide.
7. Power on the target board.
8. Set up access to the board over a serial port by opening a terminal on the host PC and setting:
   - 115200 baud
   - 8-bit
   - Parity none
   - 1 stop bit

To copy the file system to the external rootfs device

1. Plug your rootfs device into the host PC.
2. If your device is not formatted as Ext3, enter the following command to format it with an Ext3 file system:

   ```
   $ sudo mkfs.ext3 /dev/sd<port><device number>
   ```

   where <port> is the port where your device is mounted and `<device_number>` is the device number of the device attached to the port. You can use the `dmesg` command to determine the port.
3. If needed, mount your device with the following command:

   ```
   $ sudo mount /dev/sdX1 <mntpoint>
   ```

   where <mntpoint> is your rootfs device’s mount point on the host system.
4. Copy the file system. If LDK_ROOTFS_DIR is set, execute these commands:

   ```
   $ cd ${LDK_ROOTFS_DIR}
   $ sudo tar -cpf - * | ( cd <mntpoint> ; sudo tar -xpf - )
   ```

   If it is not set, copy the rootfs directory that is included in the release by executing the following commands:

```
$ cd <your_L4T_root>/rootfs
$ sudo tar -cpf - * | ( cd <mntpoint> ; sudo tar -xpf - )
```

Once you have flashed your board, you can then unmount the disk and plug it to the board. For more information about flashing, see the Flashing the Boot Loader and Kernel topic in this section. For information about configuring your board setup, see the hardware documentation for your developer board.

To copy the file system to the external rootfs device

- See the Flashing the Boot Loader and Kernel topic in this section, for flashing to internal eMMC.

Updating Drivers on an Existing Target System

These instructions are for the situation where there was a previous release or driver package loaded onto a target board and that target device is booted.

Prerequisite

You must attach an Ethernet cable to the device through either the Ethernet port (if available) or through a USB Ethernet adaptor. Alternatively, you can connect through Wi-Fi if the appropriate driver and firmware are enabled and installed.

To update drivers on an existing target system

1. Log into the target device.
2. Download the NVIDIA Tegra Linux driver release and the additional support packages (for example the codec's package) from the `nvidia.com` links with `wget`. For example, to download the last (not the latest) release do the following:

   ```
   wget
   http://developer.nvidia.com/sites/default/files/akamai/tools/f
   iles/l4t/<last_rel_num_rNN>/cardhu_Tegra-Linux-
   <last_rel_num_RNN>_<release_type>.tbz2
   wget
   http://developer.nvidia.com/sites/default/files/akamai/tools/f
   iles/l4t/<last_rel_num_rNN>/cardhu_Tegra-Linux-codecs-
   <last_rel_num_RNN>_<release_type>.tbz2
   ```

   Where:

   - <last_rel_num_rNN> is the number of the last release in this format (rNN), such as r15.
   - <last_rel_num_RNN> is the number of the last release in this format (RNN), such as R15.
   - <release_type> is `armel` (for softfp ABI) or `armhf` (for hardfp ABI).

Note: The release shown in this example is a previous release and not the current release.

3. Extract the release. For more information, see Extracting Tegra Linux Driver Package in this section.
4. Set the LDK_ROOTFS_DIR variable to point to the root '/' directory.

   ```
   $ export LDK_ROOTFS_DIR=/
   $ echo ${LDK_ROOTFS_DIR}
   ```

5. Go into the Linux_for_Tegra directory.

   ```
   $ cd Linux_for_Tegra
   ```

6. Run the `apply_binaries.sh` script to install the NVIDIA drivers onto your target board. For more information, see Setting Up the Root File System in this section.
7. (Optional) Change your X driver ABI as a variable passed into apply_binaries. For more information, see Setting Up the Root File System in this section.
8. For the codecs, or other additional packages, extract the files, being sure to extract them to your root '/' directory. For more information, see Installing Additional Packages in this section.

Determining the Success of a Driver Update

You can determine whether a driver update on a target board went successfully.

To determine the success of a driver update

- Execute the following command on a booted target device:

  ```
  $ cat /etc/nv_tegra_release | grep -v -P '^#.*$' | sha1sum -c
  ```

Increasing the Internal Memory Partition Size for the Root File System

The suggested rootfs partition size for Ventana and Cardhu is 1073741824 bytes and is specified by default in the `flash.sh` script. This 1 GB reserved in internal memory for the rootfs partition may be insufficient for installation of additional packages. Packages such as ubuntu-desktop, gst, and other gnome plug-ins may require additional space.

The “`-S <size-in-bytes>`” argument to `flash.sh` can be used to change the partition size.

To flash for a larger partition

- Execute the following command:

  ```
  $ sudo ./flash <target_device> -S <size_in_bytes> <rootdev>
  ```

  Where:

  - `<target_device>` is either cardhu or ventana.
  - `<size_in_bytes>` is the desired size for the partition.
  - `<rootdev>` is the rootfs partition's internal memory, for example mmcblk0p1.

Installing Additional Packages

This section explains how to install the additional NVIDIA packages, additional Ubuntu packages, and Google Chrome.

Installing Additional NVIDIA Packages

Additional NVIDIA packages may be posted alongside the release. To make full use of the features in the release, these additional packages must be installed. These may include the following packages:

- `restricted_codecs.tbz2`, which is included in the `<target_board>_Tegra-Linux-codecs-<version>_<release_type>.tbz2`
- `wifi.tbz2` file

Directly after the "apply_binaries" step in Setting Up the Root File System, you can install the package into the configured rootfs.

To install an NVIDIA package when the rootfs is already installed on the device

1. Mount the target rootfs device to `/mnt`:

   ```
   $ sudo mount <device> /mnt
   ```

   where `<device>` is the device such as `/dev/sda1`.

2. Install the package:

   ```
   $ tar -C /mnt -xjpf <path-to>/$<package_name>
   ```

   where `<package_name>` is one of the packages.

   For example, if you have the `restricted_codecs.tbz2` file located in:

   ```
   <home>/restricted_codecs.tbz2
   ```

   then your line will look like:

```
$ tar -C /mnt -xjpf <home>/restricted_codecs.tbz2
```

3. Unmount the device with this command:

   ```
   $ sudo umount <device>
   ```

   and attach the device to the target board.

   If the rootfs is installed on the device's internal eMMC, use the `scp` command to copy the `restricted_codecs.tbz2` package to the target board, and then extract the `restricted_codecs.tbz2` package from the root directory.

Installing Additional Ubuntu Packages

This section explains how to install additional packages from Ubuntu by using the provided sample file-system. For example, you may wish to download the following packages:

- `openssh-server` for remotely logging in
- `ubuntu-desktop` for the standard Ubuntu graphical user interface

You can receive notifications from Update Manager when new Ubuntu packages are available.

> **Note:** L4T is tested with base Ubuntu packages only. No updated packages have been tested.

To receive notifications

1. Locate and edit the following file:

   ```
   /etc/apt/sources.list
   ```

2. Add the following line:

   ```
   deb http://ports.ubuntu.com/ubuntu-ports <distribution>-
   updates main universe
   ```

   Where `<distribution>` is the name of the Ubuntu distribution your rootfs is based on. For example, for a rootfs based on the Natty Narwhal distribution of Ubuntu, add the line:

   ```
   deb http://ports.ubuntu.com/ubuntu-ports natty-updates main
   universe
   ```

Prerequisite

You have attached an Ethernet cable to the device through either the Ethernet port (if available) or through the USB Ethernet adaptor, or connect through Wi-Fi if the appropriate driver and firmware are enabled and installed.

To install more packages

1. Boot the target device.
2. Turn on networking by executing:

   ```
   $ sudo dhclient
   ```

   Note: You may need to specify eth0/eth1 and other parameters to assign an IP address to the appropriate interface.
3. Install packages using `apt-get`. For example, to install `wget` execute this command:

   ```
   $ sudo apt-get install wget
   ```

Installing Google Chrome

You can install Google Chrome from the command line, without using a browser.

Note: These installation procedures are untested.

To install Google Chrome from the command line

1. Download and install a Linux-repository public key from Google:

   ```
   $ wget -q -O - https://dl-
   ssl.google.com/linux/linux_signing_key.pub | sudo apt-key add
   -
   ```
2. Add the key to the repository:

   ```
   $ sudo add-apt-repository 'deb
   http://dl.google.com/linux/chrome/deb/ stable main'
   ```
3. Update the repository and install the browser:

   ```
   $ sudo apt-get update
   ```
4. Install the Google Chrome Stable version with:

   ```
   $ sudo apt-get install google-chrome-stable
   ```

   Or install the beta version (latest version) with:

   ```
   $ sudo apt-get install google-chrome-beta
   ```

   For instructions on installing Google Chrome, see:

   http://www.ubuntuupdates.org/ppa/google_chrome

Configuring NFS Root on the Linux Host

To boot the target device from NFS, you must provide an NFS root mount point on your Linux host machine. The procedure in this section describes the basic steps to do so.

Prerequisites

- You must have Ethernet connection to install packages on the host.
- You must have an Ethernet connection on the target, as well.

To configure NFS root on the Linux host

1. Install the `nfs` components on your host machine:

   ```
   $ sudo apt-get install nfs-common nfs-kernel-server
   ```

2. The NFS server needs to know which directories you want to 'export' for clients. This information is specified in the `/etc/exports` file.

   - Modify `/etc/exports` to look somewhat like this:

     ```
     $ /nfsroot
     *(rw,nohide,insecure,no_subtree_check,async,no_root_squash)
     ```

   - After adding the entry, restart using the following command:

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

3. Create an `/nfsroot` directory on your Linux host machine:

   ```
   $ sudo mkdir /nfsroot
   ```

4. Copy the file system to the `nfsroot` directory:

   ```
   $ cd ./rootfs
   $ sudo tar -cpf - * | ( cd /nfsroot ; sudo tar -xpf - )
   ```

5. Export the root point:

   ```
   $ sudo exportfs -a
   ```

   Alternatively, you can export or un-export all directories by using the `-a` and `-u` flags. The following command un-exports all directories:

   ```
   $ sudo exportfs -au
   ```

6. (Optional) If the Ubuntu firewall blocks NFS root access, it must be disabled depending upon your configuration. You can do so with the following command:

   ```
   $ sudo ufw disable
   ```

7. If there are issues performing the NFS boot, to separately verify everything on the ‘host’ machine is configured properly, you can perform the following

step on a booted target board through USB/SD/internal eMMC. It should be possible to mount the host NFS root point on the target device:

```
$ mkdir rootfs
$ sudo mount -v -o nfsvers=3 <IP-ADDR>:/nfsroot rootfs
```

Where `<IP-ADDR>` is the IP address of the Linux Host machine as taken from the "ifconfig" command. This proves that the host configuration is correct.

Note: You must install `the nfs-common` package using the following command prior to executing the mount command on the target machine:

```
$ sudo apt-get install nfs-common
```

To boot the target with the NFS root point, see the Flashing the Boot Loader and Kernel topic in this section and be sure to include the `-N` option for the nfs root point.

Setting Power Saving Options

This section explains how to enable the hotplug driver and the Tegra CPU power-gated state (LP2) for power savings on the target board.

Enabling the Auto-Hotplug Driver

The auto-hotplug driver implements the policy for when to bring cores online/offline. The auto-hotplug driver also implements the policy for when to switch clusters, i.e. when to switch from companion CPU to main CPU or vice versa. Cluster switching is transparent to the OS. The switch happens when software enters a power-gated state on one CPU core and hardware resumes the execution on a different physical CPU core.

To enable the auto-hotplug driver

- Enter the following command:

```
$ echo 1 > /sys/module/cpu_tegra3/parameters/auto_hotplug
```

Enabling the Tegra CPU Power-Gated State (LP2)

With the LP2 power state, the CPU core is power-gated if supported by the hardware. If all CPU cores on the VDD_CPU power rail are in LP2, Tegra hardware signals the PMIC to turn off the regulator.

To enable the LP2 power state

- Enter the following command:

```
$ echo Y > /sys/module/cpuidle/parameters/lp2_in_idle
```

Flashing the Boot Loader and Kernel

This section describes the steps that must be taken to boot the target board by flashing the kernel and boot loader (code-name Ventana or Cardhu platforms) and provides usage information for the `flash.sh` helper script.

Flash Procedure

The first step is to flash the board with the boot loader and kernel, and, optionally, flash the rootfs to internal eMMC.

Prerequisites

The following directories must be present:

- /bootloader—boot loader plus flashing tools (NvFlash, CFG, BCTs, etc.)
- /kernel—a kernel zImage plus scripts to sync/build the kernel
- /rootfs—the root file system that you download
- /nv_tegra—NVIDIA® Tegra® user space binaries

You must also have the USB cable connected to the recovery port prior to running the commands listed below. For more information, see the Requirements topic in this section.

To flash the boot loader and kernel

1. Put the target board into reset/recovery mode. Do so by first powering on the board and then holding the recovery button, and then pressing the reset button as described in the Quick Start guide for the board.
2. Run the `flash.sh` script that is in the top level directory of this release. The script must be supplied with the target board code name (`cardhu` or `ventana`) for the root file system:

   ```
   $ sudo ./flash.sh <target_board> <rootdev>
   ```

   - If the root file system will be on a USB disk, execute the script as follows:

     ```
     $ sudo ./flash.sh <target_board> sda1
     ```

   - If the root file system will be on an SD card, execute the script as follows:

     ```
     $ sudo ./flash.sh <target_board> mmcblk1p1
     ```

   - If the root file system will be on the internal eMMC, execute the script as follows:

```
$ sudo ./flash.sh <target_board> mmcblk0p1
```

Where `<target_board>` is `ventana` or `cardhu`, depending upon your target device.

The above examples are for fastboot. For U-Boot, add the following argument:

```
-L <PATH_TO_U-BOOT_BIN_FILE>
```

For example:

```
$ sudo ./flash.sh –L bootloader/<target_board>/u-boot.bin
<target_board> <rootdev>
```

The boot loader and kernel will load.

For more information on U-Boot see the U-Boot Guide chapter of this document.

Flash Script Usage

You can find the most up-to-date usage information by running `flash.sh –h` (using the `flash.sh` script included in the release). The basic usage information is as follows.

Usage

```
sudo ./flash.sh [options] <target_board> <rootdev>
```

Where you specify the required parameters and one or more of the options shown in the following table.

Parameters	Description	
<target_board>	Is one of ventana or cardhu.	
<rootdev>	Is one of following:	
	For Ventana	
	mmcblk0p1	Specifies internal eMMC.
	mmcblk1p1	Specifies external SDCARD.
	sda1	Specifies external USB device (such as, USB memory stick or HDD).
	eth0	Specifies nfsroot via external USB Ethernet interface.
	For Cardhu	
	mmcblk0p1	Specifies internal eMMC.
	mmcblk1p1	Specifies external SDCARD.
	sda1	Specifies external USB device (such as, USB memory stick or HDD).

	eth0	Specifies nfsroot via external USB Ethernet interface.

Options	Description
-h	Specifies to print this usage information.
-b <bctfile>	Specifies the NvFlash Boot Configuration Table (BCT) file.
-c <cfgfile>	Specifies the NvFlash configuration file.
-f <flashapp>	Path to flash application: nvflash or tegra-rcm.
-k <partition id>	Specifies the kernel partition ID to be updated (minimum = 5).
-n <nfs args>	Specifies the static NFS network assignments: <Client IP>:<Server IP>:<Gateway IP>:<Netmask>
-o <odmdata>	Specifies the ODM data value: 0x30098011 (ventana) 0x80080105 (Cardhu A01) 0x40080105 (Cardhu A02)
-r	Skip building and reuse existing `system.ing`
-s <ubootscript>	HUSH bootscript file for U-Boot.
-F <flasher>	Flash server such as fastboot.bin.
-L <bootloader>	Specifies the boot loader, such as fastboot.bin.
-C <cmdline>	Specifies the kernel command line. **Warning**: Each option in this kernel command-line gets higher precedence over the same option from fastboot. In case of NFS booting, this script adds NFS booting related arguments, if the -i option is omitted.
-i	Specifies the kernel command line as-provided by the user and passes it to fastboot.
-D <boot Device>	Specifies eMMC or NAND.
-K <kernel>	Specifies the kernel image, such as zImage.
-I <initrd>	Specifies `initrd` file. Null `initrd` is the default.
-R <rootfs dir>	Specifies the sample rootfs directory.
-N <nfsroot>	Specifies the nfsroot, for example: `<my IP addr>:/my/exported/nfs/rootfs`
-S <size>	Specifies the rootfs size in bytes. This is valid only for internal rootdev.

Synchronizing the Kernel Sources

You can manually rebuild the kernel used for this package. Internet access is required to do so.

Prerequisites

- You have installed Git. This can be done on Ubuntu 10.04 by running the following command:

  ```
  $ sudo apt-get install git-core
  ```

- Your system has the default Git port 9418 open for outbound connections.

To rebuild the kernel

1. Get the kernel source by running the `source_sync.sh` script:

   ```
   $ ./source_sync.sh -k
   ```

   Which will prompt you to enter a 'tag' name, which is provided in the release notes. Alternatively, you can provide the 'tag' name when calling the script. For example:

   ```
   $ ./source_sync.sh –k <TAG_NAME>
   ```

   —Or—

   You can also manually sync the sources, as follows:

   ```
   $ cd <myworkspace>
   $ git clone git://nv-tegra.nvidia.com/linux-2.6.git
   kernel_sources
   $ cd kernel_sources
   $ git checkout <TAG_NAME>
   ```

   where `<TAG_NAME>` is the 'tag' name that is available in the release notes.

   You can sync to any Linux tag you would like, but the tag provided in the release notes will sync the sources to the same source point of time the release binary was built from. To see a list of the available release tags, use:

   ```
   $ git tag –l 'tegra-l4t*
   ```

Building the NVIDIA Kernel

Follow the steps in this procedure to build the NVIDIA kernel.

Prerequisites

- You have downloaded the kernel source code.

To build the Tegra Kernel

1. Export the following environment variables:

   ```
   $ export CROSS_COMPILE=<crossbin>
   $ export TEGRA_KERNEL_OUT=<outdir>
   ```

```
$ export ARCH=arm
```

where:

- `<crossbin>` is the prefix applied to form the path to the tool chain for cross compilation, e.g., `gcc`. For a CodeSourcery tool chain, it will look something like:

  ```
  <csinstall>/arm-2009q1-203-arm-none-linux-gnueabi/bin/arm-
  none-linux-gnueabi-
  ```

- `<outdir>` is the desired destination for the compiled kernel.

2. Execute the following commands to create the `.config`:

   ```
   $ cd <myworkspace>/kernel
   $ mkdir $TEGRA_KERNEL_OUT
   ```

   - For Tegra 3, Cardhu, use:

     ```
     $ make O=$TEGRA_KERNEL_OUT tegra3_defconfig
     ```

   - For Tegra 2 Ventana use:

     ```
     $ make O=$TEGRA_KERNEL_OUT tegra_defconfig
     ```

   Where `<myworkspace>` is the parent of the Git root.

3. Execute the following commands to build the kernel:

   ```
   $ make O=$TEGRA_KERNEL_OUT zImage
   ```

4. Execute the following commands to build the kernel modules (and optionally install them)

   ```
   $ make modules DESTDIR=<your_destination>
   $ make modules_install INSTALL_MOD_PATH=<your_destination>
   ```

5. Copy the kernel zImage over the one present in the 'kernel' directory of the release.

6. Archive the kernel modules created in step 4 using the `tar` command and the filename that is used for the kernel modules tar file in the same kernel directory of the release. When both of those tar files are present, you can follow the instructions provided in this document to flash and load your newly built kernel.

OpenGL/EGL Gears Test Application

If you would like to run a sample OpenGL/EGL test application, you can run the open-source Gears application.

To install and run Gears test application

1. Boot the target system with Ethernet connection and install the `mesa-utils-extra` package.

   ```
   $ sudo apt-get install mesa-utils-extra
   ```

2. Manually create the sym-links in the target root file-system for `/usr/lib/libEGL.so.1` and `/usr/lib/libGLESv2.so.2`.

   ```
   $ mv /usr/lib/libEGL.so /usr/lib/libEGL.so.1
   $ mv /usr/lib/libGLESv2.so /usr/lib/libGLESv2.so.2
   $ ln -s /usr/lib/libEGL.so.1 /usr/lib/libEGL.so
   $ ln -s /usr/lib/libGLESv2.so.2 /usr/lib/libGLESv2.so
   ```

3. At this point you should be able to run the application with the following steps:

   ```
   $ export DISPLAY=:0
   $ X&
   $ /usr/bin/es2gears
   ```

GStreamer-based Multimedia Playback (NvGstPlayer)

You can use the GStreamer open source multimedia framework and the NvGstPlayer utility for testing multimedia local playback and HTTP/RTSP streaming playback use cases. The NvGstPlayer can be used as a reference implementation.

This section tells you how to install and use this application. This section includes the following sub-topics.

- Installing GStreamer
- Using NvGstPlayer

For more information about the NvGstPlayer application, refer to the readme file included with the release.

Installing GStreamer

You install GStreamer from the Internet directly on the target. There is a wrapper library called `gst-openmax` that is an interface between GStreamer and OpenMAX, which enables accelerated NVIDIA plug-ins in the GStreamer framework

For more information about GStreamer, see the following website:

`http://gstreamer.freedesktop.org`

NvGstPlayer is a multimedia player test application.

Complete prerequisite steps 1 through 4 from the file `nvgstcapture_README.txt` before running the NvGstPlayer and NvGstCapture applications.

Instructions for installing GStreamer are also included in that text file.

Using NvGstPlayer

NvGstPlayer is a command line media file player. It will play audio/video files encapsulated in MP4, 3GP, AVI, ASF, WMA, MKV, M2TS, WEBM, and MOV. NvGstPlayer supports local file playback and playback over RSTP, HTTP, and UDP. For information about NvGstPlayer runtime commands, default settings, and important notes see the `nvgstplayer_README.txt` file included in the release.

Note: To use the NvGstPlayer application, you must install the restricted codecs available (licensed seperately) on the release website. For more information, see Installing Additional NVIDIA Packages in this guide.

Gstreamer-based Camera Capture (NvGstCapture)

The NvGstCapture application supports GStreamer version 0.10.32 by default. NvGstCapture can capture audio and video data using microphone and camera and encapsulate encoded A/V data in the container file.

To use the NvGstCapture application you must install the restricted codecs package available (licensed separately) on the release website. For more information, see Installing Additional NVIDIA Packages in this guide.

For NvGstCapture installation and usage information, see the nvgstcapture_README.txt file included with the release.

U-Boot Guide

This document describes the U-Boot implementation for NVIDIA® Tegra® Linux Driver Package. It covers the following topics:

Requirements

This topic provides software requirements and prerequisites, including Linux tools that are required for Tegra Linux Driver Package (L4T).

- Linux-based Host System.

  Functionality of the u-boot build and flashing utilities was validated using an Ubuntu 10.04 host system; however, later versions or alternative Linux distributions may work with host-specific modifications.

- Tegra Linux Driver Package (L4T).

  Download the latest L4T package from the Tegra Developer Zone and follow the installation instructions in the user documentation. You can find L4T on the Tegra Developer Zone:

  `http://developer.nvidia.com/linux-tegra`

- Flex and Bison.

  The U-Boot makefiles require flex and bison to parse various configuration files. If flex and bison are not already installed on your host machine, you can install them on an Ubuntu host machine with the following command:

```
$ sudo apt-get install flex bison
```

- Device Tree Compiler (dtc).

  Ensure that the full path to the dtc binary is available to the U-Boot make system by either passing the path as a variable or by making the dtc directory available in the local command path of the host machine. Most of the dtc packages available from standard Linux distribution package management systems (like `apt`) are not yet updated with a version of dtc with the features required by the U-Boot makefile. Therefore, an example of building dtc from source is included in this section. For the procedure, see the Using Device Tree Compiler topic in this section.

- U-Boot Kernel Image Tool (mkimage).

  The U-Boot image tool is what generates the `vmlinux.uimg` from the kernel zImage. The L4T release includes a prebuilt `mkimage` in <your_L4T_root>/Linux for Tegra/bootloader. If `mkimage` is not installed, it can be obtained by building the top-level U-Boot make target. For more information on building U-Boot, see Downloading and Building U-Boot in this section. The `mkimage` tool can then be found in the U-Boot source directory at:

  ```
  ./tools/mkimage
  ```

- ARM tool chain for cross compilation.

  For more information, see the Toolchain section in this guide.

- U-Boot source.

  For more information, see the Downloading and Building U-Boot topic in this section.

- Kernel source.

  For information, see the following sections in the Getting Started chapter:

  - Setting Up Your Environment
  - Synchronizing the Kernel Sources
  - Building the NVIDIA Kernel

  Also, see the Adding a Compiled Kernel to the Root File System topic in this section.

Using Device Tree Compiler

This topic provides an example of building the Device Tree Compiler (dtc) from source to include the features required by the U-Boot makefile.

To build dtc from source

1. If you do not want to pass in dtc as a parameter to the U-Boot environment, ensure a local command path (such as `./usr/local/bin` or another choice) is at the beginning of the shell command path.

   ```
   $ export PATH=<local_command_path>:${PATH}
   ```

   Note: If you execute:

   ```
   $ make install
   ```

   The dtc makefile installs the binary into the first entry of shell PATH variable, so it is important that the local command path is at the beginning of the shell PATH variable.

2. Create a directory to contain the dtc source code and change directories into it:

   ```
   $ mkdir -p <dtc_src_dir>
   $ cd <dtc_src_dir>
   ```

3. Download dtc source code by executing the following `git clone` command:

   ```
   $ git clone http://git.jdl.com/software/dtc.git
   ```

   Note: At time of publication, the tested commit ID for dtc is:

   `a6e6c60e3a97a6b3a033cd052bb3740fd53cbf4c`

4. Build and optionally install dtc by executing:

   ```
   $ cd <dtc_src_dir>/dtc
   $ make
   ```

   Or, alternatively, if you want it installed on your local host file system execute:

   ```
   $ make install
   ```

   Note: if you specified just `make` be sure to pass in the following to the U-Boot make system:

   `DTC=${PATH_TO_DTC_TOOL_BINARY}`

   Where PATH_TO_DTC_TOOL_BINARY is the location of the dtc binary, such as:

   `<dtc_src_dir>/dtc/dtc`

Downloading and Building U-Boot

This topic provides the steps to follow when downloading and building U-Boot to use as a boot loader for the Tegra device.

To download and build U-Boot

1. Download the L4T U-Boot source code by executing the following commands:

   ```
   $ mkdir -p <uboot_src_dir>
   $ cd <uboot_src_dir>
   $ git clone -n git://nv-tegra.nvidia.com/3rdparty/u-boot.git
   ```

   Alternatively, you can use the `source_sync.sh` script that is provided in the L4T release and skip Step 2 below.

   When running `source_sync.sh -u`, if no parameters are provided, the script prompts for the `<TAG_NAME>`, which is provided in the release notes. Also, you can run the script by passing the `<TAG_NAME>` in as follows:

   ```
   $ cd <your_L4T_root>/Linux_for_Tegra
   $ ./source_sync.sh –u <TAG_NAME>
   ```

   This will sync the source to `<source_sync.sh_location>/sources/u-boot_source`. The `<uboot_src_dir>` directory becomes `<your_L4T_root>/Linux_for_Tegra/sources/u-boot_source`. Use that path below in Step 4.

2. Check out the git tag name:

   ```
   $ cd u-boot
   $ git checkout -b mybranchname <tag_name>
   ```

   where `<tag_name>` is provided in the Release Notes.

3. Set the build environment:

   ```
   $ export ARCH=arm
   $ export CROSS_COMPILE=<your_toolchain_location>
   $ export CONFIG_L4T=1
   $ export USE_PRIVATE_LIBGCC=yes
   $ export DTC=<dtc_binary_location>
   ```

4. Build U-Boot by executing:

   ```
   $ cd <uboot_src_dir>/u-boot
   $ make distclean
   $ make <platform>_config
   $ make
   ```

   Where <platform> is the device, such as code-name cardhu for Tegra 3 or code-name ventana for Tegra 2.

5. Copy U-Boot for flashing to the device:

   ```
   $ cp u-boot.bin
   <your_L4T_root>/Linux_for_Tegra/bootloader/<platform>
   ```

   Note: Before copying U-Boot, it is recommended that you back up the original `u-boot.bin` file in this directory.

Adding a Compiled Kernel to the Root File System

This topic describes the steps to create and install the kernel image required by U-Boot into the sample file system.

Prerequisites

- You have compiled the kernel as described in Getting Started in this guide.

To configure a file system for U-Boot

- Use `the zimage_to_uimg.sh` script to create the `vmlinux.uimg` from your kernel zImage by executing the following commands:

  ```
  $ cd <your_L4T_root>/Linux_for_Tegra
  $ ./zImage_to_uimg.sh –z <path>/zImage
  ```

  With default values, `zImage_to_uimg.sh` writes `vmlinux.uimg` to the kernel directory of the extracted L4T release. Run `zImage_to_uimg.sh –h` for description of other parameters.

  The `zimage_to_uimg.sh` script runs `mkimage` with the appropriate parameters to create the `vmlinux.uimg` file in the kernel directory. The apply_binaries script copies the `vmlinux.uimg` in the kernel directory into the `rootfs` directory in the `/boot` folder. You can then install the `rootfs` directory onto your device. For U-Boot to function properly, there must be a `vmlinux.uimg` kernel image in the /boot directory of the target filesystem. For information on installing the `rootfs` directory onto your device, see Setting Up the Root File System in the Getting Started chapter.

  If you have already installed your rootfs onto a device, you can manually copy the `vmlinux.uimg` file created by the `zImage_to_uimg.sh` script to the previously installed root file system.

To configure a network file system for U-Boot

Ensure that a tftp daemon is installed on the host PC. This tftp daemon installation needs to be done once only.

1. To install the tftp daemon, execute the following command:

   ```
   $ sudo apt-get install tftpd
   ```

2. Locate and edit /etc/inetd.conf and ensure the following line is contained in the file:

   ```
   tftp dgram udp wait nobody /usr/sbin/tcpd/usr/sbin/in.tftpd
   /tftpboot
   ```

3. Execute the following commands:

   ```
   $ cd /etc/init.d;
   $ ./openbsd-inetd restart
   ```

   Note: Any tftp daemon works, but the TFTP access point must be configured as the `/tftpboot` directory of the host machine for the U-Boot on the target system to perform the tftp boot.

- For a network RootFS, use the -d option to specify the path to the host machine's `tftp_`access directory. It will look as follows:

  ```
  $ ./zImage_to_uimg.sh -z <path to zImage>/zImage –d /tftpboot
  ```

  With the vmlinux.uimg file updated in the /tftpboot directory, use the `rootfs` built and configured for NFS boot as described in the Getting Started chapter.

To configure a file system installed in the internal eMMC

1. Optionally, backup the existing release kernel to avoid overwriting it with a new kernel copy.
2. Copy the compiled zImage kernel over the current L4T release kernel by executing the following command:

   ```
      $ cp arch/arm/boot/zImage <L4T_path>/Linux_for_Tegra/kernel
   ```

   Note: Running `zImage_to_uimg.sh` is not necessary because `flash.sh` converts the zImage to the `vmlinux.uimg` automatically for internal eMMC rootfs.

Flashing U-Boot

This section presents the theory of usage for flashing U-Boot followed by the commands used to flash. It contains the following topic:

- eMMC Partition as Script Partition
- Fastboot Creation of GPT
- Example eMMC Layout with Script Partition
- Flash Commands

eMMC Partition as Script Partition

With the standard NvFlash Utility and the `fastboot.bin` flash application, U-Boot shares the same internal eMMC partition layout as fastboot does. The only difference is that L4T U-Boot does not use the `kernel` partition in the usual fashion: U-Boot uses the `kernel` partition as its **boot-script partition**. Additionally, U-Boot expects the kernel file named `vmlinux.uimg` in the following directory:

```
<rootfs>/boot
```

Fastboot Creation of GPT

Fastboot creates the GPT partition, which contains the secondary GUID Partition Table (GPT) that must be located in the last 512-byte (B) sector of the boot device.

> **Note**: The boot device is not necessarily the same as the rootfs device.

The logical block addressing (LBA) of the last-512-B sector varies from device to device. Both U-Boot and the kernel know how to query the hardware to obtain the LBA of the last 512-B sector.

Example eMMC Layout with Script Partition

This topic provides an example eMMC layout showing the script partition and configuration (CFG) file contents.

U-Boot shares the same layout as fastboot; the layout is described in the following released configuration file:

```
gnu_linux_fastboot_emmc_full.cfg
```

The kernel partition (name=LNX) gets used as the script partition.

Example CFG Contents

```
[device]
type=sdmmc
instance=3

[partition]
name=BCT
id=2
type=boot_config_table
allocation_policy=sequential
filesystem_type=basic
size=4194304
file_system_attribute=0
partition_attribute=0
```

```
allocation_attribute=8
percent_reserved=0

[partition]
name=PT
id=3
type=partition_table
allocation_policy=sequential
filesystem_type=basic
size=2097152
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0

[partition]
name=EBT
id=4
type=bootloader
allocation_policy=sequential
filesystem_type=basic
size=2097152
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0
filename=fastboot.bin

[partition]
name=SOS
id=5
type=data
allocation_policy=sequential
filesystem_type=basic
size=6291456
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0
#filename=recovery.img

[partition]
name=LNX
id=6
type=data
allocation_policy=sequential
filesystem_type=basic
size=8388608
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0
filename=boot.img

[partition]
name=GP1
id=7
```

```
type=GP1
allocation_policy=sequential
filesystem_type=basic
size=2097152
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0

[partition]
name=APP
id=8
type=data
allocation_policy=sequential
filesystem_type=basic
size=1073741824
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0
filename=system.img

[partition]
name=CAC
id=9
type=data
allocation_policy=sequential
filesystem_type=basic
size=4194304
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0

[partition]
name=MSC
id=10
type=data
allocation_policy=sequential
filesystem_type=basic
size=2097152
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0

[partition]
name=USP
id=11
type=data
allocation_policy=sequential
filesystem_type=basic
size=4194304
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0
```

```
[partition]
name=UDA
id=12
type=data
allocation_policy=sequential
filesystem_type=basic
size=0xFFFFFFFFFFFFFFFF
file_system_attribute=0
partition_attribute=0
allocation_attribute=0x808
percent_reserved=0

[partition]
name=GPT
id=13
type=GPT
allocation_policy=sequential
filesystem_type=basic
size=0xFFFFFFFFFFFFFFFF
file_system_attribute=0
partition_attribute=0
allocation_attribute=8
percent_reserved=0
```

Flash Commands

This topic provides the commands to use to flash U-Boot so that you can boot the device either from internal eMMC, from an SD Card, or from an IP network.

To flash U-Boot to boot from internal eMMC

- Execute the following command:

  ```
  $ sudo ./flash.sh –L bootloader/<platform>/u-boot.bin
  <platform> mmcblk0p1
  ```

  Where `<platform>` is the device, such as cardhu for Tegra 3 or ventana for Tegra 2.

To flash U-Boot to boot from an SD Card

- Execute the following command:

  ```
  $ sudo ./flash.sh –L bootloader/<platform>/u-boot.bin
  <platform> mmcblk1p1
  ```

  Where `<platform>` is the device, such as cardhu for Tegra 3 or ventana for Tegra 2.

To flash U-Boot to boot from an IP network

- Execute the following command:

```
$ sudo ./flash.sh –L bootloader/<platform>/u-boot.bin –N
<IPA>:/<platform> [-n <target IPA>:<host IPA>:<gateway
IPA>:<netmask>] <platform> eth0
```

  Where `<platform>` is the device, such as `cardhu` for Tegra 3 or `ventana` for Tegra 2.

RootFS Tested By Device

This topic provides the results of testing the root file system location by device. The "Y" citations shown in the following table indicate that proper U-Boot initialization and hand-off to the kernel occurred, but this does not guarantee a fully-functional system.

RootFS Location	Cardhu	Ventana
mmcblk0p1	Y	Y
mmcblk1p1	Y	Y
sda1	Y	Y
eth0	Y	Y
eth1	Y	Not tested
usb0	Not tested	Not tested
mtdblock1	Unsupported	Unsupported

Example Hush Boot Scripts

This section provides example boot scripts for eMMC, SD Card, USB, and Network booting.

To minimize the foot print of the binary, U-Boot is designed to be configured by compile-time switches; however, in the Tegra Linux Driver (L4T) environment, it is desirable to have flash-time switches available to accommodate various L4T use cases. Therefore, to support flash-time switching, L4T implements a **script partition** concept, and the script partition can be edited and flashed to alter the U-Boot environment variables and boot parameters.

There are two example Hush shell script files supplied in Tegra Driver for Linux release: `<platform>_emmc.hush` and `<platform>_net.hush`. Hush is a lightweight Bourne shell-like command line interpreter. U-Boot executes the

Hush script to configure the boot device and build kernel command-line parameters.

eMMC Hush Script

The eMMC Hush script handles booting with boot order from all local storage devices, such as internal eMMC, SD Card, and USB devices. This script checks the USB device first, the SD Card next, and then the internal eMMC memory. During these checks, if U-Boot finds a valid `vmlinux.uimg` kernel image file, it boots off of the device where it found the kernel image.

The Hush script can be edited to set desired U-Boot boot strapping characteristics. You can find the eMMC Hush script at the following location:

```
bootloader/<platform>/<platform>_emmc.hush
```

where `<platform>` is the device, such as cardhu for Tegra 3 or ventana for Tegra 2.

Example Script Contents

The following shows the contents of the eMMC Hush script.

> **Note:** These contents are provided as an example; to see the latest values used in the release, look at the appropriate Hush scripts are included in the release in at:
>
> ```
> bootloader/<platform>/
> ```

```
#
# U-boot environment for local storage boot.
#
usbname=sda
pn=1
#
# Static env:
#
board=cardhu
mmcname=mmcblk
loadaddr=0x80408000
# loadaddr=0x80408000 for T30. For T20, 0x00408000
baudrate=115200
bootdelay=3
bootfile=vmlinux.uimg
console=ttyS0,115200n8
stderr=serial,lcd
stdin=serial,tegra-kbc
stdout=serial,lcd
user=user
videospec=tegrafb
serial#=1
```

```
#
# Board Specific variable env:
#
# tegra_fbmem: handled by u-boot runtime.
lp0_vec=0x00002000@0x9C406000
# lp0_vec=0x00002000@0x9C406000 for T30
# For T20, lp0_vec=0x00002000@0x1C406000
#
# Multi purpose env containers:
#
platform_extras=vmalloc=128M mem=1024M@2048M
extra_bootargs=usbcore.old_scheme_first=1 core_edp_mv=1300
panel=lvds tegraid=30.1.2.0.0 debug_uartport=lsport smp
#
# Local Storage Boot Scripts:
#
regen_all=setenv common_bootargs console=${console}
console=tty1 lp0_vec=${lp0_vec} video=${videospec}
${platform_extras} ${dev_extras} noinitrd; setenv bootargs
${common_bootargs} ${extra_bootargs} ${bootdev_bootargs}
mmc_setup=setenv bootdev_bootargs
root=/dev/${mmcname}${mmcdev}p${pn} rw rootwait gpt; run
regen_all
mmc_boot=run mmc_setup; mmc rescan ${mmcdev}; ext2load mmc
${mmcdev}:${pn} ${loadaddr} /boot/${bootfile}; bootm
${loadaddr}
mmc0_boot=setenv mmcdev 0; run mmc_boot
mmc1_boot=setenv mmcdev 1; run mmc_boot
usb_setup=setenv bootdev_bootargs root=/dev/${usbname}${pn} rw
rootwait; run regen_all
usb_boot=usb start; run usb_setup; ext2load usb 0:${pn}
${loadaddr} /boot/${bootfile}; bootm ${loadaddr}
bootcmd=run usb_boot; run mmc1_boot; run mmc0_boot
```

Net Hush Script

The Net Hush script handles booting from the Network File System (NFS). You can find the Net Hush script at the following location:

```
bootloader/<platform>/<platform>_net.hush
```

where `<platform>` is the device, such as cardhu for Tegra 3 or ventana for Tegra 2.

Example Script Contents

The following shows the contents of the Net Hush script.

> **Note:** These contents are provided as an example; to see the latest values used in the release, look at the appropriate Hush scripts are included in the release in at:
>
> ```
> bootloader/<platform>/
> ```

```
#
# U-boot environment for network boot.
#
#
# Static env:
#
board=cardhu
mmcname=mmcblk
loadaddr=0x80408000
# loadaddr=Ox80408000 for T30. For T20, 0x00408000
baudrate=115200
bootdelay=3
bootfile=vmlinux.uimg
console=ttyS0,115200n8
stderr=serial,lcd
stdin=serial,tegra-kbc
stdout=serial,lcd
user=user
videospec=tegrafb
serial#=1
#
# All CAP nfs info tokens need to be replaced with proper
values as follow:
# IPADDR ----- 172.17.187.71
# SERVERIP --- 172.17.186.62
# GATEWAYIP -- 172.17.186.62
# NETMASK ---- 255.255.252.0
# NFSARGS ---- ip=:::::eth0:on or
ip=cip:sip:gip:mask::eth0:off
# NFSROOT ---- 172.17.187.62:/cardhu_nfsroot
#
ipaddr=IPADDR
serverip=SERVERIP
gatewayip=GATEWAYIP
netmask=NETMASK
tftppath=TFTPPATH
nfsbootargs=root=/dev/nfs NFSARGS nfsroot=NFSROOT
#
# Board Specific variable env:
#
# tegra_fbmem: handled by u-boot runtime.
lp0_vec=0x00002000@0x9C406000
# lp0_vec=0x00002000@0x9C406000 for T30
# For T20, lp0_vec=0x00002000@0x1C406000
#
# Multi purpose env containers:
#
platform_extras=vmalloc=128M mem=1024M@2048M
extra_bootargs=usbcore.old_scheme_first=1 core_edp_mv=1300
panel=lvds tegraid=30.1.2.0.0 debug_uartport=lsport smp
#
# Local Storage Boot Scripts:
#
regen_all=setenv common_bootargs console=${console}
console=tty1 lp0_vec=${lp0_vec} video=${videospec}
${platform_extras} ${dev_extras} noinitrd; setenv bootargs
${common_bootargs} ${extra_bootargs} ${bootdev_bootargs}
```

```
regen_net_bootargs=setenv bootdev_bootargs rw ${nfsbootargs} ;
run regen_all

nfs_setup=setenv autoload n; run regen_net_bootargs
nfs_boot=run nfs_setup; dhcp; tftpboot ${loadaddr}
${tftppath}; bootm ${loadaddr}

bootcmd=usb start; run nfs_boot
```

Debugging U-Boot Environment

This section provides some debugging tips for your U-Boot environment. The examples shown, however, do not represent a comprehensive listing for U-Boot functionality. For a full listing of supported commands and their usage by U-Boot, see the U-Boot documentation and source.

For example, one common problem that can occur is when you create your own kernel and U-Boot has trouble finding it. To verify that U-Boot can read the device and also see the files in the file system, the commands listed in examples in this section may be beneficial. If a boot device is not found or the device has trouble booting with a kernel other than the reference kernel provided in the L4T release, check the examples in this section for debug assistance.

Interrupting U-Boot

You can interrupt U-Boot during boot.

To interrupt U-Boot

- Press any key during boot.

Getting Help

On the U-Boot terminal screen, you can type `help` at any time for the list of supported commands from the U-Boot terminal.

To see the U-Boot Help screen on Tegra 3

- Enter:

  ```
  Tegra3 # help
  ```

  The below shows example Help information printed when executing `help` on a Tegra 3 device.

  ```
  ?       - alias for 'help'
  ```

```
base    - print or set address offset
bdinfo  - print Board Info structure
boot    - boot default, i.e., run 'bootcmd'
bootd   - boot default, i.e., run 'bootcmd'
bootm   - boot application image from memory
bootp   - boot image via network using BOOTP/TFTP protocol
cls     - clear screen
cmp     - memory compare
coninfo - print console devices and information
cp      - memory copy
crc32   - checksum calculation
dcache  - enable or disable data cache
dhcp    - boot image via network using DHCP/TFTP protocol
echo    - echo args to console
editenv - edit environment variable
env     - environment handling commands
exit    - exit script
ext2load- load binary file from a Ext2 filesystem
ext2ls  - list files in a directory (default /)
false   - do nothing, unsuccessfully
fatinfo - print information about filesystem
fatload - load binary file from a dos filesystem
fatls   - list files in a directory (default /)
fdt     - flattened device tree utility commands
go      - start application at address 'addr'
gpio    - GPIO access
help    - print command description/usage
i2c     - I2C sub-system
icache  - enable or disable instruction cache
iminfo  - print header information for application image
imxtract- extract a part of a multi-image
itest   - return true/false on integer compare
loadb   - load binary file over serial line (kermit mode)
loads   - load S-Record file over serial line
loady   - load binary file over serial line (ymodem mode)
loop    - infinite loop on address range
md      - memory display
mm      - memory modify (auto-incrementing address)
mmc     - MMC sub system
mmcinfo - display MMC info
mtest   - simple RAM read/write test
mw      - memory write (fill)
nm      - memory modify (constant address)
ping    - send ICMP ECHO_REQUEST to network host
printenv- print environment variables
reset   - Perform RESET of the CPU
run     - run commands in an environment variable
saveenv - save environment variables to persistent storage
setenv  - set environment variables
sf      - SPI flash sub-system
showvar - print local hushshell variables
sleep   - delay execution for some time
source  - run script from memory
sspi    - SPI utility command
test    - minimal test like /bin/sh
tftpboot- boot image via network using TFTP protocol
time    - run a command and report its run time
```

```
true    - do nothing, successfully
usb     - USB sub-system
usbboot - boot from USB device
version - print monitor, compiler and linker version
```

Listing a Directory Structure

You can list the directory structure of a particular device. For example, you can list the directory structure of sda1 in U-Boot by typing: `usb 0:1` (for USB device 0 partition 1).

To list the directory structure

- Execute the following command:

  ```
    Tegra3 # ext2ls usb 0:1
  ```

  Note: This works on EXT3 file systems, as well.

  Example output follows:

```
 <DIR>       4096 .
 <DIR>       4096 ..
 <DIR>       4096 bin
 <DIR>       4096 boot
 <DIR>       4096 dev
 <DIR>       4096 etc
 <DIR>       4096 home
 <DIR>       4096 lib
 <DIR>       4096 lost+found
 <DIR>       4096 media
 <DIR>       4096 mnt
 <DIR>       4096 opt
 <DIR>       4096 proc
 <DIR>       4096 root
 <DIR>       4096 sbin
 <DIR>       4096 selinux
 <DIR>       4096 srv
 <DIR>       4096 sys
 <DIR>       4096 tmp
 <DIR>       4096 usr
 <DIR>       4096 var
```

Listing the Contents of a Directory

You can list the contents of any directory.

To list contents of a directory

- Execute following command:

  ```
  Tegra3 # ext2ls usb 0:1 $DIRECTORY
  ```

where $DIRECTORY is an expected path on the device.

For example, to list contents of the /boot directory where the vmlinux.uimg file should be, execute:

```
Tegra3 # ext2ls usb 0:1 /boot
```

Printing the U-Boot Environment

You can print the entire U-Boot environment.

To print the U-Boot environment

- Execute the following command:

  ```
  Tegra3 # printenv
  ```

Printing/Setting Environment Variables

You can print and set variables in the environment.

To print a variable in the environment

- Execute the following command:

  ```
  Tegra3 # printenv $ENV_VARIABLE
  ```

  where $ENV_VARIABLE refers to an environment variable in U-Boot.

  For example, to print the boot device partition number, execute:

  ```
  Tegra3 # printenv pn
  ```

  Output can be as follows:

  ```
  pn=1
  ```

To set a variable in the environment

- Execute the following command:

  ```
  Tegra3 # setenv $ENV_VARIABLE $NEW_VALUE
  ```

  where $ENV_VARIABLE refers to an environment variable in U-Boot and $NEW_VALUE is the new value for that variable.

  For example, to set the partition number variable, enter the following command:

```
Tegra3 # setenv pn 1
```

To save the modified environment

- Execute the following command:

  ```
  Tegra3 # saveenv
  ```

  The saved modified environment is preserved in case of resets and reboots.

Building Crosstool-ng Toolchain and glibc

This chapter provides instructions to build the Crosstool-ng supplied cross toolchain suite version 4.5.3 and the glibc suite with an Ubuntu host machine. The Crosstool-ng toolchain suite is similar to the toolchain used to produce the L4T release binaries.

Note: For an example Crosstool-ng configuration file, see the Appendix.

This document covers the following topics:

- Toolchain Information
- Host Setup
- Dependent Packages
- Building the Toolchain Suite
- Verifying the Build

Toolchain Information

The toolchain consists of the following elements:

- Crosstool-ng reference (http://crosstool-ng.org/)
- Cross Toolchain Version : 4.5.3
- glibc Version : 2.11

Host Setup

Ubuntu host systems must include the following:

- Ubuntu 10.04 32-bit distribution (Note: 64-bit distribution is not supported for building the toolchain)
- Fast host CPU like Core 2 Duo (to reduce build time)
- 1GB Free space on HDD
- 2GB SDRAM

Dependent Packages

On the Ubuntu distribution host machine, ensure the following packages are installed:

- mercurial
- bison
- flex
- gperf
- texinfo
- m4
- libtool
- automake

Note: The host system must be connected to the internet before running the commands below:

You can install the above packages using the following command:

```
$ sudo apt-get install mercurial bison flex gperf texinfo m4
libtool automake
```

Building the Toolchain Suite

To build the toolchain suite, perform the following tasks:

- Set the TOP_DIR Environment Variable and Create Directories
- Install autoconf-2.68
- Configuring crosstool-ng
- Invoke the Build

To set the TOP_DIR environment variable and create directories

1. To set the TOP_DIR variable to ${HOME}/crosstool enter the following command:

```
$ export TOP_DIR="${HOME}/crosstool"
```

2. In the ${TOP_DIR} directory, create subdirectories:

```
$ mkdir depends
$ mkdir crosstool-ng
$ cd depends
$ mkdir src
$ mkdir install
$ cd src
$ mkdir autoconf
$ mkdir ct-ng
```

To instal autoconf-2.68

1. Change to the autoconf directory and download autoconf-2.68.tar.bz2 by executing the following commands:

   ```
   $ cd ${TOP_DIR}/depends/src/autoconf
   $ wget http://ftp.gnu.org/gnu/autoconf/autoconf-2.68.tar.bz2
   ```

2. Extract and configure autoconf-2.68:

   ```
   $ tar xf autoconf-2.68.tar.bz2
   $ cd autoconf-2.68
   $ ./configure --
   prefix=${TOP_DIR}/depends/install/autoconf_install/autoconf-
   2.68-install
   ```

3. Make and install autoconf-2.68:

   ```
   $ make
   $ make install
   ```

To configure crosstool-ng

1. Change to the ct-ng directory:

   ```
   $ cd ${TOP_DIR}/depends/src/ct-ng
   ```

2. Add the autoconf-2.68-install directory to your path:

   ```
   $ export
   PATH=${TOP_DIR}/depends/install/autoconf_install/autoconf-
   2.68-install/bin:${PATH}
   ```

3. Clone the crosstool-ng repository:

   ```
   $ hg clone http://crosstool-ng.org/hg/crosstool-ng
   ```

4. Configure crosstool-ng:

   ```
   $ cd crosstool-ng
   $ ./bootstrap
   $ ./configure --prefix=${TOP_DIR}/depends/install/ct-
   ng_install/crosstool-ng-hg-install
   ```

5. Make and install crosstool-ng:

   ```
   $ make
   $ make install
   ```

6. Create the `${TOP_DIR}/crosstool-ng/src` directory for locally saving downloaded packages:

   ```
   mkdir ${TOP_DIR}/crosstool-ng/src
   ```

To invoke the build

1. Change to the `/crosstool-ng-hg-install/bin` directory:

```
$ cd ${TOP_DIR}/depends/install/ct-ng_install/crosstool-ng-hg-
install/bin
```

2. Copy the following content of `.config` from the Sample Crosstool-ng Configuration File appendix to this guide to a file called `.config.`

   Note: `.config` is a hidden file. After creating it, confirm it exists in the correct location by running `ls -a` in the directory.

3. Build ct-ng using 8 parallel paths:

   ```
   $./ct-ng oldconfig
   $./ct-ng build.8
   ```

   This will build the complete suite and install the binary components in `${TOP_DIR}/crosstool-ng/install`.

Verifying the Build

After a successful build, the following are the directories and files contained in the `${TOP_DIR}/crosstool-ng/install` directory, as reported by the `tree` application (where available).

```
$ tree -L 2

|-- arm-cortex_a9-linux-gnueabi
|   |-- bin
|   |-- debug-root
|   |-- include
|   |-- lib -> sysroot/lib
|   |-- lib32 -> lib
|   |-- lib64 -> lib
|   `-- sysroot
|-- bin
|   |-- arm-cortex_a9-linux-gnueabi-addr2line
|   |-- arm-cortex_a9-linux-gnueabi-ar
|   |-- arm-cortex_a9-linux-gnueabi-as
|   |-- arm-cortex_a9-linux-gnueabi-c++
|   |-- arm-cortex_a9-linux-gnueabi-cc -> arm-cortex_a9-linux-
gnueabi-gcc
|   |-- arm-cortex_a9-linux-gnueabi-c++filt
|   |-- arm-cortex_a9-linux-gnueabi-cpp
|   |-- arm-cortex_a9-linux-gnueabi-ct-ng.config
|   |-- arm-cortex_a9-linux-gnueabi-g++
|   |-- arm-cortex_a9-linux-gnueabi-gcc
|   |-- arm-cortex_a9-linux-gnueabi-gcc-4.5.3
|   |-- arm-cortex_a9-linux-gnueabi-gccbug
|   |-- arm-cortex_a9-linux-gnueabi-gcov
|   |-- arm-cortex_a9-linux-gnueabi-gprof
|   |-- arm-cortex_a9-linux-gnueabi-ld
|   |-- arm-cortex_a9-linux-gnueabi-ldd
|   |-- arm-cortex_a9-linux-gnueabi-nm
```

```
|   |-- arm-cortex_a9-linux-gnueabi-objcopy
|   |-- arm-cortex_a9-linux-gnueabi-objdump
|   |-- arm-cortex_a9-linux-gnueabi-populate
|   |-- arm-cortex_a9-linux-gnueabi-ranlib
|   |-- arm-cortex_a9-linux-gnueabi-readelf
|   |-- arm-cortex_a9-linux-gnueabi-size
|   |-- arm-cortex_a9-linux-gnueabi-strings
|   `-- arm-cortex_a9-linux-gnueabi-strip
|-- build.log.bz2
|-- include
|-- lib
|   |-- gcc
|   |-- ldscripts
|   `-- libiberty.a
|-- libexec
|   `-- gcc
`-- share
    `-- gcc-4.5.3
```

Software Features

This section describes the software features expected to be supported with this release of NVIDIA® Tegra® Linux Driver Package, which provides users with a complete package to bring up Linux on certain Tegra devices.

This release supports NVIDIA® Tegra® 3 series code-name Cardhu and NVIDIA® Tegra® 2 series code-name Ventana devices.

> **Note**: Always check the *Release Notes* for constraints related to these features.

Read the following sections to learn more about supported features in this release.

- Linux
- Graphics and Multimedia
- Decoders
- Encoders
- Container Formats
- Streaming Protocols
- Playback
- Camera
- Displays
- Power
- Boot Loaders

Linux

Kernel	Notes
Linux Kernel	3.1
Kernel Native Drivers	**Notes**
Audio	ALSA
External SD card	-
USB	Keyboard/mouse, mass storage device (MSD)
Wi-Fi	Firmware provided separately
Boot Devices	**Notes**
eMMC	-
External boot media	USB flash devices and hard-drives, SD Cards
Network file-system (NFS) boot	-
Additional Notes	
Unless otherwise noted, all features pertain to Cardhu and Ventana.	

Graphics and Multimedia

Media APIs	Notes
EGL 1.4 with EGLImage	-
GStreamer OpenMAX-IL plug-in	Both for A/V playback and A/V capture
Open GL ES path extensions	-
OpenGL ES 1.1	-
OpenGL ES 2.0	For automatic stereo support
OpenMAX IL 1.1 decoding	Planned to be deprecated in a later release
OpenMAX IL 1.1 encoding	Planned to be deprecated in a later release
X11	ABI: 5, 6, 7, 8, 10, 11, 12
X Resize, Rotate and Reflect Extension (RandR) 1.3	-
Additional Notes	
Unless otherwise noted, all features pertain to Cardhu and Ventana.	

Decoders

Audio Decoders (Tegra 3/Tegra 2)

The features in this table are supported by both Tegra 3 and Tegra 2.

Audio Decode	Profile	Sampling	Bitrate
AAC+	AAC-LC with SBR; mono, stereo, and 2-channel mixing only	Up to 48 kHz	Up to 320 kilobits per second (Kbps)
AAC-LC	Mono, stereo, and 2-channel [5.1]	Up to 48 kHz	Up to 320 Kbps
eAAC+	AAC-LC with SBR+PS; mono, stereo, and 2-channel mixing only	Up to 48 kHz	Up to 320 Kbps
AMR-NB	1 channel	Up to 8 kHz	Up to 12.2 Kbps
AMR-WB	1 channel	Up to 16 kHz	Up to 23.85 Kbps
MP3	2 channel	Up to 48 kHz	Up to 320 Kbps
MPEG-2 (MPEG-1 Layer 2)	2 channel	Up to 48 kHz	Up to 320 Kbps
Vorbis	Ogg Audio	Up to 48 kHz	Up to 256 Kbps
WAV linear PCM	16-bit, 2 channels	Up to 48 kHz	-
WMA 9 Std.	Standard 2-channel	Up to 48 kHz	Up to 384 Kbps
WMA Lossless	Lossless: Up to N1 Profile; WMA 10: 2 channel	Up to 48 kHz	-
WMA Pro LBR 10	M2 Profile; 2 channel [5.1]	Up to 96 kHz	Up to 768 Kbps

Image Decoders (Tegra 3/Tegra 2)

The features in this table are supported by both Tegra 3 and Tegra 2.

Image Decode	Profile	Resolution	Notes
JPEG	Baseline	Up to 20 MP	Gstreamer JPEG decoder supported by default

Video Decoders (Tegra 3)

The features in this table are supported by Tegra 3.

Video Decode	Profile and Level	Sampling Frequency and Bit rate/Frame rate	Notes
DivX 4/5/6	1080p HD Profile	Up to 1080p @ 30 fps Up to 10 Mbps	-
DivX 4/5/6	PlusHD	Up to 1080p @ 30 fps Up to 20 Mbps	-
H.263	Baseline Profile 0	Standard H.263 picture formats up to 4CIF @ 30 fps Up to 8 Mbps	Baseline Profile 0 Tools; Also supports custom formats at any resolution
H.264 AVC	Baseline Profile, Main Profile High Profile @ L4.1	Up to 720p @ 60 fps Up to 40 Mbps	-
H.264 AVC	Baseline Profile Main Profile High Profile @ L4.1	Up to 720p @ 60 fps Up to 40 Mbps	-
H.264 AVC	Baseline Profile Main Profile High Profile @ L4.1	Up to 1080i @ 60 fps Up to 40 Mbps	-
H.264 AVC	Baseline Profile Main Profile High Profile @ L4.1	Up to 1140p @ 30 fps Up to 40 Mbps	-
H.264 AVC (SEI)	High Profile	Up to 1080p @ 30 fps Up to 40 Mbps	-
MPEG-2 Video	Main Profile @ High Level	Up to 720p @ 60p Up to 80 Mbps	-
MPEG-2 Video	Main Profile @ High Level	Up to 1080p @ 30p/1080i 60 fps Up to 80 Mbps	-
MPEG-4	Advanced Simple Profile @ L5	Up to 1080p @ 30 fps Up to 10 Mbps	-

VC-1/WMV	Simple Profile	Up to 1080p @ 30 fps Up to 45 Mbps	-
VC-1/WMV	Main Profile	Up to 1080p @ 30 fps Up to 45 Mbps	-
VC-1/WMV	Advanced Profile	Up to 1080p @ 30 fps Up to 45 Mbps	-
Xvid	Highdef Profile	Up to 1080p @ 30 fps Up to 10 Mbps	-

Video Decoders (Tegra 2)

The features in this table are supported by Tegra 2.

Video Decode	Profile and Level	Sampling Frequency and Bit rate/Frame rate	Notes
DivX 4/5/6	1080p HD	Up to 1080p 30 fps Up to 10 Mbps	-
H.263	Baseline Profile 0	Standard H.263 formats up to 4CIF 30 fps Up to 8 Mbps peak	Standard H.263 picture formats = SQCIF, QCIF, CIF, 4CIF; Also supports custom formats at any resolution
H.264	Baseline Profile @ L4	Up to 1080p 30 fps Up to 20 Mbps peak	
H.264 Dynamic Resolution Change (DRC)	-	-	AP20 only
H.264	Main Profile @ L2.2	Up to 576i 60 fps Up to 4 Mbps	CABAC; MBAFF; Interlaced
H.264	Main Profile @ L3.1	Up to 720p 30 fps Up to 4 Mbps	CAVLC; With weighted prediction
H.264	Main Profile @ L3.1	Up to 720p 30 fps Up to 5 Mbps	CABAC; CAVLC; No weighted prediction
H.264	High Profile @ L2.2	Up to 576i 60 fps Up to 4 Mbps	CABAC; MBAFF; Interlaced

H.264	High Profile @ L3.1	Up to 720p 30 fps Up to 4 Mbps	CAVLC; With weighted prediction
H.264	High Profile @ L3.1	Up to 720p 30 fps Up to 5 Mbps	CAVLC; CABAC; No weighted prediction
MPEG-4	Advanced Simple Profile @ L5	Up to 1080p 30 fps Up to 10 Mbps	-
VC-1	Simple Profile	Up to 1080p 30 fps Up to 10 Mbps	-
VC-1	Main Profile	Up to 1080p 30 fps Up to 10 Mbps	-
VC-1	Advanced Profile	Up to 1080p 30 fps Up to 10 Mbps	-
Xvid	Highdef Profile	Up to 1080p 30 fps Up to 10 Mbps	-

Encoders

Audio Encoders (Tegra 3/Tegra 2)

The features in this table are supported by both Tegra 3 and Tegra 2.

Audio Encode	Profile	Resolution	Bit Rates
AAC-LC	-	-	Up to 320 kbps

Image Encoders (Tegra 3/Tegra 2)

The features in this table are supported by both Tegra 3 and Tegra 2.

Image Decode	Profile	Resolution	Bit Rates
JPEG	Exif	Up to 14 MP	Q-100

Video Encoders (Tegra 3)

The features in this table are supported by Tegra 3 only.

Video Encode	Profile and Level	Sampling Frequency and Bit rate/Frame rate	Notes
H.264	Baseline Profile	Up to 1080p 30 fps Up to 20 Mbps	-
MPEG4	Simple Profile	Up to 30fps	-
H.263	Baseline Profile	Up to 30fps	-

Video Encoders (Tegra 2)

The features in this table are supported by Tegra 2 only.

Video Encode	Profile and Level	Sampling Frequency and Bit rate/Frame rate	Notes
H.264	Baseline Profile	Up to 1080p 24 fps Up to 14 Mbps	-

Container Formats

Codecs are provided by GStreamer. You can download GStreamer codecs from the gstreamer opensource project at:

http://gstreamer.freedesktop.org

Or you can use `apt-get` in the provided Ubuntu-derived sample file system.

Hardware codecs are not included in the base release but can be provided separately under a software license agreement.

Streaming Protocols

Streaming protocols are provided by GStreamer. You can download GStreamer codecs from the gstreamer opensource project at:

http://gstreamer.freedesktop.org

Or you can use `apt-get` in the provided Ubuntu-derived sample file system.

Hardware codecs are not included in the base release but can be provided separately under a software license agreement.

Protocol	Notes
HTTP 1.0	ASF, AVI,3GP, MOV, MP3, MP4, WMA, WMV
HTTP 1.1	ASF, AVI,3GP, MOV, MP3, MP4, WMA, WMV
RTP/RTSP/RTCP	3GP, MP3
SDP (RFC 4566)	3GP

Playback

General	Notes
NvGstPlayer application	Supports the following functionality: · Basic local playback · Streaming playback with HTTP and RTSP · Trick modes · Audio/video synchronization · Automation for batch mode execution · Playbin2 support · Native pipeline support

Camera

Application	Feature	Notes
NvGstCapture application	Bayer camera sensor	-
	Camera capture mode selection	-
	Still image capture	-
	Video record	-
	Camera focus position control	-
	Camera exposure control	-
	Camera ISO control (exposure control)	-
	Camera contrast control	-
	Camera flash control	-
	Camera zoom control	-
	Camera pre-capture converge control	-
	Camera half press with 3A control	-
	Camera brightness control	-
	Camera saturation control	-
	Camera hue control	-
	Camera white balance control	-
	Image multishot	-
	Camera image filter control (color effect)	-
	Camera pause preview after capture	-
	Camera EV compensation Control	-
	Camera flicker control	-
	Camera edge enhancement control	-
	Camera meter mode control (exposure control)	-
	Camera preview rotate control	-
	Camera capture rotate control	-
	Camera timer control (delay capture)	-
	Camera scene mode control (non-standard modes)	-
	Camera noise reduction	-
	Camera still capture resolution	-
	Camera video capture resolution	-
	Camera video encoder selection	-
	Voice control	-

Application	Feature	Notes
	Camerabin2 support	GStreamer 10.36 only
	GstPhotography support	GStreamer 10.36 only

Displays

Feature	Description		Notes
HDMI 1.4a	Up to 1080p		-
Internal LCD	-		-
Feature	**Mode No.**	**Resolution**	**Rate**
HDMI supported modes	1	640 X 480	60
	2	640 X 480	72
	3	640 X 480	74
	4	720 x 480	59
	5	720 X 576	50
	6	800 X 600	59
	7	800 X 600	72
	8	800 X 600	74
	9	1024 X 768	60
	10	1024 X 768	70
	11	1024 X 768	74
	12	1152 X 864	74
	13	1280 X 720	60
	14	1280 X 1024	74
	15	1280 X 1024	59
	16	1680 X 1050	59
	17	1920 X 1080	50
	18	1920 X 1080	60
Feature	**Description**		**Notes**
HDMI DDC modesetting support	-		-
Dual display (LVDS + HDMI)	Mirror mode, Extended mode		-
Additional Notes			
DC fallback mode (640x480 @ 60Hz) is the default video mode for an HDMI monitor when using an fb_console device. See the *Release Notes* for known issues.			

Power

General	Notes
DVFS	-
Tegra Active Power Management	**Notes**
Dynamic voltage scaling for core	-
Dynamic voltage scaling for CPU	-
Dynamic frequency scaling for CPU	-
Dynamic frequency scaling for memory controller	-
Dynamic frequency scaling for AVP	-
Dynamic frequency scaling for system buses	-
Low power idle state (LP2)	Cardhu only
Frequency-boosted burst modes	Cardhu only; CPU running at higher frequencies in single-core mode
EDP support	Cardhu only
External temperature sensing	-
CPU hotplug support	Cardhu only
Switch to ULP CPU mode	Cardhu only
Dynamic thermal throttling	Software with hardware fail-safe
DRAM self-refresh	
Ultra-Low Power Standby	Notes
Suspend (LP1)	-
Deep Sleep (LP0)	-
USB Suspend during Deep Sleep	-
Wake with power button and RTC	-
Peripheral Power Management	Notes
NVIDIA Pixel Rendering Intensity and Saturation Management (PRISM)	Cardhu only; Display driver
Additional Notes	
Unless otherwise noted, all features pertain to Cardhu and Ventana.	

Boot Loaders

General	Notes
Fastboot	Supports booting from USB, SD card, internal memory (eMMC), and network file-system (NFS) boot
U-Boot	Source code available. Supports booting from USB, SD card, internal memory (eMMC), network file-system (NFS) boot, and Trivial File Transfer Protocol (TFTP) boot

Licenses

This section provides license information for the following:

- NVIDIA Software
- Sample File System
- Package Codecs
- GST OpenMAX
- Linux Kernel
- mkbootimg and mkubootscript
- mkyaffs2img
- U-Boot and mkimage

NVIDIA Software

Note: This software license applies to software in the "nv_tegra" directory (NVIDIA binary drivers and supporting software).

License For Customer Use of NVIDIA Software

IMPORTANT NOTICE -- READ CAREFULLY: This License For Customer Use of NVIDIA Software ("LICENSE") is the agreement which governs use of the software of NVIDIA Corporation and its subsidiaries ("NVIDIA") downloadable, including computer software and associated printed materials ("SOFTWARE"). By downloading, installing, copying, or otherwise using the SOFTWARE, you agree to be bound by the terms of this LICENSE. If you do not agree to the terms of this LICENSE, do not download the SOFTWARE.

RECITALS

Use of NVIDIA's products requires three elements: the SOFTWARE, the hardware on a graphics controller board, and a personal computer. The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is not sold, and instead is only licensed for use, strictly in accordance with this document. The hardware is protected by various patents, and is sold, but this agreement does not cover that sale, since it may not necessarily be sold as a package with the SOFTWARE. This agreement sets forth the terms and conditions of the SOFTWARE LICENSE only.

1. DEFINITIONS

1.1 Customer.

Customer means the entity or individual that downloads the SOFTWARE.

2. GRANT OF LICENSE

2.1 Rights and Limitations of Grant.

NVIDIA hereby grants Customer the following non-exclusive, non-transferable right to use the SOFTWARE, with the following limitations:

2.1.1 Rights.

Customer may install and use one copy of the SOFTWARE on a single computer, and except for making one back-up copy of the Software, may not otherwise copy the SOFTWARE. This LICENSE of SOFTWARE may not be shared or used concurrently on different computers.

2.1.2 Linux/FreeBSD Exception.

Notwithstanding the foregoing terms of Section 2.1.1, SOFTWARE designed exclusively for use on the Linux or FreeBSD operating systems, or other operating systems derived from the source code to these operating systems, may be copied and redistributed, provided that the binary files thereof are not modified in any way (except for unzipping of compressed files).

2.1.3 Limitations.

No Reverse Engineering. Customer may not reverse engineer, decompile, or disassemble the SOFTWARE, nor attempt in any other manner to obtain the source code.

No Separation of Components. The SOFTWARE is licensed as a single product. Its component parts may not be separated for use on more than one computer, nor otherwise used separately from the other parts.

No Rental. Customer may not rent or lease the SOFTWARE to someone else.

3. TERMINATION

This LICENSE will automatically terminate if Customer fails to comply with any of the terms and conditions hereof. In such event, Customer must destroy all copies of the SOFTWARE and all of its component parts.

Defensive Suspension. If Customer commences or participates in any legal proceeding against NVIDIA, then NVIDIA may, in its sole discretion, suspend or terminate all license grants and any other rights provided under this LICENSE during the pendency of such legal proceedings.

4. COPYRIGHT

All title and copyrights in and to the SOFTWARE (including but not limited to all images, photographs, animations, video, audio, music, text, and other information incorporated into the SOFTWARE), the accompanying printed materials, and any copies of the SOFTWARE, are owned by NVIDIA, or its suppliers. The SOFTWARE is protected by copyright laws and international treaty provisions. Accordingly, Customer is required to treat the SOFTWARE like any other copyrighted material, except as otherwise allowed pursuant to this LICENSE and that it may make one copy of the SOFTWARE solely for backup or archive purposes.

5. APPLICABLE LAW

This agreement shall be deemed to have been made in, and shall be construed pursuant to, the laws of the State of California.

6. DISCLAIMER OF WARRANTIES AND LIMITATION ON LIABILITY

6.1 No Warranties.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE IS PROVIDED "AS IS" AND NVIDIA AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

6.2 No Liability for Consequential Damages.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL NVIDIA OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. MISCELLANEOUS

The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed. If any provision of this LICENSE is inconsistent with, or cannot be fully enforced under the law, such provision will be construed as limited to the extent necessary to be consistent with and fully enforceable under the law. This agreement is the final, complete and exclusive agreement between the parties relating to the subject matter hereof, and supersedes all prior or contemporaneous understandings and agreements relating to such subject matter, whether oral or written. Customer agrees that it will not ship, transfer or export the SOFTWARE into any country, or use the SOFTWARE in any manner, prohibited by the United States Bureau of Export Administration or any export laws, restrictions or regulations. This LICENSE may only be modified in writing signed by an authorized officer of NVIDIA.

Sample File System

The sample root file system is derived from Ubuntu Linux, version 12.04 for the hardware floating point (hardfp) release. Information on re-creating the root file system is provided in the T*egra Linux Driver Package Developers' Guide*. The license agreement for each software component is located in the software component's source code, made available from the same location from which this software was downloaded, or by request to oss-requests@nvidia.com.

Package Codecs

Note: This software license applies to codec package of NVIDIA® Tegra® Linux Driver Package.

NVIDIA Tegra Software License Agreement – Tegra Linux Driver Package

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE SOFTWARE (AS DEFINED BELOW) THE END USER OF THE LICENSED MATERIALS ("YOU" OR "LICENSEE") AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL

ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

- **DO NOT (A) DOWNLOAD, INSTALL, COPY THE SOFTWARE; OR (B) ACCESS OR USE THE LICENSED MATERIALS; AND**
- **PROMPTLY DESTROY THE LICENSED MATERIALS, OR RETURN THEM TO THE PARTY FROM WHOM YOU ACQUIRED IT.**

NVIDIA Tegra Software License Agreement– Tegra Linux Driver Package (the "Agreement") is entered into by and between NVIDIA Corporation, a Delaware corporation, having its principal place of business at 2701 San Tomas Expressway, Santa Clara, CA 95050 ("NVIDIA") and the individual person or single legal entity ("Licensee" or "You") who acknowledges and agrees to fully abide the terms and conditions of this Agreement.

1. **DEFINITIONS**.

1.1 "Affiliate" means any company or legal entity that at various times controls, is controlled by, or is under common control with Licensee. Only for the purposes of this definition, "Control" means (a) direct or indirect ownership of at least fifty percent (50%) of the voting power of the shares or other securities for election of directors (or other managing authority) of the controlled or commonly controlled entity; (b) holding, directly or indirectly, the power to exercise more than fifty percent (50%) of the entity's voting rights; or (c) holding, directly or indirectly, the power to appoint the majority of the members of the entity's board of directors (or similar governing body), or in each case, the maximum percentage permitted where a lesser percentage is required in a jurisdiction. The parties shall be fully responsible for the actions / inactions of their Affiliates under this Agreement.

1.2 "Competitors" shall mean any company that develops, manufactures, produces, sells, distributes or licenses application processors, computers-on-chips, systems-on-chips, CPUs, DSPs or GPU technology.

1.3 "Confidential Information" shall mean (a) the Licensed Materials; (b) either parties' technology, ideas, know-how, documentation, processes, algorithms and trade secrets embodied in the Licensed Materials; (c) any other information disclosed by either party to the other that is (i) identified as "confidential," "proprietary" or with a similar legend at the time of disclosure, or (ii) if unmarked or disclosed orally or visually, are identified as confidential at the time of disclosure and confirmed by a written memorandum sent to the receiving party within thirty (30) calendar days of disclosure summarizing the confidential information sufficiently for identification; and (d) the terms and conditions of this Agreement.

Confidential Information shall not include any information which is (e) published or otherwise available to the public other than by breach of this Agreement by the receiving party; (f) rightfully received by the receiving party from a third party without confidentiality limitations; (g) independently developed by the receiving party or its Affiliates as evidenced by appropriate records; (h) known to the receiving party prior to its first receipt of same from the disclosing party as evidenced by appropriate records; (i) hereinafter disclosed by the disclosing party to a third party without restriction on disclosure; or (j) approved for public release by written authorization of the disclosing party.

1.4 "Contractor" shall mean any third party company or individuals, including but not limited to original device manufacturers, who Licensee engages for the purpose of such third party performing services for the benefit of Licensee in connection with this Agreement.

1.5 "Derivative Work(s)" means derivatives or modifications of the Licensed Materials created by Licensee or NVIDIA, or a third party on behalf of Licensee or NVIDIA respectively, which term shall include: (i) for copyrightable or copyrighted material, any translation, abridgement, revision or other form in which an existing work may be recast, transformed or adapted; (ii) for work protected by topography or mask right, any translation, abridgement, revision or other form in which an existing work may be recast, transformed or adapted; (iii) for patentable or patented material, any improvement; and (iv) for material protected by trade secret, any new material derived from or employing such existing trade secret.

1.6 "Excluded License" means any license that requires as a condition of use, modification and/or distribution of software subject to the Excluded License, that such software or other software distributed and/or combined with such software be (i) disclosed or distributed in source code form, (ii) licensed for the purpose of making derivative works, or (iii) redistributable at no charge.

1.7 "Intellectual Property Rights" shall mean all proprietary rights, including all patents, trademarks, copyrights, know-how, trade secrets, mask works, including all applications and registrations thereto, and any other similar protected rights in any country.

1.8 "Licensed Materials" shall mean the Software, related documentation, Tegra development hardware (if applicable), and other materials as NVIDIA may deliver hereunder from time to time.

1.9 "Licensee Products" shall mean Licensee's devices that have or will contain NVIDIA's family of graphics, or media and communication, or applications processors and related Software (as defined below) supplied directly or indirectly by NVIDIA.

1.10 "Software" shall mean the NVIDIA Tegra Linux Driver Package, full or partial copies thereof, and any Derivative Work(s) thereto owned by NVIDIA.

2. **LICENSE**.

2.1 Grant. Subject to the terms and conditions of this Agreement, including applicable Exhibits, NVIDIA grants to Licensee and its Affiliates a personal, nonexclusive, worldwide, nonsublicensable, nontransferable, nonassignable and royalty-free right and license:

(a) to execute, compile, reproduce, display, perform, modify, and to prepare and have prepared Derivative Work(s) of the Software (in source code form as provided by NVIDIA) solely to develop and customize Licensee Products for Licensee's internal development purposes only;

(b) to reproduce, transmit, transfer, distribute and sublicense object code forms of the Software and/or Derivative Work(s) and related documentation incorporated into Licensee Products with a form of end user license agreement that is as protective of NVIDIA's Intellectual Property Rights as this Agreement;

(c) for Contractors to exercise the foregoing rights of Section 2.1 of this Agreement solely on behalf of Licensee; and

(d) NVIDIA may, in its sole discretion, require Licensee to accept, distribute, and/or incorporate certain modifications, updates, fixes, changes, or revisions to the Licensed Materials used in Licensee Products in a timely manner.

2.2 Reservation of Rights. NVIDIA reserves all rights not expressly granted to Licensee in Section 2.1 herein.

2.3 License Grant Back. Licensee hereby grants to NVIDIA and its Affiliates an exclusive, worldwide, irrevocable, perpetual, sublicensable (through multiple tiers of sublicensees), royalty-free right, fully paid-up right and license to the Derivative Work(s) (in source and object code form) created by Licensee's employees, Affiliates or Contractors so that NVIDIA may copy, modify, create Derivative Works thereof, to use, have used, import, make, have made, sell, offer to sell, sublicense (through multiple tiers of sublicensees), distribute (through multiple tiers of distributors) such Derivative Work(s) on a stand-alone basis or as incorporated into the

Software or other NVIDIA products. For the sake of clarity, NVIDIA is not prohibited or otherwise restricted from independently developing new features or functionality with respect to the Licensed Materials.

2.4 Delivery Obligation of Derivative Work(s) Licensee shall deliver, upon NVIDIA's request, the Derivative Work(s) created by Licensee or on behalf of Licensee to NVIDIA pursuant to Section 2.3 of this Agreement.

3. LIMITATIONS; OBLIGATIONS.

3.1 Restrictions. Except as expressly permitted by this Agreement, Licensee shall not:

(a) use the Software and/or the Derivative Work(s) created by Licensee or on behalf of Licensee on any non-NVIDIA application processors ("External Systems"), except on External Systems for the sole purpose of programming, configuration or performing diagnostics on an NVIDIA application processor;

(b) reverse engineer, decompile, disassemble, modify or create derivative works of any portion of the Licensed Materials (in object code form) or allow any third party (including Licensee's Affiliates or Contractors) to do any of the foregoing;

(c) sublicense, rent, lease, loan, timeshare, sell, distribute, disclose, publish, assign or transfer any rights, grant a security interest in, or transfer possession of the Licensed Materials to any third party without NVIDIA's express prior written consent;

(d) distribute the Licensed Materials on a standalone basis; or

(e) under any circumstances allow the Software to be used, pursuant to this Agreement, on NVIDIA's Competitors' software operating and/or hardware platforms.

3.2 No Implied Licenses. Nothing in this Agreement shall be construed as granting to Licensee by implication, estoppel or otherwise, (a) a license to any NVIDIA technology other than the Licensed Materials; or (b) any additional license rights for the Licensed Materials other than the licenses expressly granted in this Agreement.

3.3 Additional Licensing Obligations. Licensee acknowledges and agrees that it is Licensee's sole responsibility to obtain any, additional, third party licenses required to make, have made, use, have used, sell, import, and offer for sale Licensee Products that include or incorporate any third party technology such as operating systems, audio and/or video encoders and decoders or any technology from, including but not limited to, Microsoft, Thomson, Fraunhofer IIS, Sisvel S.p.A., MPEG-LA, and Coding Technologies (“Third Party Components”). Licensee acknowledges and agrees that NVIDIA has not granted to Licensee under this Agreement any necessary patent rights with respect to those Third Party Components identified in the exhibits of this Agreement (“Third Party Licensing Terms and Notices”). As such, Licensee’s use of the Third Party Components may be subject to further restrictions and terms and conditions described in the Third Party Licensing Terms and Notices. Licensee acknowledges and agrees that Licensee is solely and exclusively responsible for obtaining any and all authorizations and licenses required for the distribution and/or incorporation of the Third Party Components specified in the Third Party Licensing Terms and Notices.

For the avoidance of doubt, except as expressly authorized by a separate written agreement by and between Licensee and Adobe Systems, Inc. (“Adobe”), Licensee has no right to distribute, sublicense, or otherwise commercialize Adobe’s Third Party Components identified in Exhibit A-2.

Excluding Section 9.5 of this Agreement, Licensee acknowledges and agrees that NVIDIA may at various times update the Third Party Licensing Terms and Notices without any advance written notice to Licensee. Licensee agrees to be bound by such Third Party Licensing Terms and Notices as they may be updated. In the event that NVIDIA’s license rights to the Third Party Components are terminated and/or expired, Licensee agrees NVIDIA shall no longer have the obligation to deliver such Third Party Components to Licensee affected by such termination and/or expiration.

Licensee shall, at its own expense fully indemnify, hold harmless, defend, and settle any claim, suit or proceeding that is instituted by a third party against NVIDIA and its officers, employees or agents, to the extent such claim, suit or proceeding is based on (a) a breach by Licensee of any of the representations and warranties in Section 7 (“Warranties”) of this Agreement; or (b) Licensee’s failure to fully satisfy and/or comply with the third party licensing obligations expressly contained in the Third Party Licensing Terms and Notices (a “Claim”).

In the event of a Claim, NVIDIA agrees to:

(a) promptly inform Licensee and furnish Licensee a copy of the Claim;

(b) make commercially reasonable efforts to give such evidence in NVIDIA’s possession, custody or control as is reasonable to Licensee, at

Licensee's request and expense, specifically and reasonably applicable to the Claim;

(c) provide Licensee commercially reasonable assistance in the defense thereof, at Licensee's expense; and

(d) give Licensee sole control of the defense thereof and all negotiations for its settlement and compromise, which shall not be finalized without the prior written consent of NVIDIA.

NVIDIA's failure to promptly notify Licensee shall not relieve Licensee of any liability or obligations that it has to NVIDIA, except to the extent Licensee demonstrates that the defense of such action is prejudiced by the failure or delay in giving notice. If NVIDIA retains counsel, it will be at NVIDIA's own expense.

In the event of a Claim, Licensee agrees to:

(a) pay all damages finally awarded against NVIDIA or agreed upon in settlement by Licensee, which shall not be finalized without the prior written consent of NVIDIA, (including other reasonable costs incurred by NVIDIA, including reasonable attorneys fees, in connection with enforcing this paragraph);

(b) reimburse NVIDIA for any licensing fees and/or penalties incurred by NVIDIA in connection with a Claim; and

(c) immediately procure/satisfy the third party licensing obligations expressly contained in the Third Party Licensing Terms and Notices.

3.4 Proprietary Rights Notices. Licensee shall not remove, alter or obscure any copyright, trademark, patent notices or other proprietary rights notices that appear on the Licensed Materials. Licensee shall use commercially reasonable efforts to require its channel entities to comply with the provisions of this Section 3.4.

3.5 No Excluded Licenses. The licenses granted in Section 2.1 do not include the right to, and Licensee shall not: (a) create Derivative Work(s) of the Licensed Materials in any manner that would cause the Licensed Materials, in whole or in part, to become subject to the terms of an Excluded License; or (b) distribute the Licensed Materials (or Derivative Works thereof) in any manner that would cause the Licensed Materials, or any component thereof, to become subject to the terms of an Excluded License.

3.6 Source Code Protection. In addition to Licensee's restrictions and obligations in connection with the Licensed Materials set forth in this Agreement, Licensee agrees that source code to the Licensed Materials constitutes highly Confidential Information and proprietary trade secrets of NVIDIA and shall be protected by (a) the confidentiality obligations set forth in Section 5.1; and (b) any applicable non-disclosure agreement ("NDA"). In addition to the confidentiality obligations set forth in Section 5.1 and the NDA, Licensee agrees to the following:

(a) Licensee shall only allow its employees, Contractors, and its Affiliates' employees and Contractors who have a need to know basis to use the source code to the Software in order for Licensee or its Affiliates to exercise their license rights under this Agreement, provided that any breach of this Agreement by such parties is considered Licensee's breach of this Agreement and Licensee shall be liable for such breach to the same extent as if it committed the breach itself. Upon NVIDIA's request, Licensee shall provide NVIDIA a list of all employees (including employees of Affiliates), and Contractors who have been granted source code access to the Licensed Materials and update and maintain the accuracy of this list at all times;

(b) Licensee shall protect the source code of the Licensed Materials to the same degree as Licensee protects its own Confidential Information;

(c) Licensee shall not grant third parties, excluding Affiliates or Contractors, access to the source code of the Licensed Materials;

(d) Licensee shall restrict disclosure and access to and use of the Licensed Materials (in source code form) to those employees (including those of its Affiliates and/or Contractors) who have agreed to be bound by a written confidentiality agreement which incorporates the protections and restrictions no less protective than those set forth in this Agreement with respect to the Licensed Materials;

(e) Licensee shall secure the source code to the Software and Licensed Materials in a secure location at all times;

(f) Licensee shall not use the Licensed Materials and/or Derivative Work(s) created by Licensee to compete against NVIDIA or shall not use the Licensed Materials and/or Derivative Work(s) in litigation against NVIDIA; and

(g) Licensee's employees, Affiliates, or Contractors who have been exposed to source code of the Licensed Materials shall not be permitted to use any ideas, techniques or know-how obtained from their respective use of the Licensed Materials for any engagement, including but not limited to services or product development (hardware or software) work for the benefit of NVIDIA's Competitors.

3.7 Defensive Suspension. If Licensee and/or its Affiliates commence or participates in any legal proceeding against NVIDIA, then NVIDIA may, in its sole discretion, suspend or terminate all license grants and any other rights provided under this Agreement during the pendency of such legal proceedings.

4. OWNERSHIP; FEEDBACK.

4.1 By NVIDIA. Except as expressly licensed to Licensee under this Agreement, NVIDIA reserves all right, title and interest, including but not limited to all Intellectual Property Rights, in and to the Licensed Materials and any Derivative Work(s) made thereto by or on behalf of NVIDIA.

4.2 Feedback by Licensee. Licensee may, but is not obligated to, provide to NVIDIA any suggestions, comments and feedback regarding the Licensed Materials that are delivered by NVIDIA to Licensee under this Agreement (collectively, "Licensee Feedback"). NVIDIA may use and include any Licensee Feedback that Licensee voluntarily provides to improve the Licensed Materials or other related NVIDIA technologies. Accordingly, if Licensee provides Licensee Feedback, Licensee grants NVIDIA and its licensees a perpetual, irrevocable, worldwide, royalty-free, fully paid-up license grant to freely use, have used, sell, modify, reproduce, transmit, license, sublicense (through multiple tiers of sublicensees), distribute (through multiple tiers of distributors), and otherwise commercialize the Licensee Feedback in the Licensed Materials or other related technologies.

5. CONFIDENTIAL INFORMATION; ACCESS TO SOFTWARE.

5.1 Protection of Confidential Information. The parties shall not use or disclose any Confidential Information received from the other party, except as expressly authorized by this Agreement, and shall protect all such Confidential Information using the same degree of care which the receiving party uses with respect to its own proprietary information, but in no event with safeguards less than a reasonably prudent business would exercise under similar circumstances. The parties shall not use the Confidential Information for purposes other than those necessary to directly further the purposes of this Agreement. Except as expressly provided in this Agreement, no ownership or license rights are granted in any Confidential Information. The parties shall use commercially reasonable efforts to prevent any actual or threatened unauthorized copying, use or disclosure of

Confidential Information, and shall promptly notify the other party of any such actual or threatened unauthorized disclosure or use. If any Confidential Information must be disclosed to any third party by reason of legal, accounting or regulatory requirements beyond the reasonable control of the disclosing party, the disclosing party shall promptly notify the other party of the order or request and permit the other party (at its own expense) to seek an appropriate protective order.

For the sake of clarity, the parties agree that, notwithstanding the preceding paragraph, any and all information identified as Confidential Information (as defined in the NDA) by the disclosing party in connection with this Agreement shall also be protected under the NDA; provided, however, that in the event of any conflict between the confidentiality obligations pursuant to this Agreement and the obligations pursuant to the NDA with regard to any Confidential Information (as defined in the NDA) in connection with this Agreement, including, without limitation, the source code to the Licensed Materials, the terms of this Agreement shall prevail.

6. TERM; TERMINATION

6.1 Term. This Agreement and the licenses granted hereunder shall be effective as of the date Licensee first uses the Licensed Materials ("Effective Date") and continue for a period of one (1) year (the "Initial Term"), unless terminated in accordance with Section 6.2. Unless either party notifies the other party of its intent to terminate this Agreement at least one (1) month prior to the end of the Initial Term or the applicable renewal period ("Renewal Period(s)"), this Agreement will be automatically renewed for one (1) year Renewal Periods, provided however that this Agreement will automatically expire at such time when Licensee no longer intends to use the Licensed Materials for the authorized purposes described in this Agreement, at this time Licensee will comply with the termination provisions in Section 6.2 below.

6.2 Termination. Either party may terminate this Agreement immediately upon written notice for the material breach of the other party, which material breach is curable and has remained uncured for a period of thirty (30) days from the date of delivery of written notice thereof to the other party. Upon the termination or expiration of this Agreement,

(a) Licensee shall (i) immediately cease using the Licensed Materials for any purpose whatsoever; (ii) immediately destroy or return to NVIDIA all materials belonging to NVIDIA, including without limitation all copies of the Software and NVIDIA Confidential Information then in

Licensee's possession or control; and (iii) certify to NVIDIA in writing that it has done so; and

(b) NVIDIA shall (i) immediately destroy or return to Licensee all materials belonging to Licensee that were provided to NVIDIA pursuant to this Agreement, including without limitation, Licensee's Confidential Information then in NVIDIA's possession or control; and (ii) certify to Licensee in writing that it has done so. These remedies shall be cumulative and in addition to any other remedies available to NVIDIA.

6.3 Survival. Those provisions in this Agreement, which by their nature need to survive the termination or expiration of this Agreement, The following Sections shall survive termination or expiration of the Agreement, including but not limited to Sections 1, 2.2, 2.3, 2.4, 3, 4, 5, 6.2 , 6.3, 7, 8, 9 and Exhibit A.

7. WARRANTIES

THE LICENSED MATERIALS ARE LICENSED FOR LICENSEE'S USE "AS IS" AND NVIDIA AND ITS LICENSORS DISCLAIM ALL WARRANTIES, EXPRESS, IMPLIED AND STATUTORY INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. NVIDIA DOES NOT REPRESENT OR WARRANT THAT THE LICENSED MATERIALS WILL MEET LICENSEE'S REQUIREMENTS OR THAT THE OPERATION OF THE SOFTWARE CONTAINED THEREIN OR RESULTING THEREFROM WILL BE UNINTERRUPTED OR ERROR-FREE. NO INFORMATION OR ADVICE GIVEN BY NVIDIA, ITS REPRESENTATIVES, AGENTS OR EMPLOYEES SHALL IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY.

(A) Licensee represents and warrants that it has, or will have prior to the commercial release of the Licensee Products, a valid and current license to all the Third Party Components referenced in the exhibits of this Agreement, for use in connection with Licensed Materials provided pursuant to this Agreement and Licensee Products.

8. LIMITATION OF LIABILITY

IN NO EVENT SHALL:

(A) NVIDIA BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, INCIDENTAL, PUNITIVE OR SPECIAL DAMAGES, OF ANY KIND OR CHARACTER, INCLUDING LOST PROFITS, LOST REVENUE, FAILURE TO REALIZE SAVINGS OR OTHER BENEFITS, LOSS OF DATA OR USE, AND CLAIMS BY ANY THIRD PARTY, ARISING OUT OF OR RELATED TO THE SUBJECT MATTER OF THIS AGREEMENT; AND

(B) NVIDIA'S AGGREGATE LIABILITY ARISING OUT OF THIS AGREEMENT EXCEED THE AMOUNT PAID BY LICENSEE FOR USE OF THE LICENSED MATERIALS. THE FOREGOING EXCLUSION AND LIABILITY LIMITATIONS APPLY EVEN IF SUCH PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS EXCLUSION AND LIABILITY LIMITATION SHALL APPLY EVEN IF ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE.

9. GENERAL

9.2 Governing Law. This Agreement shall be governed in all respects by the laws of the United States and of the State of Delaware, without regard to the conflicts of laws principles thereof.

9.3 Jurisdiction. The state and/or federal courts residing in Santa Clara County, California shall have exclusive jurisdiction over any dispute or claim arising out of this Agreement.

9.4 Severability. If for any reason a court of competent jurisdiction finds any provision of this Agreement, or portion thereof, to be unenforceable, that provision of the Agreement will be enforced to the maximum extent permissible so as to affect the intent of the parties, and the remainder of this Agreement will continue in full force and effect. This Agreement has been negotiated by the parties and their respective counsel and will be interpreted fairly in accordance with its terms and without any strict construction in favor of or against either party.

9.5 Amendments. The Agreement shall not be modified except by a written agreement that names this Agreement and any provision to be modified, is dated subsequent to the Effective Date, and is signed by duly authorized representatives of both parties.

9.6 No Waiver. No failure or delay on the part of either party in the exercise of any right, power or remedy under this Agreement or under law, or to insist upon or enforce performance by the other party of any of the provisions of this Agreement or under law, shall operate as a waiver thereof, nor shall any single or partial exercise of any right, power or remedy preclude other or further exercise thereof, or the exercise of any other right, power or remedy; rather the provision, right, or remedy shall be and remain in full force and effect.

9.7 No Assignment. This Agreement, and each party's rights and obligations herein, may not be assigned, subcontracted, delegated, or otherwise transferred by either party without the other party's prior written consent, and any attempted assignment, subcontract, delegation, or transfer in violation of the foregoing will be null and void. The terms of this Agreement shall be binding upon assignees.

9.8 Independent Contractors. NVIDIA's relationship to Licensee is that of an independent Contractor, and neither party is an agent or partner of the other. Neither party will have, and will not represent to any third party that it has, any authority to act on behalf of the other party.

9.9 Export Restrictions. The parties acknowledge that the Licensed Materials are subject to U.S. export control laws and regulations. The parties agree to comply with all applicable international and national laws that apply to the Licensed Materials, including the U.S. Export Administration Regulations, as well as end-user, end-use and destination restrictions issued by U.S. and other governments.

9.10 U.S. Government Legend. If Licensee is a branch or agency of the United States Government, the following provision applies. Any software provided under this Agreement, including any releases are comprised of "commercial computer software" and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 and are provided to the Government (i) for acquisition by or on behalf of civilian agencies, consistent with the policy set forth in 48 C.F.R. 12.212; or (ii) for acquisition by or on behalf of units of the Department of Defense, consistent with the policies set forth in 48 C.F.R. 227.7202-1 and 227.7202-3.

9.11 Headings. The headings in this Agreement are for the sole purpose of convenience of reference and shall not in any way limit or affect the meaning or interpretation of any of the terms or provisions of this Agreement.

9.12 Counterparts. This Agreement may be executed in counterparts, each of which shall be deemed an original, and all of which together shall constitute one instrument.

9.13 No Third Party Beneficiaries. This Agreement is solely between NVIDIA and Licensee. There are no third party beneficiaries, express or implied, to this Agreement.

9.14 Entire Agreement. This Agreement constitutes the entire agreement between the parties with respect to the subject matter contemplated herein, and merges all prior and contemporaneous communications.

Exhibit A

1. Coding Technologies/AAC+

Licensee shall be solely responsible for either obtaining a proper patent license under the Essential Patents for end products or to notify Licensee's respective customers of their obligations to obtain a proper patent license under the Essential Patents for end products in which the NVIDIA application processor(s) and/or the NVIDIA software package may be used. For the purpose of this paragraph, "Essential Patents" means patents which are infringed by the manufacture, offer for sale, sale (or other form of commercialization), use or import of products (hardware or software) implementing, incorporating, containing or using AACPLUSV2 or by the application of processes involving AACPLUSV2, including those which are infringed by any source code provided as part of any specification characterizing AACPLUSV2.

2. Thomson Multimedia/MP3

Supply of the Licensed Materials does not convey a license under the relevant intellectual property of Thomson Multimedia and/or Fraunhofer Gesellschaft nor imply any right to use the Licensed Materials in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit http://www.mp3licensing.com"

3. MPEG L.A., L.L.C./MPEG-2

USE OF THE APPLICABLE NVIDIA SOFTWARE PACKAGE IN ANY MANNER THAT COMPLIES WITH THE MPEG-2 STANDARD IS EXPRESSLY PROHIBITED WITHOUT A LICENSE UNDER APPLICABLE PATENTS IN THE MPEG-2 PATENT PORTFOLIO, WHICH LICENSE IS AVAILABLE FROM MPEG LA, L.L.C., 250 STREELE STREET, SUITE 300, DENVER, COLORADO 80206. NO LICENSE IS GRANTED HEREIN, BY IMPLICATION OR OTHERWISE, TO LICENSEE TO USE MPEG 2 INTERMEDIATE PRODUCTS MANUFACTURED OR SOLD BY LICENSEE.

4. MPEG-2 AAC

Licensee shall be solely responsible for either obtaining a valid and current license from AT&T Corp., Dolby Laboratories Licensing Corporation, Fraunhofer-Gesellschaft, and Sony Corporation for the applicable version of MPEG-2 AAC.

5. Fraunhofer-Gesellschaft MPEG-4 HE-AAC

Licensee understands and accepts that (a) it may be necessary to execute a patent license with the appropriate licensing entities in order to obtain all rights necessary to create Licensee's products; and (b) Licensee will contact the appropriate licensing entities, e.g. Via Licensing, and negotiate in good faith the adequate contracts, if any. In addition, it is hereby understood that in the event that, besides the Fraunhofer-Gesellschaft patents licensed through such appropriate licensing entities, any further Fraunhofer-Gesellschaft patent shall be required in order to use the Licensed Materials, Fraunhofer-Gesellschaft shall not request from Licensee any additional payment in order to receive a license to such further Fraunhofer-Gesellschaft patent, as long as Licensee remains a valid licensee of such appropriate licensing entity.

6. Microsoft Windows Media

Licensee acknowledges that Microsoft Windows Media is provided in object code form only, solely for the Licensee's own internal evaluation and testing purposes.

Licensee further acknowledges the following notice: "This product includes technology owned by Microsoft Corporation and cannot be used or further distributed without a license from Microsoft or a Microsoft affiliate."

7. Microsoft PlayReady or WMDRM technology

Licensee acknowledges that the Licensed Materials (i) contain a certain version of Microsoft PlayReady or WMDRM technology ("PlayReady Technology"); and (ii) are subject to certain intellectual property rights of Microsoft and cannot be used or distributed further without the appropriate license(s) from Microsoft.

Licensee represents and warrants that (i) Licensee holds a current and valid license under a PlayReady Device Agreement and Intermediated Product Distribution License, a PlayReady Final Product Distribution License, or a like agreement, with Microsoft or a Microsoft affiliate; and (ii) Licensee will use the PlayReady Technology provided under this Agreement in Licensees' software, hardware product, or service offering that (a) is intended for distribution to and/or use by end users; and (b) is in a final form with Licensee-owned brand and/or logo most prominently displayed brand in a fully functional user interface.

8. Ogg Vorbis Legal Information

Copyright (c) 2002, Xiph.org Foundation

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Xiph.org Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Exhibit A-2

Adobe Flash Demonstration and Evaluation License (“Adobe License”) Terms and Conditions

1. Software License. NVIDIA hereby grants to Licensee a non-exclusive, non-transferable, royalty free right to use, install, evaluate, test, demonstrate, publicly perform and display object code versions of the Adobe Software together with Licensee’s product. For the avoidance of doubt, Licensee has no right to distribute, sublicense, or otherwise commercialize the Adobe Software, unless and until Adobe confirms in writing that Licensee has entered into an appropriate license agreement with Adobe.

   “Adobe Software” means the Adobe Flash Player software version 10 as modified by NVIDIA, in object code form as delivered by NVIDIA hereunder. NVIDIA may update the Adobe Software from time to time, in its sole discretion.

2. Period. Licensee’s limited license hereunder shall commence on the Effective Date and remain in effect until terminated by NVIDIA upon five (5) days written notice.

3. Rights. Licensee agrees that it shall take no action in furtherance of seeking any patent rights or other intellectual property rights to the Adobe Software. The Software shall be returned to NVIDIA within fifteen (15) days of the end of the Evaluation Period. Licensee shall have no rights to sublicense or distribute the Software.

4. Fees. There shall be no fees owed by either party under this Adobe License.

5. Delivery. NVIDIA shall deliver the Adobe Software to Licensee shortly after execution of this Adobe License.

6. Restrictions. Licensee does not have any rights to make use of the Adobe Software, or in any manner, copy, disseminate, or in any way circulate the Adobe Software other than as permitted under Section 1 above. Licensee shall limit access to the Adobe Software to its employees who need to know such information and who have agreed, either as a condition to employment or prior to obtaining the Adobe Software, to be bound by terms and conditions of confidentiality. The rights herein do not entitle Licensee to use the Adobe Software, or any technology or intellectual property contained within it, as reference or inspiration for developing or creating another product in any way based upon the Adobe Software. Licensee agrees not to decompile, reverse engineer, reverse assemble, disassemble, or otherwise

reverse engineer or reduce the Adobe Software provided in object code form to a human-perceivable form.

7. Ownership. All right, title, and interest in the Adobe Software, shall be owned by Adobe. Except as set forth in Section 1 above, Licensee acquires no license to any NVIDIA or Adobe intellectual property rights pursuant to this Adobe License. The Adobe Software, and any partial or whole copies thereof, and all copyright, patent, trade secret and other intellectual property rights therein, are and remain the property of Adobe and NVIDIA. The provisions of this paragraph shall survive expiration or earlier termination of this Adobe License. NVIDIA does not directly or indirectly grant, or purport to grant, to Licensee any rights or immunities under Adobe's intellectual property rights that will subject such intellectual property rights to an open source license or scheme in which there is or could be interpreted to be a requirement that as a condition of use, modification and/or distribution, the Adobe Software be: (i) disclosed or distributed in source code form; (ii) licensed for the purpose of making derivative works; or (iii) redistributable at no charge.

8. Effect of Termination. Upon termination, the rights granted hereunder shall cease and all materials furnished to Licensee by NVIDIA hereunder relating to the Adobe Software shall be returned to it promptly, together with any copies thereof.

9. Disclaimer. NVIDIA PROVIDES THE ADOBE SOFTWARE "AS IS" AND WITHOUT ANY WARRANTIES. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE ADOBE SOFTWARE IS ASSUMED BY LICENSEE. NVIDIA DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, WITH REGARD TO THE ADOBE SOFTWARE OR ANY OTHER INFORMATION PROVIDED HEREUNDER, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

10. Limitation on Liability. NOTWITHSTANDING ANY PROVISION IN THIS AGREEMENT, NEITHER NVIDIA NOR ADOBE SHALL BE LIABLE TO LICENSEE OR ANY THIRD PARTY FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATED TO THIS AGREEMENT, INCLUDING, WITHOUT LIMITATION, DAMAGES RESULTING FROM LOSS OF PROFITS, DATA, BUSINESS, OR GOODWILL, HOWEVER CAUSED AND ON WHATEVER THEORY, WHETHER BASED ON BREACH OF CONTRACT OR WARRANTY, TORT (INCLUDING NEGLIGENCE), THE FAILURE OR ASSERTED FAILURE OF NVIDIA TO PERFORM ITS OBLIGATIONS HEREUNDER, OR OTHERWISE, AND WHETHER OR NOT NVIDIA HAS BEEN ADVISED OR

IS AWARE OF THE POSSIBILITY OF SUCH DAMAGES. NVIDIA's aggregate liability to Licensee or any third party arising out of or in connection with this Adobe License or any collateral agreement, whether in contract, tort (including negligence), or otherwise, shall be limited to fifty dollars (US$50).

11. Relief. As the unauthorized distribution of the Adobe Software may diminish the value to NVIDIA or Adobe of the proprietary interests that are the subject of this Adobe License, if Licensee breaches any of its obligations under this Adobe License, NVIDIA or Adobe shall be entitled to seek equitable relief to protect its interests therein, including but not limited to injunctive relief, as well as money damages.

12. Export Restrictions. The parties acknowledge that the Adobe Software is subject to U.S. export control laws and regulations. The parties agree to comply with all applicable international and national laws that apply to the Adobe Software, including the U.S. Export Administration Regulations and the United States Department of Commerce, as well as end-user, end-use and destination restrictions issued by U.S. and other governments.

13. Evaluation Feedback.

13.1 Feedback by Licensee. You must provide to NVIDIA any suggestions, comments and feedback regarding the Adobe Software ("Licensee Feedback"). NVIDIA and Adobe may use and include any Licensee Feedback that you provide to improve the Software or other technologies and / or products. Accordingly, you grant to NVIDIA, its subsidiaries, its affiliates and its licensees a perpetual, irrevocable, worldwide, royalty-free, fully paid-up license to freely use, have used, sell, modify, reproduce, transmit, license, sublicense (through multiple tiers of sublicensees, including to Adobe), distribute (through multiple tiers of distributors), and otherwise commercialize the Licensee Feedback in the Adobe Software or other NVIDIA or Adobe technologies and/or products.

13.2 Confidential Information. Licensee Feedback is considered Adobe's confidential information ("Adobe Confidential Information"). You shall not use or disclose any Adobe Confidential Information except as expressly authorized herein, and you shall protect all such Adobe Confidential Information using the same degree of care you use with respect to your own proprietary information, but in no event with safeguards less than a reasonably prudent business would exercise under similar circumstances. You agree to take prompt and appropriate action to prevent unauthorized use or disclosure of any Adobe Confidential Information.

Exhibit E

(Open Source Portions)

Licensee agrees that the following terms and conditions shall apply to its use of certain portions (as referenced below) of the applicable software packages selected by the Licensee in connection with this Agreement. For the sake of clarity, Licensee agrees that the terms and conditions of the Agreement shall continue to govern Licensee's use of the Software and Licensed Materials. The parties agree that the capitalized terms used in this exhibit shall have the same meaning ascribed to such term in the Agreement or any amendment thereto.

1. NVIDIA agrees that the open source portions expressly licensed under terms and conditions of Excluded Licenses (collectively the "Open Source Portions"), shall not be subject to the restrictions set forth in the following section ("No Excluded Licenses") of the Agreement (or substantially similar provision in the Agreement signed by Licensee):

   "3.5 No Excluded Licenses. The licenses granted in Section 2.1 do not include the right to, and Licensee shall not: (a) create Derivative Work(s) of the Licensed Materials in any manner that would cause the Licensed Materials, in whole or in part, to become subject to the terms of an Excluded License; or (b) distribute the Licensed Materials (or Derivative Works thereof) in any manner that would cause the Licensed Materials, or any component thereof, to become subject to the terms of an Excluded License."

2. Licensee agrees that it shall not externally distribute, license or otherwise disclose in any manner the Open Source Portions until the later of (a) the Licensee Products (as defined in the Agreement), that incorporates the Open Source Portions, in whole or in part, is commercialized and made generally available for sale; or (b) NVIDIA makes generally available to the public the Open Source Portions in source code form.

3. Except as noted otherwise in this Exhibit E, the terms and conditions of this Exhibit E will supercede any conflicting terms and conditions between Exhibit E and the Agreement.

EXHIBIT F

Licensee acknowledges and agrees with this following third party licensing obligations and/or notices in connection with its use of (a) Tegra Linux Driver Package; and (b) Chromium:

1. GNU General Public License 2.0

(For notice purposes only)

This product includes copyrighted third-party software licensed under the terms of the GNU General Public License. All third-party software packages are copyright by their respective authors. GNU General Public License is hereby incorporated into the Agreement by this reference.

http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt

2. Apache License v2.0

(For notice purposes only)

This product includes copyrighted third-party software licensed under the terms of the Apache License. All third-party software packages are copyright by their respective authors. Apache License is hereby incorporated into the Agreement by this reference.

http://www.apache.org/licenses/LICENSE-2.0.html

3. BSD License

(For notice purposes only)

This product includes copyrighted third-party software licensed under the terms of the BSD License. All third-party software packages are copyright by their respective authors. BSD License is incorporated into the Agreement by this reference.

http://www.opensource.org/licenses/bsd-license.php

4. MIT License

(For notice purposes only)

This product includes copyrighted third-party software licensed under the terms of the MIT License. All third-party software packages are copyright by their respective authors. MIT License is hereby incorporated into the Agreement by this reference http://www.opensource.org/licenses/mit-license.php

GST OpenMAX

The software listed below is licensed under the terms of the LGPLv2.1 (see below). To obtain source code, contact oss-requests@nvidia.com.

gst-openmax (libgstomx.so and libgstnvxvimagesink.so)

Version 2.1, February 1999

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights. We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library. To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.) Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications. You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate

distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY

APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.
>
> <signature of Ty Coon>, 1 April 1990
>
> Ty Coon, President of Vice

That's all there is to it!

Linux Kernel

The Linux kernel in this release is licensed under the terms of the GPLv2 (see below). The revision of Linux kernel source code used to build this binary can be retrieved by running the 'source_sync.sh' script or by request to oss-requests@nvidia.com.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit

royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all

derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

> Copyright (C) <year> <name of author>
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) year name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> <signature of Ty Coon>, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

mkbootimg and mkubootscript

mkbootimg and mkubootscript are provided under the following terms:

Copyright 2007, The Android Open Source Project

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the

Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (C) 1995, 1996, 1997, and 1998 WIDE Project.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

FIPS pub 180-1: Secure Hash Algorithm (SHA-1)
based on: http://csrc.nist.gov/fips/fip180-1.txt
implemented by Jun-ichiro itojun Itoh <itojun@itojun.org>

mkyaffs2img

The software listed below is licensed under the terms of the GPLv2 (see below). To obtain source code, contact oss-requests@nvidia.com.

mkyaffs2img

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
>
> b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
>
> c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

> a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
>
> b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
>
> c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or

rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS

PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.>
> Copyright (C) <year> <name of author>
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc.,
> 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) year name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> <signature of Ty Coon>, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT

NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2007, The Android Open Source Project

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by

such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the

Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

U-Boot and mkimage

U-Boot is Free Software. It is copyrighted by Wolfgang Denk and many others who contributed code (see the actual source code for details). You can redistribute U-Boot and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation. Most of it can also be distributed, at your option, under any later version of the GNU General Public License -- see individual files for exceptions.

NOTE! This license does *not* cover the so-called "standalone" applications that use U-Boot services by means of the jump table provided by U-Boot exactly for this purpose - this is merely considered normal use of U-Boot, and does *not* fall under the heading of "derived work".

The header files "include/image.h" and "include/asm-*/u-boot.h" define interfaces to U-Boot. Including these (unmodified) header files in another file is considered normal use of U-Boot, and does*not* fall under the heading of "derived work".

Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the U-Boot source code) is copyrighted by me and others who actually wrote it.-- Wolfgang Denk

==

GNU GENERAL PUBLIC LICENSE, Version 2, June 1991

For the complete text of this license, see GNU GENERAL PUBLIC LICENSE in this section.

Appendix: Example Crosstool-ng Configuration File

The following is a listing of an example .config file for the Crosstool-ng toolchain component. For more information see the Building Crosstool-ng Toolchain and glibc chapter of this guide.

```
# Automatically generated make config: don't edit
# crosstool-NG hg+-11c23aa9c9f9 Configuration
# Tue Aug 21 15:05:23 2012
#
CT_CONFIGURE_has_xz=y
CT_CONFIGURE_has_cvs=y
CT_CONFIGURE_has_svn=y
CT_MODULES=y

#
# Paths and misc options
#

#
# crosstool-NG behavior
#
# CT_OBSOLETE is not set
CT_EXPERIMENTAL=y
# CT_DEBUG_CT is not set

#
# Paths
#
CT_LOCAL_TARBALLS_DIR="${TOP_DIR}/crosstool-ng/src"
CT_SAVE_TARBALLS=y
CT_WORK_DIR="${TOP_DIR}/crosstool-ng/work"
CT_PREFIX_DIR="${TOP_DIR}/crosstool-ng/install"
CT_INSTALL_DIR="${CT_PREFIX_DIR}"
CT_RM_RF_PREFIX_DIR=y
CT_REMOVE_DOCS=y
CT_INSTALL_DIR_RO=y
CT_STRIP_ALL_TOOLCHAIN_EXECUTABLES=y

#
# Downloading
#
# CT_FORBID_DOWNLOAD is not set
# CT_FORCE_DOWNLOAD is not set
```

```
CT_CONNECT_TIMEOUT=10
# CT_ONLY_DOWNLOAD is not set
# CT_USE_MIRROR is not set

#
# Extracting
#
# CT_FORCE_EXTRACT is not set
CT_OVERIDE_CONFIG_GUESS_SUB=y
# CT_ONLY_EXTRACT is not set
CT_PATCH_BUNDLED=y
# CT_PATCH_LOCAL is not set
# CT_PATCH_BUNDLED_LOCAL is not set
# CT_PATCH_LOCAL_BUNDLED is not set
# CT_PATCH_BUNDLED_FALLBACK_LOCAL is not set
# CT_PATCH_LOCAL_FALLBACK_BUNDLED is not set
# CT_PATCH_NONE is not set
CT_PATCH_ORDER="bundled"

#
# Build behavior
#
CT_PARALLEL_JOBS=1
CT_LOAD=0
CT_USE_PIPES=y
CT_EXTRA_FLAGS_FOR_HOST=""
# CT_CONFIG_SHELL_SH is not set
# CT_CONFIG_SHELL_ASH is not set
CT_CONFIG_SHELL_BASH=y
# CT_CONFIG_SHELL_CUSTOM is not set
CT_CONFIG_SHELL="${bash}"

#
# Logging
#
# CT_LOG_ERROR is not set
# CT_LOG_WARN is not set
# CT_LOG_INFO is not set
CT_LOG_EXTRA=y
# CT_LOG_ALL is not set
# CT_LOG_DEBUG is not set
CT_LOG_LEVEL_MAX="EXTRA"
# CT_LOG_SEE_TOOLS_WARN is not set
CT_LOG_PROGRESS_BAR=y
CT_LOG_TO_FILE=y
CT_LOG_FILE_COMPRESS=y

#
# Target options
#
CT_ARCH="arm"
CT_ARCH_SUPPORTS_BOTH_MMU=y
CT_ARCH_SUPPORTS_BOTH_ENDIAN=y
CT_ARCH_SUPPORTS_32=y
CT_ARCH_SUPPORTS_WITH_ARCH=y
CT_ARCH_SUPPORTS_WITH_CPU=y
CT_ARCH_SUPPORTS_WITH_TUNE=y
```

```
CT_ARCH_SUPPORTS_WITH_FLOAT=y
CT_ARCH_SUPPORTS_WITH_FPU=y
CT_ARCH_SUPPORTS_SOFTFP=y
CT_ARCH_DEFAULT_HAS_MMU=y
CT_ARCH_DEFAULT_LE=y
CT_ARCH_DEFAULT_32=y
CT_ARCH_ARCH="armv7-a"
CT_ARCH_CPU="cortex-a9"
CT_ARCH_TUNE="cortex-a9"
CT_ARCH_FPU=""
# CT_ARCH_BE is not set
CT_ARCH_LE=y
CT_ARCH_32=y
CT_ARCH_BITNESS=32
CT_ARCH_FLOAT_HW=y
# CT_ARCH_FLOAT_SW is not set
CT_TARGET_CFLAGS=""
CT_TARGET_LDFLAGS=""
# CT_ARCH_alpha is not set
CT_ARCH_arm=y
# CT_ARCH_avr32 is not set
# CT_ARCH_blackfin is not set
# CT_ARCH_m68k is not set
# CT_ARCH_mips is not set
# CT_ARCH_powerpc is not set
# CT_ARCH_s390 is not set
# CT_ARCH_sh is not set
# CT_ARCH_sparc is not set
# CT_ARCH_x86 is not set
CT_ARCH_alpha_AVAILABLE=y
CT_ARCH_arm_AVAILABLE=y
CT_ARCH_avr32_AVAILABLE=y
CT_ARCH_blackfin_AVAILABLE=y
CT_ARCH_m68k_AVAILABLE=y
CT_ARCH_mips_AVAILABLE=y
CT_ARCH_powerpc_AVAILABLE=y
CT_ARCH_s390_AVAILABLE=y
CT_ARCH_sh_AVAILABLE=y
CT_ARCH_sparc_AVAILABLE=y
CT_ARCH_x86_AVAILABLE=y

#
# Generic target options
#
# CT_MULTILIB is not set
CT_ARCH_USE_MMU=y
CT_ARCH_ENDIAN="little"

#
# Target optimisations
#
# CT_ARCH_FLOAT_SOFTFP is not set
CT_ARCH_FLOAT="hard"

#
# arm other options
#
```

```
CT_ARCH_ARM_MODE="arm"
CT_ARCH_ARM_MODE_ARM=y
# CT_ARCH_ARM_MODE_THUMB is not set
# CT_ARCH_ARM_INTERWORKING is not set
CT_ARCH_ARM_EABI=y

#
# Toolchain options
#

#
# General toolchain options
#
CT_FORCE_SYSROOT=y
CT_USE_SYSROOT=y
CT_SYSROOT_NAME="sysroot"
CT_SYSROOT_DIR_PREFIX=""
CT_WANTS_STATIC_LINK=y
CT_STATIC_TOOLCHAIN=y
CT_TOOLCHAIN_PKGVERSION=""
CT_TOOLCHAIN_BUGURL=""

#
# Tuple completion and aliasing
#
CT_TARGET_VENDOR="cortex_a9"
CT_TARGET_ALIAS_SED_EXPR=""
CT_TARGET_ALIAS=""

#
# Toolchain type
#
# CT_NATIVE is not set
CT_CROSS=y
# CT_CROSS_NATIVE is not set
# CT_CANADIAN is not set
CT_TOOLCHAIN_TYPE="cross"

#
# Build system
#
CT_BUILD=""
CT_BUILD_PREFIX=""
CT_BUILD_SUFFIX=""

#
# Misc options
#
# CT_TOOLCHAIN_ENABLE_NLS is not set

#
# Operating System
#
CT_KERNEL_SUPPORTS_SHARED_LIBS=y
CT_KERNEL="linux"
CT_KERNEL_VERSION="2.6.36.4"
# CT_KERNEL_bare_metal is not set
```

```
CT_KERNEL_linux=y
CT_KERNEL_bare_metal_AVAILABLE=y
CT_KERNEL_linux_AVAILABLE=y
# CT_KERNEL_V_3_5 is not set
# CT_KERNEL_V_3_4_7 is not set
# CT_KERNEL_V_3_3_8 is not set
# CT_KERNEL_V_3_2_25 is not set
# CT_KERNEL_V_3_1_10 is not set
# CT_KERNEL_V_3_0_39 is not set
# CT_KERNEL_V_2_6_39_4 is not set
# CT_KERNEL_V_2_6_38_8 is not set
# CT_KERNEL_V_2_6_37_6 is not set
CT_KERNEL_V_2_6_36_4=y
# CT_KERNEL_V_2_6_33_20 is not set
# CT_KERNEL_V_2_6_32_59 is not set
# CT_KERNEL_V_2_6_31_14 is not set
# CT_KERNEL_V_2_6_27_62 is not set
# CT_KERNEL_LINUX_CUSTOM is not set
CT_KERNEL_mingw32_AVAILABLE=y

#
# Common kernel options
#
CT_SHARED_LIBS=y

#
# linux other options
#
CT_KERNEL_LINUX_VERBOSITY_0=y
# CT_KERNEL_LINUX_VERBOSITY_1 is not set
# CT_KERNEL_LINUX_VERBOSITY_2 is not set
CT_KERNEL_LINUX_VERBOSE_LEVEL=0
CT_KERNEL_LINUX_INSTALL_CHECK=y

#
# Binary utilities
#
CT_ARCH_BINFMT_ELF=y

#
# GNU binutils
#
# CT_BINUTILS_V_2_22 is not set
# CT_BINUTILS_V_2_21_53 is not set
# CT_BINUTILS_V_2_21_1a is not set
CT_BINUTILS_V_2_20_1a=y
# CT_BINUTILS_V_2_19_1a is not set
# CT_BINUTILS_V_2_18a is not set
CT_BINUTILS_VERSION="2.20.1a"
CT_BINUTILS_2_20_or_later=y
CT_BINUTILS_2_19_or_later=y
CT_BINUTILS_2_18_or_later=y
CT_BINUTILS_HAS_HASH_STYLE=y
CT_BINUTILS_GOLD_SUPPORTS_ARCH=y
CT_BINUTILS_HAS_PKGVERSION_BUGURL=y
CT_BINUTILS_FORCE_LD_BFD=y
CT_BINUTILS_LINKER_LD=y
```

```
CT_BINUTILS_LINKERS_LIST="ld"
CT_BINUTILS_LINKER_DEFAULT="bfd"
CT_BINUTILS_EXTRA_CONFIG_ARRAY=""
# CT_BINUTILS_FOR_TARGET is not set

#
# C compiler
#
CT_CC="gcc"
CT_CC_VERSION="4.5.3"
CT_CC_gcc=y
# CT_CC_GCC_SHOW_LINARO is not set
# CT_CC_V_4_7_1 is not set
# CT_CC_V_4_7_0 is not set
# CT_CC_V_4_6_3 is not set
# CT_CC_V_4_6_2 is not set
# CT_CC_V_4_6_1 is not set
# CT_CC_V_4_6_0 is not set
CT_CC_V_4_5_3=y
# CT_CC_V_4_5_2 is not set
# CT_CC_V_4_5_1 is not set
# CT_CC_V_4_5_0 is not set
# CT_CC_V_4_4_7 is not set
# CT_CC_V_4_4_6 is not set
# CT_CC_V_4_4_5 is not set
# CT_CC_V_4_4_4 is not set
# CT_CC_V_4_4_3 is not set
# CT_CC_V_4_4_2 is not set
# CT_CC_V_4_4_1 is not set
# CT_CC_V_4_4_0 is not set
# CT_CC_V_4_3_6 is not set
# CT_CC_V_4_3_5 is not set
# CT_CC_V_4_3_4 is not set
# CT_CC_V_4_3_3 is not set
# CT_CC_V_4_3_2 is not set
# CT_CC_V_4_3_1 is not set
# CT_CC_V_4_2_4 is not set
# CT_CC_V_4_2_2 is not set
CT_CC_GCC_4_2_or_later=y
CT_CC_GCC_4_3_or_later=y
CT_CC_GCC_4_4_or_later=y
CT_CC_GCC_4_5=y
CT_CC_GCC_4_5_or_later=y
CT_CC_GCC_HAS_GRAPHITE=y
CT_CC_GCC_HAS_LTO=y
CT_CC_GCC_HAS_PKGVERSION_BUGURL=y
CT_CC_GCC_HAS_BUILD_ID=y
CT_CC_GCC_USE_GMP_MPFR=y
CT_CC_GCC_USE_MPC=y
CT_CC_GCC_USE_LIBELF=y
# CT_CC_LANG_FORTRAN is not set
CT_CC_SUPPORT_CXX=y
CT_CC_SUPPORT_FORTRAN=y
CT_CC_SUPPORT_JAVA=y
CT_CC_SUPPORT_ADA=y
CT_CC_SUPPORT_OBJC=y
CT_CC_SUPPORT_OBJCXX=y
```

```
#
# Additional supported languages:
#
CT_CC_LANG_CXX=y
# CT_CC_LANG_JAVA is not set
# CT_CC_LANG_ADA is not set
# CT_CC_LANG_OBJC is not set
# CT_CC_LANG_OBJCXX is not set
CT_CC_LANG_OTHERS=""

#
# gcc other options
#
CT_CC_ENABLE_CXX_FLAGS=""
CT_CC_CORE_EXTRA_CONFIG_ARRAY="--with-float=hard"
CT_CC_EXTRA_CONFIG_ARRAY="--with-float=hard"
CT_CC_STATIC_LIBSTDCXX=y
# CT_CC_GCC_SYSTEM_ZLIB is not set

#
# Optimisation features
#
# CT_CC_GCC_USE_GRAPHITE is not set
CT_CC_GCC_USE_LTO=y

#
# Settings for libraries running on target
#
CT_CC_GCC_ENABLE_TARGET_OPTSPACE=y
# CT_CC_GCC_LIBMUDFLAP is not set
# CT_CC_GCC_LIBGOMP is not set
# CT_CC_GCC_LIBSSP is not set

#
# Misc. obscure options.
#
CT_CC_CXA_ATEXIT=y
# CT_CC_GCC_DISABLE_PCH is not set
CT_CC_GCC_SJLJ_EXCEPTIONS=m
CT_CC_GCC_LDBL_128=m
# CT_CC_GCC_BUILD_ID is not set

#
# C-library
#
CT_LIBC="glibc"
CT_LIBC_VERSION="2.11"
# CT_LIBC_eglibc is not set
CT_LIBC_glibc=y
# CT_LIBC_uClibc is not set
CT_LIBC_eglibc_AVAILABLE=y
CT_LIBC_glibc_AVAILABLE=y
CT_LIBC_GLIBC_TARBALL=y
# CT_LIBC_GLIBC_V_2_14_1 is not set
# CT_LIBC_GLIBC_V_2_14 is not set
# CT_LIBC_GLIBC_V_2_13 is not set
# CT_LIBC_GLIBC_V_2_12_2 is not set
# CT_LIBC_GLIBC_V_2_12_1 is not set
# CT_LIBC_GLIBC_V_2_11_1 is not set
CT_LIBC_GLIBC_V_2_11=y
# CT_LIBC_GLIBC_V_2_10_1 is not set
# CT_LIBC_GLIBC_V_2_9 is not set
# CT_LIBC_GLIBC_V_2_8 is not set
CT_LIBC_mingw_AVAILABLE=y
CT_LIBC_newlib_AVAILABLE=y
CT_LIBC_none_AVAILABLE=y
CT_LIBC_uClibc_AVAILABLE=y
CT_LIBC_SUPPORT_THREADS_ANY=y
CT_LIBC_SUPPORT_NPTL=y
CT_THREADS="nptl"

#
# Common C library options
#
CT_THREADS_NPTL=y
CT_LIBC_XLDD=y
CT_LIBC_GLIBC_MAY_FORCE_PORTS=y
CT_LIBC_glibc_familly=y
CT_LIBC_GLIBC_EXTRA_CONFIG_ARRAY=""
CT_LIBC_GLIBC_CONFIGPARMS=""
CT_LIBC_GLIBC_EXTRA_CFLAGS=""
CT_LIBC_EXTRA_CC_ARGS=""
# CT_LIBC_ENABLE_FORTIFIED_BUILD is not set
# CT_LIBC_DISABLE_VERSIONING is not set
CT_LIBC_OLDEST_ABI=""
CT_LIBC_GLIBC_FORCE_UNWIND=y
CT_LIBC_GLIBC_USE_PORTS=y
CT_LIBC_ADDONS_LIST=""
# CT_LIBC_LOCALES is not set
# CT_LIBC_GLIBC_KERNEL_VERSION_NONE is not set
CT_LIBC_GLIBC_KERNEL_VERSION_AS_HEADERS=y
# CT_LIBC_GLIBC_KERNEL_VERSION_CHOSEN is not set
CT_LIBC_GLIBC_MIN_KERNEL="2.6.36.4"

#
# glibc other options
#

#
# WARNING !!!
#

#
#   For glibc >= 2.8, it can happen that the tarballs
#

#
#   for the addons are not available for download.
#

#
#   If that happens, bad luck... Try a previous version
#
```

```
#
#   or try again later... :-(
#

#
# Debug facilities
#
# CT_DEBUG_dmalloc is not set
# CT_DEBUG_duma is not set
# CT_DEBUG_gdb is not set
# CT_DEBUG_ltrace is not set
# CT_DEBUG_strace is not set

#
# Companion libraries
#
CT_COMPLIBS_NEEDED=y
CT_GMP_NEEDED=y
CT_MPFR_NEEDED=y
CT_MPC_NEEDED=y
CT_LIBELF_NEEDED=y
CT_COMPLIBS=y
CT_GMP=y
CT_MPFR=y
CT_MPC=y
CT_LIBELF=y
# CT_GMP_V_5_0_2 is not set
# CT_GMP_V_5_0_1 is not set
CT_GMP_V_4_3_2=y
# CT_GMP_V_4_3_1 is not set
# CT_GMP_V_4_3_0 is not set
CT_GMP_VERSION="4.3.2"
# CT_MPFR_V_3_1_0 is not set
# CT_MPFR_V_3_0_1 is not set
# CT_MPFR_V_3_0_0 is not set
CT_MPFR_V_2_4_2=y
# CT_MPFR_V_2_4_1 is not set
# CT_MPFR_V_2_4_0 is not set
CT_MPFR_VERSION="2.4.2"
# CT_MPC_V_0_9 is not set
# CT_MPC_V_0_8_2 is not set
CT_MPC_V_0_8_1=y
# CT_MPC_V_0_7 is not set
CT_MPC_VERSION="0.8.1"
CT_LIBELF_V_0_8_13=y
# CT_LIBELF_V_0_8_12 is not set
CT_LIBELF_VERSION="0.8.13"

#
# Companion libraries common options
#
# CT_COMPLIBS_CHECK is not set

#
# Companion tools
#
```

```
#
# READ HELP before you say 'Y' below !!!
#
# CT_COMP_TOOLS is not set

#
# Test suite
#
# CT_TEST_SUITE_GCC is not set
```

FAQs

How do I use a 64-bit Ubuntu host system with NVIDIA® Tegra® Linux Driver Package?

If you are running a 64-bit Ubuntu installation on the host PC, this release requires that you have 32-bit runtime support installed as well. The simple steps for installing 32-bit support on an Ubuntu host use the apt-get command.

To install 32-bit runtime support

- Execute the following commands:

  ```
  $ sudo apt-get update
  $ sudo apt-get install ia32-libs
  ```

  Note: If ia32-libs is not installed, flashing the boot loader with the kernel (and, possibly, other steps in the process) will fail.

How do I use display mode and resolution configuration with the X RandR application?

You can use the X Resize, Rotate and Reflect Extension (RandR) extension to manipulate and configure the attached displays (both the internal panel and any externally connected HDMI® panel). The `xrandr(1)` utility is the most common way to do this.

A tutorial on xrandr can be found on the following website:

`http://www.thinkwiki.org/wiki/Xorg_RandR_1.2`

Are there generated ssh host keys for the sample file system?

There are no keys in the `/etc/ssh` directory of the provided sample file system. For information about creating the ssh host keys, see the `ssh-keygen` man page. For more information, see the About the Root File System topic in the Getting Started chapter.

How do I determine the X driver ABI of the X server used in the root file system?

All `tegra_drv.abi*.so` files are in the driver package. By default the `apply_binaries.sh` script creates a sym-link from `tegra_drv.so` to the X ABI driver compatible with the provided sample file system. For more information, see the Setting Up the Root File System topic in the Getting Started chapter.

How do I install Ubuntu Desktop?

The default sample file-system that is provided with the release was created with minimal packages pre-installed. For more information on which packages are present, see the Setting Up Your File System topic in the Getting Started chapter.

Please note the possible space constraints with the default partition size specified in the `flash.sh` that is referenced in this document. To increase the partition size so that it can accommodate additional packages, see the Increasing the Internal Memory Partition Size for the Root File System topic in the Getting Started chapter.

To install the full Ubuntu desktop

1. First follow the instructions from the Installing Additional Packages topic in the Getting Started chapter.
2. Execute the following command:

   ```
   $ sudo apt-get install ubuntu-desktop
   ```

   Note: For the default username and password, see the About the Root File System topic in the Getting Started chapter.

Can I install a different X driver Application Binary Interface (ABI)?

Yes. For more information, see the Setting Up the Root File System topic in the Getting Started chapter.

How do I disable the Ubuntu firewall which can block NFS root access?

See the Configuring NFS Root on the Linux Host topic in the Getting Started chapter.

What is the sample rootfs log-in password?

The username is `ubuntu` and the password `ubuntu`. For more information about the sample file system, see the Setting Up Your File System topic in the Getting Started chapter.

How do I prevent the system display from blanking out?

Linux kernel 3.1 (and later) adds a power saving feature that may blank the display of an idle system even when applications are running. The feature is called console blank (screen saver). It is defined as:

```
consoleblank= [KNL]
```

Where `[KNL]` is the console blank (screen saver) timeout in seconds. This defaults to `10*60 = 10 mins`. A value of `0` disables the blank timer.

By passing arguments to the kernel command line, you can:

- Disable this feature, or
- Set the timeout to a longer interval.

With the `flash.sh` script, you can override the kernel command line options passed from fastboot to the kernel. For more information, see the Flash Script Usage topic in the Getting Started chapter.

To disable the console blank (screen saver) from the kernel command line

1. Add the following line to the kernel parameters in the grub configuration:
   ```
   consoleblank=0
   ```
2. View the current `consoleblank` value with the following command:
   ```
   $ cat /sys/module/kernel/parameters/consoleblank
   ```

To disable the console blank feature with an escape sequence

- Enter the following escape sequence:
  ```
  $ echo -ne "\033[9;0]"
  ```

To change the console blank timeout value with an escape sequence

- Enter the following escape sequence:
  ```
  $ echo -ne "\033[9;<timeout>]"
  ```

  where `<timeout>` is the timeout in seconds.

For more information on this escape sequence, see the `console_codes(4)` man page documents. For information on the input/output controls that provide some of the same functionality, see the `console_ioctl(4)` man page.

What is 'softfp' (also known as armel)?

Softfp means "software floating point". Softfp systems perform floating point instructions with software floating point instructions instead of with the full capabilities of the Floating Point Unit (FPU).

What is 'hardfp' (also known as armhf)?

Hardfp means "hardware floating point". Hardfp systems execute floating point instructions with full hardware floating-point support.

Are softfp binaries compatible with a hardfp distro? Are hardfp binaries compatible with a softfp distro?

No. Softfp binaries can only be used in a full softfp distro, and hardfp binaries can only be used in a full hardfp distro.

How do I prevent the file system from becoming read-only on resume if an SD card is in the Card Reader?

If the device resumes from Suspend (LP1) while an SD card is in a connected card reader device, the file system is mounted as read-only resulting in a card reader read/write error and a system reboot is required. The workaround is to install the udisks utility (The usdisks utility has a Debian package name similar to `udisks_1.0.2-4ubuntu_armel.deb` for the appropriate Ubuntu distro and softfp build. Hardfp packages are often labeled "armhf").

Can I use an SD card with a card reader already connected to the device?

If the user connects the card reader to the device before inserting the SD card into the card reader, the SD card is not detected. The SD card and card reader must be connected to the device simultaneously. The workaround is to install the udisks utility (The usdisks utility has a Debian package name similar to `udisks_1.0.2-4ubuntu_armel.deb` Hardfp packages are often labeled "armhf").

Glossary

[3] [4] [A] [B] [C] [D] [E] [F] [G] [H] [I] [J] [K] [L] [M] [N] [O] [P] [Q] [R] [S] [T] [U] [V] [W] [X] [Y] [Z]

3

3G

Third generation mobile phone standard/technology, based on standards defined by the International Telecommunication Union (ITU).

3G2

A standard for 3GP format for CDMA-based phones (3GPP2) and container format with filename extension (.3gp).

3GP

Simplified version of MPEG-4 Part 14 (.mp4) container format.

3GPP

3rd Generation Partnership Project. A collaboration among telecommunications associations to define globally applicable third generation (3G) mobile phone system specifications. For more information, see http://www.3gpp.org.

3P

Platform Programming Protocol, developed by NVIDIA for client-server communications between PC and device.

4

4CIF

4 x CIF (704 x 576), Common International Format (CIF) for horizontal and vertical resolutions of YCbCr.

A

A2DP

Advanced Audio Distribution Profile. For streaming stereo or mono audio from one device to another over Bluetooth. For more information, see http://www.atheros.com/.

AAC

Advanced Audio Coding. A lossy compression and encoding standard for digital audio.

AAC-LC

Advanced Audio Coding-Low Complexity. A standardized, lossy compression and encoding scheme for digital audio.

AAC+

Advanced Audio Coding Plus, or aacPlus. Same as High Efficiency AAC (HE-AAC), which extends the Low Complexity AAC (AAC LC) optimized for low-bit rate applications such as streaming audio.

ABI

Application Binary Interface. A low-level interface between applications and other applications or the operating system.

ADB

Android Debug Bridge. A client-server tool for managing an emulator instance or Android-based device. For more information, see http://developer.android.com/guide/developing/tools/adb.html.

ADMA

Advanced Direct Memory Access.

ADPCM

Adaptive DPCM (differential pulse-code modulation).

AE

Auto exposure.

AES

Advanced Encryption Standard.

AF

Auto focus.

AGC

Automatic gain control.

ALSA

Advanced Linux Sound Architecture.

AMR

Adaptive multi-rate. An audio data compression scheme optimized for speech coding.

AMR-NB

Adaptive multi-rate (AMR) narrow band.

AMR-WB

Adaptive multi-rate wide band.

ANR

In Android, “Application Not Responding” error.

In camera, advanced noise reduction.

AP

Application Processor. An application processor is a computer that processes data (as opposed to one that controls data flow, like a database server). The Tegra® series application processors offer low power, high performance ARM® processors that handle 2D, 3D, audio, and high-definition (HD) video data streams. These decoding and encoding functionalities are provided by a set of interfaces including multiple memory, storage, video, audio, and peripheral interfaces.

AVC

Advanced Video Coding.

AVI

Audio Video Interleave. A multimedia container format, special-case Resource Interchange File Format (RIFF) file that can contain both audio and video data; this format enables synchronous audio-with-video playback. For more information, see http://msdn.microsoft.com/en-us/library/ms779631(VS.85).aspx.

AWB

Container format for AMR-WB speech encoding with filename extension (.awb).

B

BCB

Boot Control Block.

BCT

NVIDIA Boot Configuration Table.

bitblt

A graphics operation that combines several bitmap patterns into one, typically using a raster operator.

Bpp

Bytes per pixel, used to specify pixel depth (color depth).

bpp

Bits per pixel, used to specify pixel depth (color depth).

Bluetooth

Wireless standard for data exchange over short distances. For more information, see http://www.bluetooth.com/English/Pages/default.aspx.

BSAC

Bit Sliced Arithmetic Coding. An MPEG-4 standard (ISO/IEC 14496-3 subpart 4) for scalable audio coding.

BusyBox

Utility providing small versions of common UNIX utilities in a single executable. For more information, see http://www.busybox.net.

C

CABAC

Context-adaptive binary arithmetic coding. A type of entropy coding used in H.264/MPEG-4 AVC video encoding.

CBR

Constant bit rate.

CDC

USB Communications Device Class.

CDMA

Code division multiple access. Channel access method for radio communication.

CE

NVIDIA customer engineer.

Cg

C for Graphics. A high-level shading language for programming vertex and pixel shaders, created by NVIDIA Corporation.

CIF

Common International Format (352 x 288), standardizes horizontal/vertical resolutions for video.

color space

Specifies how color is represented, such as YUV, RGB, or gray scale.

D

D3DM

Microsoft Direct3D Mobile technologies.

DCC

Debug communications channel.

DCT

Discrete cosine transform. A Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers.

DDI

Device driver interface for Windows CE.

DDK

NVIDIA® Driver Development Kit.

deprecated

This feature is slated to be removed at a later release. Developers should begin to remove dependencies on this feature in preparation for its eventual removal.

development system

Board with NVIDIA® Tegra® processor used to do engineering work, which is typically focused on firmware/software development. Development boards have a user manual but may or may not include detailed documents, like schematics.

DFS

Dynamic frequency scaling.

DIDIM

Dynamic Image-based Display Intensity Modulation, which has been renamed pixel rendering intensity and saturation management (PRISM) since CES 2012.

DivX

Codec by DivX, Inc., that uses lossy MPEG-4 Part 2 compression to compress lengthy video into small sizes with high visual quality and is often used for "ripping". For more information, see http://www.divx.com.

DMO

Microsoft DirectX Media Object. For more information, see http://msdn2.microsoft.com/en-us/library/ms783356.aspx.

DPB

In H.264, Decode Picture Buffer.

DRC

Dynamic range compression.

DSI

Display Serial Interface a communication protocol specification by the Mobile Industry Processor Interface (MIPI) Alliance for reducing cost of displays in mobile devices.

DVB-H

Digital video broadcasting—handheld.

DVB-T

Digital video broadcasting—terrestrial.

DVFS

Dynamic voltage frequency scaling.

DVS

Dynamic voltage scaling.

E

eAAC+

Enhanced AAC+. Combines HE-AAC v1 (or AAC+) coupled with Parametric Stereo to 3GPP.

ECI

NVIDIA Embedded Controller Interface. Communication interface between NVIDIA® Tegra® processor and an embedded controller (EC) for netbook/smartbook applications.

EGL

Embedded-Systems Graphics Library. For OpenGL ES.

eMMC

Embedded MMC. Developed by JEDEC and MMCA for embedded flash memory applications.

EQ

Equalizer.

Escape code base + value

Microsoft supports definition of additional driver-specific escape codes, starting at an `ESCAPECODEBASE` of decimal value 100,000. So an NVIDIA-defined escape code whose value is 7 is actually 100007. (`100000 + 7 = 100007`)

Exif

Exchangeable image file format. A specification for digital camera image file formats.

Ext2

Second extended file system for the Linux kernel, designed to replace the extended file system (ext).

Ext3

Third extended file system. A journaling file system often used by the Linux kernel, the default file system for some distributions.

Ext4

Fourth extended file system. A journaling file system often used by the Linux kernel. It is the successor to Ext3.

F

Fastboot

Software supporting a protocol for updating flash file systems and unsigned partition images for Android-based devices.

Flash 11

Adobe multimedia platform enabling animation and interactivity on Web pages. For more information, see http://get.adobe.com/flashplayer.

FMO

Flexible macroblock ordering. Technique for restructuring the ordering of the representation of the fundamental regions in pictures, known as macroblocks. FMO is also referred to as slice groups and arbitrary slice ordering (ASO).

FOV

In photography, field of view.

G

GLES

See OpenGL ES.

GLSL

OpenGL Shading Language. A high level, C-language shading language.

GPIO

General purpose input/output. This is a generic pin on a chip whose behavior can be controlled with software.

GPS

Global positioning system.

GPU

Graphics processing unit.

H

H.263

A video codec standard for low-bit rate compressed format videoconferencing, designed by the ITU-T in a project ending in 1995/1996. For more information, see http://en.wikipedia.org/wiki/H.263.

H.264

A standard for video compression, also known as MPEG-4 Part 10, or AVC (for Advanced Video Coding). For more information, http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC.

HD

High-definition.

HDCP

High-bandwidth Digital Content Protection. Digital copy protection technology developed by Intel Corporation to protect digital audio and video content as it travels across connections. For more information, see http://www.digital-cp.com.

HDMI

High-Definition Multimedia Interface. A compact audio/video connector interface used to connect HDMI-enabled digital audio devices for transmitting uncompressed digital streams. NVIDIA® Tegra® BSP incorporates support for HDMI® technology.

HID

Human interface device. A computer device that receives human input and may deliver output.

HSMMC

High-speed MultiMediaCard (MMC).

HTTP

Hypertext transfer protocol. A client-server communications protocol used for hyperlinked text documents on the Internet.

I

I2C

Inter-Integrated Circuit. A serial computer bus used to attach low-speed peripherals to an embedded system or cell phone.

I2S

Inter-IC Sound (or Integrated Interchip Sound). A serial bus interface standard for connecting to digital audio devices.

ID3

Metadata container typically used with MP3 formatted content.

IIR

Infinite impulse response, a property of signal processing systems.

ISDB-T

Terrestrial Integrated Services Digital Broadcasting.

ISP

File extension for NVIDIA Image Signal Processing pipeline (.isp) configuration files.

ISV

Independent software vendor.

J

JPEG

Method for compressing photographic images. For more information, see http://www.jpeg.org.

JTAG

Joint Test Action Group (JTAG). Common term used for the IEEE 1149.1 standard “Standard Test Access Port and Boundary-Scan Architecture” for testing printed circuit boards. In embedded development, in-circuit emulators use JTAG as a transport mechanism to provide a way into the embedded system for debugging.

K

Kconfig

Linux kernel configuration files, which are present in almost each directory. Kconfig syntax is documented in the `Documentation/kbuild/kconfig-language.txt` file.

L

LBR

Low bit rate.

LCD

Liquid crystal display.

LP

Low power, or low power filter bank.

M

M4A

Multimedia MPEG-4 container format file extension (.m4a), first popularized by Apple to assure presence of audio/video content as distinguished from .mp4 files which may or may not have video content.

M4B

Multimedia MPEG-4 container format file extension (.m4b) for audio book and podcast files. Typically contain metadata for chapters, images, and hyperlinks.

Meebo

An instant messaging program based on Ajax and libpurple free/open source library. For more information, see http://www.meebo.com and http://www.pidgin.im.

MIDI

Musical instrument digital interface. For synchronization of electronic musical instrument and computer communications of digital data events (such as for pitch and volume) in real time.

MIO

Modular input/output. Enables adding peripheral cards to laser printers. For more information, see http://www.hp.com/.

MJPEG

Motion JPEG (M-JPEG) are video formats where video frames/ interlaced fields in digital video is compressed separately as a JPEG image.

MLC

Multilevel cell. Flash memory that stores more than one bit per cell by using voltage levels.

MMC

MultiMediaCard. Removable solid-state memory card for use in mobile devices. For more information, see http://en.wikipedia.org/wiki/MultiMediaCard.

MOV

File format for QuickTime that functions as a multimedia container file containing one or multiple tracks that stores audio, video, effects, or text.

moviNAND

High-density MLC NAND Flash combined with MMC controller.

MP

Megapixel.

MP3

MPEG-1 Audio Layer 3. Also the container format or filename extension (.mp3) for MPEG-1 Audio Layer 3 files.

MP4

Container format or filename extension (.mp4) for MPEG-4 Part 14 files.

MPEG-2

Generic coding standard for movies, which specifies a combination of lossy video compression and lossy audio compression (audio data compression).

MPEG-4

MPEG-4 Part 2 video compression technology. A DCT compression standard belonging to the MPEG-4 ISO/IEC standard (ISO/IEC 14496-2). For more information, see http://www.mpeg.org.

MPIO

Multi-purpose input output. This is a type of pin-mux pad that can be configured as GPIO or SFIO.

MSC

Mass storage device class. USB Implementers Forum computing communications protocols for the Universal Serial Bus (USB). For more information, see http://www.usb.org/developers/devclass_docs/usb_msc_overview_1.2.pdf.

MSD

Mass storage device.

MSDN

Microsoft Developer Network. For more information, see http://msdn2.microsoft.com/en-us/default.aspx.

MTD

Memory technology device, used by Linux to interact with flash memory.

MVC

Multiview Video Coding (MVC), amends H.264/MPEG-4 AVC standard to enable encoding simultaneously from multiple cameras using a single video stream.

N

NAND

Type of flash memory, typically used in USB devices and memory cards.

NB

Narrow band.

NDK

Android toolset enabling embedded components to use native code in Android applications. For more information, see http://developer.android.com/sdk/ndk/overview.html.

Netflix

Provides rental-by-mail of digital video content as well as Internet streaming on demand. For more information, see https://www.netflix.com.

NFS

Network File System, an open standard protocol.

Nv3P

NVIDIA platform programming protocol (includes 3P server and 3P client).

NvBL

NVIDIA Boot Library.

NvBlob

A Python script for producing blob files for updating hidden partitions, like for boot loader or microboot. OTA or Fastboot uses these blobs to perform the updates.

NvDDK

NVIDIA Driver Development Kit.

NVIDIA production mode

This is the mode in which Tegra chips are provided from NVIDIA. In this mode, fuses can still be programmed via recovery mode. Boot configuration tables (BCTs) and boot loaders are signed with a key of all 0's, but are not encrypted.

NvFlash

NVIDIA application for flashing devices over USB connections.

NvRM

NVIDIA Resource Manager.

NvSBK tool

NVIDIA application for producing blob objects for flashing ODM secure mode devices. The NvFlash tool uses these blobs to flash devices.

O

OAL

OEM adaptation layer for Windows CE.

ODM

Original design or device manufacturer.

ODM non-secure mode

This is the mode in which ODMs ship products without stringent security mechanisms; however, in this mode, fuses can no longer be programmed. As in NVIDIA production mode, boot configuration tables (BCTs) and boot loaders are signed with a key of all 0's and not encrypted. This mode is sometimes called ODM production mode.

ODM secure mode

This is the mode in which ODMs ship products with strict security measures in force. Fuses cannot be programmed, and all boot configuration tables (BCTs), boot loaders, and microboots must be signed and encrypted with the secure boot key (SBK).

OEM

Original equipment manufacturer.

OGA

Container for Vorbis audio-only files. For more information, see http://xiph.org.

Ogg

Container for Vorbis codec. For more information, see http://xiph.org.

Ogg Vorbis

A free/open source, lossy audio codec (Vorbis) and its container (Ogg). For more information, see http://xiph.org.

OGM

Early file format for embedding video into Ogg. Use of this format is currently discouraged by Xiph. For more information, see http://xiph.org.

ONFI

Open NAND Flash Interface, an industry workgroup that build, design-in, or enable NAND Flash memory.

OpenAL

Free cross-platform audio API (resembling OpenGL API style) for efficient rendering of multichannel three dimensional positional audio.

OpenGL ES

A subset of OpenGL 3D graphics API designed for embedded systems, defined by the Khronos Group. For more information, see http://www.khronos.org.

OpenKODE

A set of APIs for handheld games and media applications providing a cross-platform abstraction layer for other "open" media technologies. For more information, see http://www.khronos.org.

OpenSL ES

Open Sound Library for Embedded Systems. A royalty-free, cross-platform, hardware-accelerated audio API for 2D and 3D audio. For more information, see http://www.khronos.org.

OpenMAX

An application programming interface that provides abstractions for routines especially useful for computer graphics, video, and sound, defined by the Khronos Group. For more information, see http://www.khronos.org.

OpenMAX IL

OpenMAX Integration Layer. Provides an abstraction layer API between a media framework, such as DirectShow, and a set of multimedia components, such as audio and video codecs. For more information, see http://www.khronos.org.

OpenVG

A standard API for hardware-accelerated 2D vector graphics, defined by the Khronos Group. For more information, see http://www.khronos.org.

OTA

Over-the-air or wireless.

OTG

USB On-The-Go.

P

platform

The code-name of an NVIDIA® Tegra® development system, such as `kai`, `cardhu` and, `enterprise` for a Tegra 3 family device, or `ventana` or `whistler` for Tegra 2 family.

PAN

Personal area networking. A Bluetooth profile. For more information, see http://www.atheros.com/.

PCM

Pulse-code modulation.

PIP

Picture-in-picture.

PMU

Power Management Unit.

pixel depth

Number of bits per pixel (bpp).

PRISM

NVIDIA Pixel Rendering Intensity and Saturation Management (PRISM) display technology (formerly known as DIDIM). To save battery life, PRISM separates color and backlight intensity while preserving fidelity, so the amount of backlighting needed is reduced without making images appear dim.

PS

Parametric stereo.

Q

QCELP

Qualcomm Code Excited Linear Prediction, also known as Qualcomm PureVoice. Speech codec that increases the speech quality of the IS-96A codec used in CDMA. For more information, see http://www.qualcomm.com/qct.

QP

Quantization Parameter.

QuickTime

Apple multimedia framework for digital multimedia, text, animation, etc., playback/streaming. For more information, see http://www.apple.com/quicktime/download.

R

RCK

Recovery kernel.

RCM

Recovery mode messages.

RFC

Request for Comments.

RIL

Radio Interface Layer.

RNDIS

Remote NDIS. A specification for network devices on buses such as USB. For more information, see http://www.microsoft.com/whdc/device/network/NDIS/rmNDIS.mspx.

ROP

Raster operator.

RTC

Real-time clock.

RTP

Real-time transport protocol for delivering A/V content over the Internet.

RTSP

Real time streaming protocol allowing clients to issue transport commands and control a streaming media server remotely.

S

SBC

Sub-band codec. For breaking signals into different frequency bands to encode them independently.

SBK

Secure boot key.

SBR

Spectral band replication.

scan code

The physical key on the keypad.

SCO

Synchronous Connection Oriented link. For a mono, PCM audio channel.

SD

Secure Digital card. Non-volatile memory card. For more information, see http://www.sdcard.org/home.

SDHC

Secure Digital High Capacity. For more information, see http://www.sdcard.org/home.

SDHCI

Secure Digital Host Controller Interface.

SDIO

Secure Digital Input Output. SD card combined with an I/O device. For more information, see http://www.sdcard.org/home.

SDRAM

Synchronous dynamic random access memory.

SDP

Session Description Protocol, an IETF Proposed Standard that describes streaming communication sessions to announce and invite the session and to negotiate parameters.

secure boot

A common term used to refer to a boot loader that uses enhanced security, such as asymmetric encryption (public key encryption). For more information, see the Windows CE 6.0 Technical Article "Secure Download Boot Loader in Windows Embedded CE" at http://msdn2.microsoft.com/en-us/library/bb643805.aspx.

SFIO

Special function input output. This term is a category of roles that MPIO pads can be configured with.

SHOUTcast

Cross-platform media-streaming server (freeware), developed by Nullsoft, which enables Internet radio network creation. For more information, see http://www.shoutcast.com.

SIP

Session Initiation Protocol. Signaling protocol from the Internet Engineering Task Force (IETF) used to control multimedia communication sessions for voice and video over Internet protocol (VoIP).

S-LINK

Simple link interface. A high-performance data acquisition standard where data will be collected and stored by computers at both ends of the link. For more information, see http://hsi.web.cern.ch/HSI/s-link.

SLC

Single-level cell. Flash memory that stores one bit per cell.

SMP

Symmetric multiprocessing.

SMS

Short Message Service. Allows sending short text messages between mobile telephone devices.

SNR

Signal-to-noise ratio.

Sorenson

Sorenson codec used in Apple's QuickTime and in Adobe Flash. For more information, see http://www.sorensonmedia.com.

SOC

System-on-chip, which integrates computer components and other electronics into a single integrated circuit or chip. Also SoC.

S/PDIF

Sony/Philips Digital Interface.

SPI

Serial Peripheral Interface bus. A full-duplex mode, synchronous serial data link.

SPI flash

Small, low-power flash memory that uses a serial interface (usually SPI) for sequential data access.

SRC

Sample rate conversion.

SSK

Unique, per-chip Secure Storage Key used to protect customer-defined data. Typically a 128-bit key computed from the following fuse settings:

- 128-bit customer-programmed SBK.

- 32-bit customer-programmed Device Key (DK).

- 64-bit NVIDIA-programmed Unique ID (UID), which is different for every chip.

Stagefright

Media framework new in Android 2.2. For more information see http://developer.android.com/sdk/android-2.2-highlights.html#PlatformTechnologies.

T

Tegra

The world's first mobile super chip. The families of Tegra chipsets for mobile devices include:

- Tegra 3

- Tegra 2

- Tegra APX

THD

Total harmonic distortion.

TVO

Television output.

U

UART

Universal asynchronous receiver/transmitter. Computer hardware that translates data between parallel and serial forms, usually used for computer or peripheral device serial communications over a serial port.

U-Boot

Das U-Boot, a free (GNU GPL software) bootstrap loader for embedded systems. For more information, see http://www.denx.de/wiki/U-Boot.

Ubuntu

Supported Linux operating system by certain Tegra-based development products. For the specific Ubuntu version supported, see your *Release Notes*. For more information about Ubuntu, see http://www.ubuntu.com.

UIP

Update Image Partition.

ULP

Ultra low power.

USB

Universal serial bus. A standard that allows connections of many peripherals via a standardized interface socket. For more information, see http://www.usb.org.

USBNET

Linux usbnet driver. For more information, see http://www.linux-usb.org/usbnet/.

USP

Update Staging Partition.

V

VAD

Voice activation detection.

VBO

An OpenGL extension for faster rendering of triangles.

VBR

Variable bit rate.

VC-1

Common name of the SMPTE 421M video codec standard from Microsoft. For more information, see http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx.

VoIP

Voice-over-Internet protocol. Transmits voice through the Internet or other packet-switched networks.

Vorbis

A free/open source, lossy audio codec (Vorbis). For more information, see http://xiph.org.

VP6

TrueMotion VP6 video codec developed by On2 Technologies used in broadcasting, as well as by Adobe Flash and Flash Video files. For more information, see http://en.wikipedia.org/wiki/VP6.

W

WAV

Microsoft and IBM waveform audio format for storing audio bitstreams.

WEP

Wired Equivalent Privacy. Secures IEEE 802.11 wireless networks.

WMA

Microsoft Windows Media Audio technologies. Also the compressed audio file format (.wma).

WMA Lossless

Microsoft Window Media Audio lossless audio codec, provides duplication of original audio so that no data are lost.

WMA Pro

Microsoft Windows Media Audio Professional technologies.

WMA Pro LBR

Low bit rate mode of Microsoft Windows Media Audio Professional technologies.

WMV

Microsoft Windows Media Video technologies. Also the compressed video file format (.wmv).

WPA

Wi-Fi Protected Access. Certified security for wireless computer networks.

X

Xvid

Free video codec library based on the MPEG-4 standard. Xvid uses MPEG-4 Advanced Simple Profile (ASP) compression with video encoded with MPEG-4 ASP video, and so can be decoded by all MPEG-4 ASP-based decoders. For more information, see http://www.xvid.org.

Y

YAFFS

Yet Another Flash File System. The first file system designed for NAND flash.

Z

zImage

Conventional (but not required) name for the uncompressed kernel boot image file in Linux. **bzImage** is the compressed or “big” zImage file for systems requiring the kernel image to be under a certain size.

Legal Information

Notice

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

Copyright