

TI-Android-GingerBread-2.3-DevKit-2.0 DeveloperGuide



TI Android GingerBread 2.3 DevKit 2.0 Developers Guide

The objective of this document is to guide Android developers to get access to Android sources for TI devices, setting up host environment for compilation and enabling debug environment to ease the app development, debugging and deployment.

NOTE: This document is applicable for Android on AM38xx/TI81xx devices.

The TI Android Development Kit for DM3730, AM37x and AM35x devices can be downloaded from <http://focus.ti.com/docs/toolsw/folders/print/androidsdk-sitara.html>

Host (PC) setup requirements

The host development environment for Android is based on ubuntu, please install ubuntu version 10.04 or later <http://www.ubuntu.com/desktop/get-ubuntu/download>. The host installation would need few more Android specific dependencies, these can be installed dynamically over network using below commands.

For Ubuntu on 32-bit machines

```
$ sudo add-apt-repository "deb http://archive.canonical.com/  
lucid partner"  
$ sudo add-apt-repository "deb-src http://archive.canonical.com/  
ubuntu lucid partner"  
$ sudo apt-get update  
$ sudo apt-get install git-core gnupg sun-java6-jdk flex bison gperf  
libstdc++-dev libstdc++0-dev libwxgtk2.6-dev build-essential zip curl  
libncurses5-dev zlib1g-dev minicom tftpd u-boot-mkimage expect  
$ sudo update-java-alternatives -s java-6-sun
```

NOTE: Android Gingerbread (2.3) needs Java 6 on ubuntu, whereas the previous version FroYo (2.2) was using Java 5.

Source locations

TI provides Android sources for all the supported devices in multiple locations, developers can download the sources from the gitorious.org/rowboat repository or use the pre-packaged repo in the DevKit.

Using gitorious.org/rowboat

The following commands help developers to clone sources from [Gitorious.org/rowboat](https://gitorious.org/rowboat) repository

```
$ mkdir rowboat-android
$ cd rowboat-android
$ repo init -u git://gitorious.org/rowboat/manifest.git -m
TI-Android-GingerBread-2.3-DevKit-2.0.xml
$ repo sync
```

Using pre-packaged repo

The above method is ideal but is time consuming, so developers can use the pre-packaged Android sources in DevKit package.

NOTE: Once the sources are generated developers can do a “repo sync” to pull latest updates (if any) from gitorious.org/rowboat

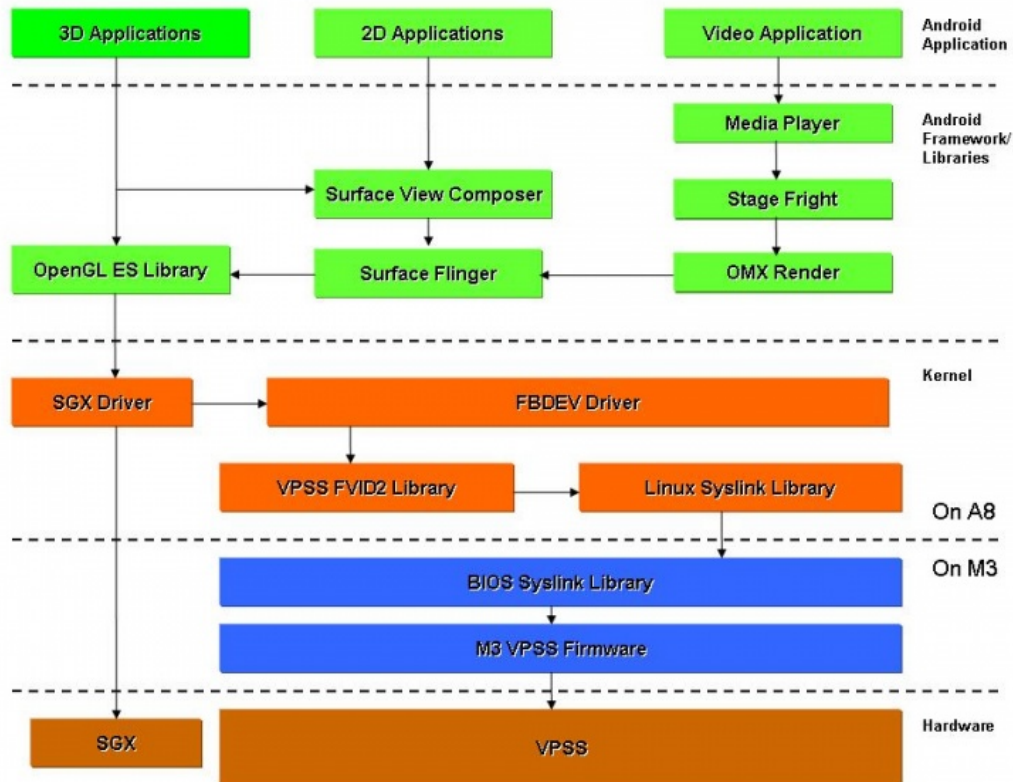
```
$ mkdir rowboat-android
$ cd rowboat-android
$ Download the pre-packaged repo from http://software-dl.ti.com/
dsps/dsps_public_sw/sdo_tii/TI_Android_DevKit/
TI_Android_GingerBread_2_3_DevKit_2_0/index_FDS.html
$ tar -xvzf TI_Android_GingerBread_2.3_DevKit_2.0_Sources.tar.bz
$ cd TI_Android_GingerBread_2.3_DevKit_2.0_Sources
$ ./repo/repo/repo sync --local-only
```

This will generate the sources for

- Android File system
- Android Linux Kernel (in kernel directory)
- u-boot boot loader (in u-boot-omap3 directory)
- Toolchain (prebuilt/linux-x86/toolchain/arm-eabi-4.4.0/bin)

Software Design Interfaces

Following diagram shows the major components that makes up the android display sub-system for AM38xx/TI81xx.



- M3 VPSS firmware

This is a firmware running over processor Cotex M3 controlling the Video Processing subsystem hardware.

M3 VPSS firmware location:
 rowboat-android/hardware/ti/ti81xx/syslink_vpss/hdvpss

- Syslink library

This is a functional layer controlling the inter-processor communication between Cotex A8 and Cotex M3.

Syslink source location:
 rowboat-android/hardware/ti/ti81xx/syslink_vpss/syslink_02_00_00_67_alpha2/

- VPSS FVID2 library

This is a HAL/functional layer controlling the Firmware running over the M3. It exposes the number of APIs controlling the video compositors, VENCs, graphics/video pipelines to the user interface drivers like V4L2 and FBDEV. It translates the Fbdev data structures and ioctl to FVID2 data structures and command and passes these to firmware running over the M3 processor.

- Frame buffer driver

This driver is registered with the FBDEV subsystem, and is responsible for managing the graphics layer frame buffer. Driver creates /dev/fb0, /dev/fb1 and /dev/fb2 as the device nodes. Application can open these device nodes to open the driver and negotiate parameters with the driver through frame buffer ioctls. Application maps driver allocated buffers in the application memory space and fills them for the driver to display.

- SGX driver

This Driver enables the SGX core which is capable of accelerating 2D/3D graphics oprations. This driver accelerates OpenGL APIs.

```
SGX source Location: rowboat-android/hardware/ti/sgx.
```

Tool chain setup

Setup the tool-chain path to point to arm-eabi- tools in prebuilt/linux-x86/toolchain/arm-eabi-4.0.0/bin

```
$ export
PATH=rowboat-android/prebuilt/linux-x86/toolchain/arm-eabi-4.4.0/bin:$PATH
```

Compilation procedure

x-loader and u-boot

Change directory to u-boot-omap3

```
$ cd rowboat-android/u-boot-omap3
```

Execute the following commands to the u-boot sources

```
$ make CROSS_COMPILE=arm-eabi- ARCH=arm distclean
$ make CROSS_COMPILE=arm-eabi- ARCH=arm <default_config_sd>
$ make CROSS_COMPILE=arm-eabi- ARCH=arm u-boot.ti
Where default_config_sd is
    ti8148_evm_min_sd      : For TI814X EVM
    ti8168_evm_min_sd      : For TI816X EVM
```

This will generate a file u-boot.min.sd which needs to be renamed to MLO

```
$ make ARCH=arm CROSS_COMPILE=arm-eabi- distclean
$ make ARCH=arm CROSS_COMPILE=arm-eabi- <default_config>
$ make ARCH=arm CROSS_COMPILE=arm-eabi-
Where default_config_sd is
    ti8148_evm_config      : For TI814X EVM
    ti8168_evm_config      : For TI816X EVM
```

This command will build the u-boot Image "u-boot.bin"

To build Linux kernel

Change directory to kernel

```
$ cd rowboat-android/kernel
```

Execute the following commands to the kernel sources

```
#> make ARCH=arm CROSS_COMPILE=arm-eabi- distclean
#> make ARCH=arm CROSS_COMPILE=arm-eabi- <default config>

Where default config is

    ti8148_evm_android_defconfig      : For TI814X EVM
    ti8168_evm_android_defconfig      : For TI816X EVM
```

```
#> make ARCH=arm CROSS_COMPILE=arm-eabi- uImage
```

This command will build the Linux Kernel Image in arch/arm/boot
"uImage"

To build Android filesystem

Building Android

To build android root file system for TI814X and TI816X

```
$ cd rowboat-android
$ make TARGET_PRODUCT= <product_name> OMAPES=6.x
```

The above command will build android along with kernel, syslink, sgx and kernel modules. To build syslink, sgx and kernel modules separately, please follow To build syslink, kernel modules and sgx section

Prepare the root filesystem

```
$ cd rowboat-android/out/target/product/<product_name>
$ mkdir android_rootfs
$ cp -r root/* android_rootfs
$ cp -r system android_rootfs
```

```
$ sudo ../../../../../../build/tools/mktarball.sh
../../../../host/linux-x86/bin/fs_get_stats android_rootfs . rootfs
rootfs.tar.bz2
```

NOTE: product_name can be ti814xevm or ti816xevm

The rootfs.tar.bz2 is the android filesystem, it can be put on a SD/MMC Card or used our NFS.

To build syslink, kernel modules and sgx (SEPARATELY)

NOTE: The above command builds all the components together, but for some reason if each component had to be build separately then please refer to the steps below.

Build setup

Export the following environment variables.

```
$export ANDROID_ROOT=<android_root>
$export
SYSLINK_ROOT=$ANDROID_ROOT/hardware/ti/ti81xx/syslink_vpss/syslink_02_00_00_67_alpha2
$export FS_ROOT=$ANDROID_ROOT/out/target/product/<product_name>
$export
IPCDIR=$ANDROID_ROOT/hardware/ti/ti81xx/syslink_vpss/ipc_1_22_03_23/packages
$export TARGET_PRODUCT=<product_name>
$export SYSLINK_VARIANT=<variant_name>
```

Note: product_name is ti814xevm for TI814X and ti816xevm for TI816X
variant_name is TI814X or TI816X

Building and installing SysLink

- Building the SysLink module

```
cd $SYSLINK_ROOT/ti/syslink/utils/hlos/knl/Linux
make ARCH=arm CROSS_COMPILE=arm-eabi- SYSLINK_PLATFORM=TI81XX clean
make ARCH=arm CROSS_COMPILE=arm-eabi- SYSLINK_PLATFORM=TI81XX
```

- Building the SysLink user library

```
$cd $SYSLINK_ROOT/ti/syslink/utils/hlos/usr/Linux
$make ARCH=arm CROSS_COMPILE=arm-eabi- SYSLINK_PLATFORM=TI81XX clean
$make ARCH=arm CROSS_COMPILE=arm-eabi- SYSLINK_PLATFORM=TI81XX
```

- Building the ProcMgr binary

```
$cd $SYSLINK_ROOT/ti/syslink/samples/hlos/procMgr/usr/Linux
$make ARCH=arm CROSS_COMPILE=arm-eabi- SYSLINK_PLATFORM=TI81XX
SYSLINK_VARIANT=TI816X clean
$make ARCH=arm CROSS_COMPILE=arm-eabi- SYSLINK_PLATFORM=TI81XX
SYSLINK_VARIANT=TI816X
```

- Installing the SysLink module, ProcMgr and M3 firmware to the filesystem

```
$cp -r $ANDROID_ROOT/device/ti/$TARGET_PRODUCT/syslink
$ANDROID_ROOT/out/target/product/$TARGET_PRODUCT/system/bin
$cp -r $SYSLINK_ROOT/ti/syslink/bin/$SYSLINK_VARIANT/syslink.ko
$SYSLINK_ROOT/ti/syslink/bin/$SYSLINK_VARIANT/samples/procmgrapp_release
\

$ANDROID_ROOT/hardware/ti/ti81xx/syslink_vpss/hdvpss/$SYSLINK_VARIANT/*
$ANDROID_ROOT/out/target/product/$TARGET_PRODUCT/system/bin/syslink/
```

Building and installing HDVPSS kernel modules

- To build VPSS/TI81xxfb and other kernel modules

```
$cd $ANDROID_ROOT/kernel
$make ARCH=arm CROSS_COMPILE=arm-eabi-
KBUILD_EXTRA_SYMBOLS=$SYSLINK_ROOT/ti/syslink/utils/hlos/knl/Linux/Module.symvers
modules
```

- To install modules

```
$make ARCH=arm CROSS_COMPILE=arm-eabi-
INSTALL_MOD_PATH=$FS_ROOT/system/ \

KBUILD_EXTRA_SYMBOLS=$SYSLINK_ROOT/ti/syslink/utils/hlos/knl/Linux/Module.symvers
modules_install
```

Building and installing 3D graphics acceleration (SGX) drivers

To build and install SGX

```
$cd $ANDROID_ROOT/hardware/ti/sgx
$make ANDROID_ROOT_DIR=$ANDROID_ROOT OMAPES=6.x
$make ANDROID_ROOT_DIR=$ANDROID_ROOT OMAPES=6.x install
```

To build rootfs, please follow Prepare the root filesystem section.

To generate SD/MMC card to boot Android

These compiled Images can be copied to a SD / MMC card to boot Android on TI814X/TI816EVM EVM

- Generate boot.scr using mk-bootscr from tools folder

```
$ ./mkbootscr
```

- Copy all the images to one folder

```
$ mkdir image_folder
$ cp rowboat-android/kernel/arch/arm/boot/uImage image_folder
$ cp rowboat-android/u-boot-omap3/u-boot.bin image_folder
$ cp rowboat-android/u-boot-omap3/MLO image_folder
$ cp <devkit>/Tools/mk-bootscr/boot.scr image_folder
$ cp rowboat-android/out/target/product/ti816xevm/rootfs.tar.bz2
image_folder
$ cp -rf media_clips image_folder
```

```
$ cp Tools/mk-mmc/mkmmc-android.sh image_folder
$ ./mkmmc-android <sd card mounted dev folder example:/dev/sdc>
MLO u-boot.bin uImage boot.scr rootfs.tar.bz2 Media_Clips
```

NOTE: mkmmc-android.sh and mk-bootscr are provided in tools folder in DevKit

- Configure and boot

TI814X:

```
u-boot#> setenv bootargs 'mem=166M@0x80000000 mem=256M@0xC0000000
console=ttyO0,115200n8 androidboot.console=ttyO0 root=/dev/mmcblk0p2 \
rw rootfstype=ext3 rootdelay=1 init=/init ip=off vram=50M'
u-boot#> mmc init
u-boot#> fatload mmc 0 81000000 uImage
u-boot#> bootm 0x81000000
```

TI816X:

```
u-boot#> setenv bootargs 'mem=166M@0x80000000 mem=768M@0x90000000
console=ttyO2,115200n8 androidboot.console=ttyO2 root=/dev/mmcblk0p2 \
rw rootfstype=ext3 rootdelay=1 init=/init ip=off'
u-boot#> mmc init
u-boot#> fatload mmc 0 81000000 uImage
u-boot#> bootm 0x81000000
```

NOTE:

- If the board has bootargs configured already, then the board will not boot for Android automatically,
- It is suggested to either delete the bootargs or use the following commands on u-boot prompt through UART console.

```
u-boot#>; mmc init
u-boot#>; fatload mmc 0 0x81000000 boot.scr
u-boot#>; source 0x81000000
```

If the board is not configured for bootargs, then it automatically boots.

Hardware Setup and Requirements

This release of TI Android GingerBread 2.3 DevKit 2.0 is evaluated on the below given list of platforms. This package should be easily portable on other platforms on similar TI devices.

Requirements

TI Device	Platform Supported	Version	Other Accessories
TI814X			
	TI814X EVM	Rev B	HDMI Monitor, USB Keyboard, USB Mouse, Ethernet, UART Cable, Audio Speakers, MMC/SD Card (2GB min)
TI816X			
	TI816X EVM ^[1]	Rev X	HDMI Monitor, USB Keyboard, USB Mouse, Ethernet, UART Cable, Audio Speakers, MMC/SD Card (2GB min)

Setup

This section describes the setup and instructions to run TI's Android DevKit OOB demo on TI814X and TI816X EVM.

Step 1: Insert SD card into MMC/SD slot on the EVM

Step 2: Set the DIP switch settings to boot from MMC/SD card For MMC/SD boot - On TI814X EVM the DIP switch S1 and on TI816X EVM the DIP switch SW3 should be set as shown below

Switch	1	2	3	4	5	6	7	8	9	10
State	ON	ON	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF

Step 3: Turn ON the EVM

Refer to User guide ^[2] and RowboPERF user guide ^[3] from document folder in DevKit release

Application Development and Debugging

Using ADB Android Debugger, Downloader

Android Debug Bridge (adb) is a versatile tool lets you manage the state of the Android-powered device. For more information about what is possible with adb, see Android Debug Bridge page at <http://developer.android.com/guide/developing/tools/adb.html>. The ADB tool can be used to

- Download an application from a host machine, install & run it on the target board.
- Start a remote shell in the target instance.
- Debug applications running on the device using the debugging tool DDMS (Dalvik Debug Monitor Server) which runs on top of adb connection.
- Copy files to and from the board to host machine

Downloading "ADB" & Host setup

The adb tool is a part of Android SDK package located at <http://developer.android.com/sdk/index.html>. For an overview of how to install and set up the Android SDK, follow download & setup instructions from <http://developer.android.com/sdk/index.html>. Once you install Android SDK, the directory contents look like this.

```
|-- add-ons
  |-- platforms
  |-- platform-tools
  |   |-- aapt
  |   |-- adb
  |   |-- aidl
  |   |-- dexdump
  |   |-- dx
  |   |-- lib
  |   |-- llvm-rs-cc
  |   |-- NOTICE.txt
  |   `-- source.properties
|-- SDK Readme.txt
|-- temp
`-- tools
    |-- adb_has_moved.txt
    |-- android
    |-- ant
    |-- apkbuilder
    |-- ddms
    |-- dmtracedump
    |-- draw9patch
    |-- emulator
    |-- etcltool
    |-- hierarchyviewer
    |-- hprof-conv
    |-- layoutopt
    |-- lib
    |-- mksdcard
    |-- monkeyrunner
```

```
|-- NOTICE.txt
|-- proguard
|-- source.properties
|-- sqlite3
|-- traceview
`-- zipalign
```

The adb tool is located in platform-tools/ directory under the Android SDK installation. Export the tools and platform-tools directory path as shown below.

```
Host$: export
PATH=${PATH} :<your_sdk_dir>/tools:<your_sdk_dir>/platform-tools
```

Connecting Host machine to board through adb

This release of DevKit has been tested for adb over Ethernet method of connecting a given board with host machine. The below section describes this method and provides necessary instructions for the same.

adb over Ethernet

- Make sure Ethernet port on board and host machine are connected to the network
- Check Ethernet configuration for the board

```
target #> netcfg

lo          UP      127.0.0.1      255.0.0.0      0x00000049

eth0        UP      172.24.190.59  255.255.252.0  0x00001043
```

- If Ethernet was not configured, configure Ethernet of the board using ifconfig/netcfg as shown below.

```
target #> netcfg eth0 dhcp
```

- Configure the ADB Daemon to use an ethernet connection using setprop as shown below.

```
target #> setprop service.adb.tcp.port 5555
```

- If network is configured successfully (above steps) then Restart service adbd on the target,

```
target #> stop adbd
target #> start adbd
```

- On the host machine use following commands to establish adb connection

```
$> export ADBHOST=<target's ip address>
$> adb kill-server
$> adb start-server
```

- Verify for device connectivity, by executing the following commands

```
$> adb devices
```

If connected, find the device name listed as a "emulator"

```
List of devices attached
emulator-5554    device
$ adb shell
```

For more information about adb commands, see Android Debug Bridge page at <http://developer.android.com/guide/developing/tools/adb.html>

Copy any files to and from the board over ADB

- Using the adb commands "pull" and "push" user can copy files to and from the board.
- Unlike the install command, which only copies an .apk file to a specific location, the pull and push commands let you copy arbitrary directories and files to any location on the board.
- To copy a file or directory (recursively) from the board, use

```
adb pull <remote> <local>
```

- To copy a file or directory (recursively) to the board, use

```
adb push <local> <remote>
```

In the commands, <local> and <remote> refer to the paths to the file or directory on your development host (local) and on the target instance (remote).

```
Here's an example:
adb push foo.txt /sdcard/foo.txt
```

Running Applications

The root File System provided in this DevKit releases contains only standard Android components and applications. User might be interested to download & run android applications (.apk) available in the market. The below procedure gives the steps to be followed to download any .apk file to the board and run it on the platform.

Installing (.apk files) application on Target Platform

- From the host: You can use adb tool for package installation.

```
$> adb install <package>.apk.
```

NOTE: Use -s option with the adb tool, to install the package on external storage.

On successful installation adb tool will report SUCCESS on host terminal, and the application would be listed on the android main menu.

Un-installing applications (.apk) using adb

- To un-install non-default components (that were installed later)
 - Method 1: On the host machine execute the following

```
$> adb uninstall <package>.apk
```

- Method 2: On target:

```
Main menu -> Menu -> Settings -> Applications -> Manage
applications -> Find the package
Tap on it -> Uninstall -> OK -> OK
```

- On successful removal, the application would have been removed from the android main menu. All the short-cuts to the application also removed.
- To un-install default components, use the following commands from abd on host machine

```
$ adb shell
#rm /system/app/app.apk
```

On successful removal, the application would have been removed from the android main menu.

Setup ADB for application Debugging

ADB and Eclipse, with ADT(Android Development Tools plug-in) allow users to create and debug Android applications. Follow Developing In Eclipse, with ADT at [http:// developer. android. com/ guide/ developing/ eclipse-adt.html](http://developer.android.com/guide/developing/eclipse-adt.html)

Steps to connect Eclipse to the board.

- Setup the adb connection with the board by following the instructions given in adb over Ethernet section

```
Verify the connectivity by executing
$ adb devices
```

- Open Eclipse IDE. Eclipse, with ADT plugin enable users to
 - Create an android project.
 - Build and Run the project on a connected board.
 - Debug the project using the Debug perspective.
 - Use DDMS (Dalvik Debug Monitor Server) to monitor the connected board.

For more detailed and complete information on the above follow Developing In Eclipse, with ADT at [http:// developer. android. com/ guide/ developing/ eclipse-adt.html](http://developer.android.com/guide/developing/eclipse-adt.html)

- Open DDMS(Dalvik Debug Monitor Server) perspective. This DDMS perspective can be opened from the eclipse menu via:

```
Window -> Open Perspective -> Other -> DDMS;
Click on OK
```

- DDMS provides port-forwarding services, screen capture on the device, thread and heap information on the device, logcat, process, and radio state information, incoming call and SMS spoofing, location data spoofing, and more.
- For more information on DDMS and to use it, follow Using the Dalvik Debug Monitor page at [http://developer. android. com/ guide/ developing/ tools/ ddms. html](http://developer.android.com/guide/developing/tools/ddms.html)

Using TI's Code Composer Studio V5 (Eclipse based)

Code Composer Studio (CCS) is the integrated development environment for TI's DSPs, microcontrollers and application processors based on the Eclipse open source software framework which includes a suite of tools used to develop and debug embedded applications. Since CCS is based on Eclipse, it is possible to integrate the Android Development Tools (ADT) like Android Debug Bridge (ADB), Dalvik Debug Monitoring System (DDMS) and ndk-gdb in CCS to enable the debugging of Android Applications directly on Android Device (i.e TI EVM) along with the inherent CCS capability of Linux Aware Debug and DSP Debugging.

This wiki walks you through installation of Android Debugging Tools in CCS, connecting CCS with Android Device and illustrates a debugging session of an HelloWorld Application based on android NDK having a mix of Java and Native C Code.

- Pre-requisite Installations :
 - CCSv5.0.1 (or Eclipse Helios) - http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v5

- Android SDK - <http://developer.android.com/sdk/index.html>
- Android NDK - <http://developer.android.com/sdk/ndk/index.html>
- Cygwin - <http://cygwin.com/install.html> (For Windows Host)
- Android Device running Android 2.3 Gingerbread – www.arowboat.com

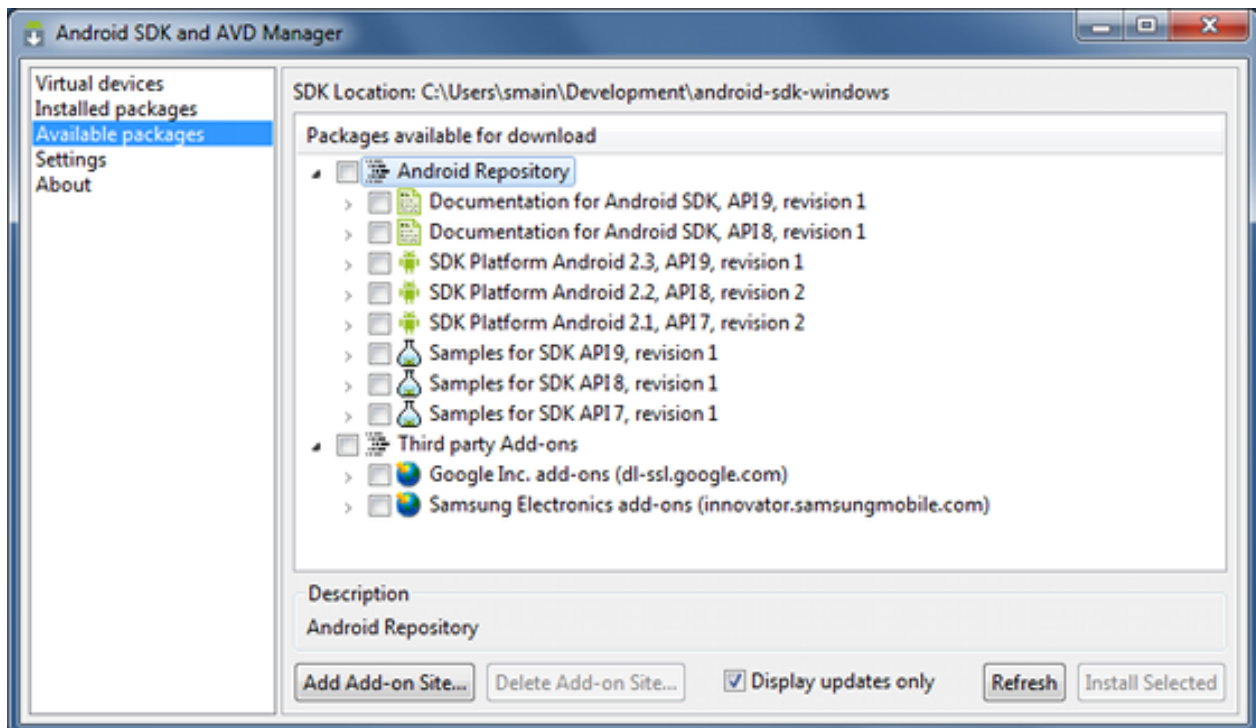
Preparing CCS for Android Development

Note: The steps below are condensed version of steps mentioned on <http://developer.android.com/sdk/installing.html>

1. Installing the ADT Plugin in CCS Refer <http://developer.android.com/sdk/eclipse-adt.html#installing>

2. Adding Platforms and Other Components

- On Windows, double-click the SDK Manager.exe file at the root of the Android SDK directory
- On Linux, run '<SDK>tools/android'
- Proxy Settings (If needed): Go to Settings and set the Proxy to work with your network
- To download components, use the graphical UI of the Android SDK and AVD Manager, shown in Figure below, to browse the SDK repository and select new or updated components. The Android SDK and AVD Manager will install the selected components in your SDK environment. Refer the list of Recommended Components: <http://developer.android.com/sdk/installing.html#which>



- To Install the Terminal (similar to Teraterm) in CCS, please follow the steps mentioned in the wiki: http://processors.wiki.ti.com/index.php/How_to_install_the_terminal_plugin_in_CCSv5

Setup debugging via ADB over network

Step 1: Upon installing the ADT plugin, DDMS should have been setup. DDMS configuration should be changed as in below:

```
Click on Window->Preferences; Select Android -> DDMS
Change - ADB debugger base port: 8700; Logging Level: Verbose
Click on Apply
```

Step 2: DDMS perspective can now be opened from the eclipse menu via:

```
Window -> Open Perspective -> Other -> DDMS;
Click on OK
```

Step 3: Attach CCS to your TI EVM:

To connect CCS to your TI EVM, please follow procedure mentioned in the section adb over Ethernet

Check if you are now connected to the TI EVM device by running the following command

```
$ adb devices
```

It should output something like:

```
emulator-5554 device
```

This confirms that EVM is connected. With this setup, you should be able to use ADB, logcat, DDMS and other tools directly from CCS ADT environment for creating your applications for Android.

Note: you can refer the following for more details: <http://developer.android.com/guide/developing/device.html>
http://www.omappedia.org/wiki/Android_Debugging#Debugging_on_Zoom2_with_Eclipse_ADT

Debugging Android with CCS

Hello World Application on Android Virtual Device (AVD)

http://processors.wiki.ti.com/index.php/TI-Android-FroYo-DevKit-V2_CCSv5SetupGuide#Hello_World_Using_CCSv5
<http://developer.android.com/resources/tutorials/hello-world.html>

Hello-jni of Android NDK on Android Device

- In CCS:
 1. Click File > New Android Project...
 2. Select the Create project from existing source radio button.
 3. Select any API level above Android 1.5.
 4. In the Location field, click Browse... and select the <ndk-root>/samples/hello-jni directory.
 5. Click Finish.
 6. Go to C/C++ Perspective. Click File->New->Convert to C/C++ Project
- Edit jni/Android.mk and add the following line before BUILD_SHARED_LIBRARY: APP_CFLAGS := -g. This will add debugging symbols in to your native C code.
- Compile the native code using the ndk-build command from cmd prompt:

```
cd <ndk-root>/samples/hello-jni
<ndk_root>/ndk-build
```

- In CCS, create a debug configuration for a C/C++ application:
 - Create a new debug configuration for a C/C++ application. Click Run->Debug Configurations.
 - Set the process launcher to "Standard Create Process Launcher"
 - On "Main" tab, set:

C/C++ Application: android-ndk-r5-windows\android-ndk-r5\samples\hello-jni\obj\local\armeabi\app_process

Select Disable auto build
 - On "Debugger" tab, set:
 - Debugger: gdbserver

- Stop on startup at: Java_com_example_hellojni_HelloJni_stringFromJNI (It is the entry function of the native C code)
- Main Tab:
 - GDB debugger:
 - android-ndk-r5-windows\android-ndk-r5\toolchains\arm-eabi-4.4.0\prebuilt\windows\bin\arm-eabi-gdb.exe
 - GDB command file: android-ndk-r5-windows\android-ndk-r5\samples\hello-jni\libs\armeabi\gdb.setup
 - GDB command set: Standard
 - Protocol: mi
 - Select only Verbose console mode
- Connection Tab:
 - Type: TCP
 - Host name or IP address: localhost
 - Port number: 5039
- Open the ndk-gdb script that came with the android NDK and comment the last line (we are not calling the usual gdb client, but we will attach an Eclipse gdb session instead)

```
# $GDBCLIENT -x $GDBSETUP -e $APP_PROCESS
```

- Only For Windows: Run dos2unix.exe on ndk-gdb from cygwin after editing to make sure that we do not have any unwanted dos symbols in the script
- Starting Java Debugger:
 - Make sure you are in Java Perspective.
 - Click Run->Debug Configurations.
 - Place a breakpoint in Java code just before the native code call
 - Select Android Debug Configuration instance "HelloJni" and click "Debug"
 - It will prompt you to switch and open to CCS Debug Perspective. Select "yes"
- Starting GDB Debugger:
 - In cygwin, make sure that the root of android-ndk is in PATH.
 - In cygwin, Go to /android-ndk-r5-windows/android-ndk-r5/samples/hello-jni and execute the following and wait until you get a prompt again:

```
ndk-gdb
--adb=<PATH_TO_android-sdk-windows/platform_tools/adb.exe>

sample command:
$ ndk-gdb
--adb=/cygdrive/c/PROGRA~1/Android/android-sdk-windows/platform-tools/adb.exe
```

- CCS, go to C/C++ Perspective and click Run->"Debug". This will launch the gdbserver in CCS Debug Perspective.
- Now, you can seamlessly debug between Java and Native C code

Adobe Flash 10 Integration

The Android version of Flash10 that runs on GingerBread is now available for customer download (by registration) at, <http://focus.ti.com/docs/toolsw/folders/print/adobeflash-a8.html>

The below steps give the procedure to download the Adobe Flash 10 library for Android Gingerbread and installing the same in File system.

- Download the flashplayer installer “Flash10.1_Android_Webkit_Plugin-0.4-Linux-x86-Install.bin” from <http://focus.ti.com/docs/toolsw/folders/print/adobeflash-a8.html>
- Execute the installer

```
$ ./ Flash10.1_Android_Webkit_Plugin-0.4-Linux-x86-Install.bin
```

```
Will result in following instruction, press "Y"
```

```
This will install Flash10.1 Android Webkit Plugin on your computer.  
Continue? [n/Y] Y
```

```
Select the source install location
```

```
Where do you want to install Flash10.1 Android Webkit Plugin?
```

```
[/home/user/flash10_androidplugin] /home/user/flash10_androidplugin
```

```
Installing Flash10.1 Android Webkit Plugin...
```

```
Installing Program
```

```
Files...
```

```
Installation complete.
```

```
After Installation the following directory structure is resulted
```

- Change to Flash installed directory on Host PC

```
$ cd flash10_androidplugin  
$ ls  
install_flash_player.apk  uninstall
```

- Install flash player plug in on target via adb

```
$ adb install install_flash_player.apk
```

- Do the browser configuration
 - Explained in the below section
- Test the Adobe Flash installation
 - Browse the link <http://www.adobe.com/software/flash/about/>
 - Should display Adobe Flash Player Successfully Installed

Acronyms

- SDK – Software Development Kit
- NDK – Native Development Kit (For more information, refer <http://developer.android.com/sdk/ndk/overview.html>)
- ADT – Android Development Tools
- ADB – Android Debug Bridge
- DDMS – Dalvik Debug Monitoring System
- AVD – Android Virtual Device

Quick References

This is Release TI-Android-GingerBread-2.3-DevKit-2.0 for AM38xx/TI81xx devices. The release is available from http://software-dl.ti.com/dsp/dsp_public_sw/sdo_tii/TI_Android_DevKit/TI_Android_GingerBread_2_3_DevKit_2_0/index_FDS.html

Content	Link Ref#
User Guide	[2]
Release Notes	[4]
Downloads	[5]

Support

For further information or to report any problems, contact <http://e2e.ti.com/android> or <http://support.ti.com>.

For community support join <http://groups.google.com/group/rowboat>

For IRC #rowboat on irc.freenode.net

References

- [1] <http://focus.ti.com/docs/toolsw/folders/print/tmdxevm8168.html>
- [2] http://processors.wiki.ti.com/index.php/TI_Android_GingerBread_2.3_DevKit_2.0_UserGuide
- [3] http://processors.wiki.ti.com/index.php/RowboPERF_User_Guide
- [4] http://processors.wiki.ti.com/index.php/TI-Android-GingerBread-2.3-DevKit-2.0_ReleaseNotes
- [5] <http://focus.ti.com/docs/toolsw/folders/print/androidsdk-sitara.html>

Article Sources and Contributors

TI-Android-GingerBread-2.3-DevKit-2.0 DeveloperGuide *Source:* <http://processors.wiki.ti.com/index.php?oldid=57492> *Contributors:* Khasim, Pankaj.bharadiya

Image Sources, Licenses and Contributors

Image:TiBanner.png *Source:* <http://processors.wiki.ti.com/index.php?title=File:TiBanner.png> *License:* unknown *Contributors:* Nsnehaprabha

File:Android AM389x stack.jpg *Source:* http://processors.wiki.ti.com/index.php?title=File:Android_AM389x_stack.jpg *License:* unknown *Contributors:* Pankaj.bharadiya

File:Android-sdk-manager.png *Source:* <http://processors.wiki.ti.com/index.php?title=File:Android-sdk-manager.png> *License:* unknown *Contributors:* Satish npatel

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED. BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

License

1. Definitions

- "**Adaptation**" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
- "**Collection**" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.
- "**Creative Commons Compatible License**" means a license that is listed at <http://creativecommons.org/compatibility/licenses> that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
- "**Distribute**" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
- "**License Elements**" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
- "**Licensor**" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
- "**Original Author**" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer (being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
- "**Work**" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
- "**You**" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
- "**Publicly Perform**" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
- "**Reproduce**" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
 - to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
 - to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
 - to Distribute and Publicly Perform Adaptations.
 - For the avoidance of doubt:
- Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
 - Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
 - Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.
- You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.
- If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
- Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

- This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

1. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
 2. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
 3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
 4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
 5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
 6. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.
-