FastCV 1.0 -- Getting Started Guide

FastCV Sample App Install Guide

80‑xxxxx‑x Rev. x

October 8, 2011

Submit technical questions at:

[https://support.cdmatech.com/](https://support.cdmatech.com)

Revision history

| Revision | Date | Description |
| --- | --- | --- |
| A | Month Year | Initial release |

Table of Contents

[1 What is FastCV 4](#_Toc305852410)

[2 Document Overview 4](#_Toc305852411)

[3 Step1: Setting up the Developer Environment 5](#_Toc305852412)

[3.1 Supported Development Platforms 5](#_Toc305852413)

[3.2 Set-up the Android Development Environment 5](#_Toc305852414)

[3.2.1 JDK 5](#_Toc305852415)

[3.2.2 Eclipse IDE 6](#_Toc305852416)

[3.2.3 Android SDK Downloader 6](#_Toc305852417)

[3.2.4 Android ADT 7](#_Toc305852418)

[3.2.5 Android SDK Platform Support 8](#_Toc305852419)

[3.2.6 Cygwin Environment 10](#_Toc305852420)

[3.2.7 Android NDK 11](#_Toc305852421)

[3.2.8 C/C++ Development Toolkit (CDT) 12](#_Toc305852422)

[4 Step 2: Installing the FastCV SDK 13](#_Toc305852423)

[4.1 Download 13](#_Toc305852424)

[4.1.1 Windows 13](#_Toc305852425)

[4.1.2 Mac OS X 13](#_Toc305852426)

[4.1.3 Linux 13](#_Toc305852427)

[4.2 Clean Installation 13](#_Toc305852428)

[4.2.1 Resulting Directory Structure 14](#_Toc305852429)

[4.3 Install the FastCV SDK 15](#_Toc305852430)

[4.3.1 Download 15](#_Toc305852431)

[4.3.2 Windows 15](#_Toc305852432)

[4.4 Prepare Test Device for Development 15](#_Toc305852433)

[4.4.1 Developer Settings on the Phone 15](#_Toc305852434)

[4.4.2 Install the USB Driver 17](#_Toc305852435)

[5 Step 3: Compiling and Running a FastCV Sample App 18](#_Toc305852436)

[5.1 Create the Android APK & Compiling Shared object 18](#_Toc305852437)

[5.1.1 Building the FastCVSample application 18](#_Toc305852438)

[5.2 Run the Sample Application 22](#_Toc305852439)

[6 Tips/Tricks: 23](#_Toc305852440)

[6.1 Tip 1: 23](#_Toc305852441)

[6.2 Tip 2: 23](#_Toc305852442)

[6.3 Tip3: 23](#_Toc305852443)

# What is FastCV

FastCV is a computer vision library being released free of charge by Qualcomm. The library is targeted toward very sophisticated Computer Vision (CV) application developers who are interesting in writing CV applications that run on mobile devices. FastCV 1.0 is specifically designed to run efficiently on Android mobile devices, but subsequent versions of FastCV will support mobile devices running iOS, and WinPhone operating systems.

The FastCV 1.0 library will be released as a unified binary. This unified binary file has a single API, but contains two implementations. The first implementation will run efficiently on any ARM processor, and is named ‘FastCV for ARM’. The second FastCV implementation runs only on Qualcomm SoC’s and is called ‘FastCV for Snapdragon’. ‘FastCV for Snapdragon’ has the same API as ‘FastCV for ARM’, but provides hardware accelerated implementations of the CV functions included in the library.

For more information on the specific functions supported by FastCV please refer to the online documentation located at: <http://fastcv.qualcomm.com> . Additionally, please feel free to download the FastCV unified binary, a sample application and accompanying documentation free of charge at our developer site: <http://fastcv.qualcomm.com/download>..

# Document Overview

The purpose of this document is to help a developer get started evaluating our FastCV 1.0 Library. A copy of the FastCV SDK can be downloaded from <http://fastcv.qualcomm.com>

Step 1: Settting up the Developer Environment:

* Explain how to configure a devlopment environment so that the FastCV Sample Application can be tried out.

Step 2: Install the FastCV SDK

* Explain how to install the FastCV SDK

Step3: Compile and running a FastCV sample application

* Compilation and installation instructions for our Android Fast Corner Dectection Sample Application that utilizes our FastCV1.0 library.

# Step1: Setting up the Developer Environment

## Supported Development Platforms

The FastCV SDK supports Android OS 2.1 and above.

The recommended development environment is Microsoft Windows 7 32/64bit or Windows XP.

The components to build the actual code (JDK, Eclipse+ADT and gcc) are available across multiple platforms. While building on Linux Ubuntu and Mac OS X environments is possible, we will be unable to support those platforms with detailed documentation. However, we have included hints and notes to platform specific issues that will help you set up your development environment on Linux Ubuntu 10.10 or Mac OS X 10.6 (Snow Leopard). This set-up guide has been written for the Win7 32/64bit platform with special notes for WinXP.

**NOTE**: If you already have both the Android SDK and the NDK setup, then please go directly to Chapter 4[: Installing the FastCV SDK](https://ar.qualcomm.at/qdevnet/sdk/android/Installing%20QCAR%20SDK).

## Set-up the Android Development Environment

The FastCV SDK requires both the Android SDK and the NDK. The Android NDK is an extension to the Android SDK that lets Android developers build performance-critical parts of their applications in native code. SDK and NDK communicate over the Java-Native-Interface (JNI).

To set-up the development environment please install the following components in the order they appear below:

1. [JDK](https://ar.qualcomm.at/qdevnet/sdk/android/JDK)
2. [Eclipse IDE](https://ar.qualcomm.at/qdevnet/sdk/android/Eclipse%20IDE)
3. [Android SDK Downloader](https://ar.qualcomm.at/qdevnet/sdk/android/Android%20SDK%20Downloader)
4. [Android ADT](https://ar.qualcomm.at/qdevnet/sdk/android/Android%20ADT)
5. [Android SDK platform support](https://ar.qualcomm.at/qdevnet/sdk/android/Android%20SDK%20Platform%20Support)
6. [Cygwin Environment](https://ar.qualcomm.at/qdevnet/sdk/android/Cygwin%20Environment)
7. [Android NDK](https://ar.qualcomm.at/qdevnet/sdk/android/Android%20NDK)

### JDK

Download the **Java SE Development Kit 7** (JDK)

<http://www.oracle.com/technetwork/java/javase/downloads/>

Click the 'Download' button from the JDK section of the 'Java Platform, Standard Edition' table, JavaSE7.

Install the JDK environment with default settings.

Detailed installation instructions and system requirements can be found at:

<http://www.oracle.com/technetwork/java/javase/index-137561.html>

**MAC**: The JDK is already integrated into the Mac OS X operating system.

### Eclipse IDE

In this step you will install the Eclipse IDE. Download the latest version of **Eclipse IDE for Java Developers** from

<http://www.eclipse.org/downloads/>

**NOTE**: You may use other versions of Eclipse too, but our setup guide assumes this package!

Unpack the downloaded ZIP package and copy the contents of the archive starting with the subdirectory 'eclipse' to your program directory path in C:\Program Files\eclipse. You may also create a shortcut to eclipse.exe on your desktop or start menu.

Start eclipse.exe

The very first time that Eclipse is started, the IDE will ask you to assign storage space for your workspace. This directory will contain ONLY IDE specific settings and information. This is not your application development workspace (despite the notice in this window). This type of information is typically stored in the user's home directory, which is the default value here:C:\Users\USERNAME\workspace. Check "Use this as the default and do not ask again".

**MAC**: For Mac OS X we recommend that you use the 32-bit version of Eclipse.

**LINUX**: Eclipse is not always able to determine the location of the JVM, despite being in the path, To fix this, insert into <path-to-eclipse-dir>/eclipse.ini the following line at the top:

-vm /usr/java/jdk1.6.0\_26/bin/java

### Android SDK Downloader

The Android SDK is distributed through an SDK starter package containing the SDK Tools. Download the starter package from:

<http://developer.android.com/sdk/index.html>

The FastCV SDK uses r10 of the Android SDK. Unzip the archive and copy the contents into a directory, Ex. C:\Development\Android\android-sdk-windows\.

Throughout the Get Started guide we will refer to the base directory of your development environment as

<DEVELOPMENT\_ROOT>= C:\Development\Android

**NOTE**: **Do not use pathnames with spaces.**

Add the tools\  directory to your Windows path. Right-click on Computer on the desktop and selectProperties. Use the Advanced system settings to open the System Properties window and select Environment Variables on the Advanced tab. Look for variable  Path  in the System variables window. After pressing Edit, scroll to the end of Variable value: and add the full path to the tools\ and the platform-tools\ directory to the end of the path, separated by a semicolon from the one before. For example:

;C:\Development\Android\android-sdk-windows\tools\;C:\Development\Android\android-sdk-windows\platform-tools\

**NOTE**: The last "\" at the end of the  Path variable has to be included.

Android ADT installation in the next step will use this path to identify the Android development environment.

#### Troubleshooting

For troubleshooting Android related issues and for more detailed instructions on the Android SDK set-up, please refer to the following:

* [Quick Start Android SDK](http://developer.android.com/sdk/index.html) set-up
* [Installing the Android SDK](http://developer.android.com/sdk/installing.html).

**MAC**: Update $path variable to point to the 'make' utility and the Android SDK tools directory. Add the following to the /Users/<login-name>/.tcshrc:

set path = ( $path /Developer/usr/bin  /Users/<login-name>/Development/Android/android-sdk-mac\_x86/tools )

**LINUX**: Update your PATH to point to the 'make' utility and the Android SDK tools directory. If you use bash shell add the following to the ~/.bashrc :

export PATH=~/bin:/opt/android-sdk-linux\_x86/tools/:$PATH

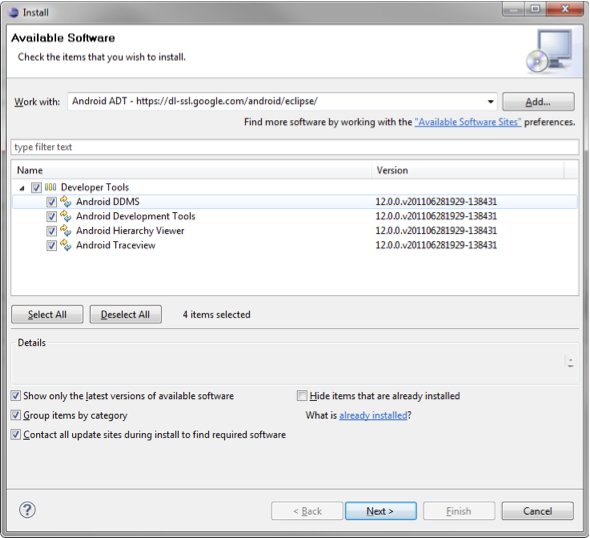
### Android ADT

Android Development Tools (ADT) is a powerful extension to Eclipse that connects it with the Android SDK and helps with app development. This package is installed from within Eclipse. Select Help->Install New Software... . Add the url

https://dl-ssl.google.com/android/eclipse/

into Work with field. Eclipse will ask you to provide an arbitrary name for the update site.

After a short while Developer Tools appears in the field. At the very minimum, please select Android Development Tools and Android DDMS - which adds debugging support - from the list, and then click on  Next-> .



Eclipse component selection

After reviewing this selection and accepting the license terms, the downloader fetches the files and puts them in your Eclipse directory. After accepting the certificate and an automatic restart of Eclipse, the installation of the Android ADT is complete.

### Android SDK Platform Support

To develop for Android, support for the appropriate Android platform must be installed. The **Android SDK and AVD Manager** is used to install additional components and support for different platforms. There are two methods to invoke the SDK and AVD Manager. Start the SDK SDK Manager.exe from the root of the SDK setup directory, or select in Eclipse the menu Window->Android SDK and AVD Manager. In case the Android SDK location was not setup correctly within Eclipse, go to Windows->Preferences->Android  and set the SDK location field to the root of your SDK install directory.

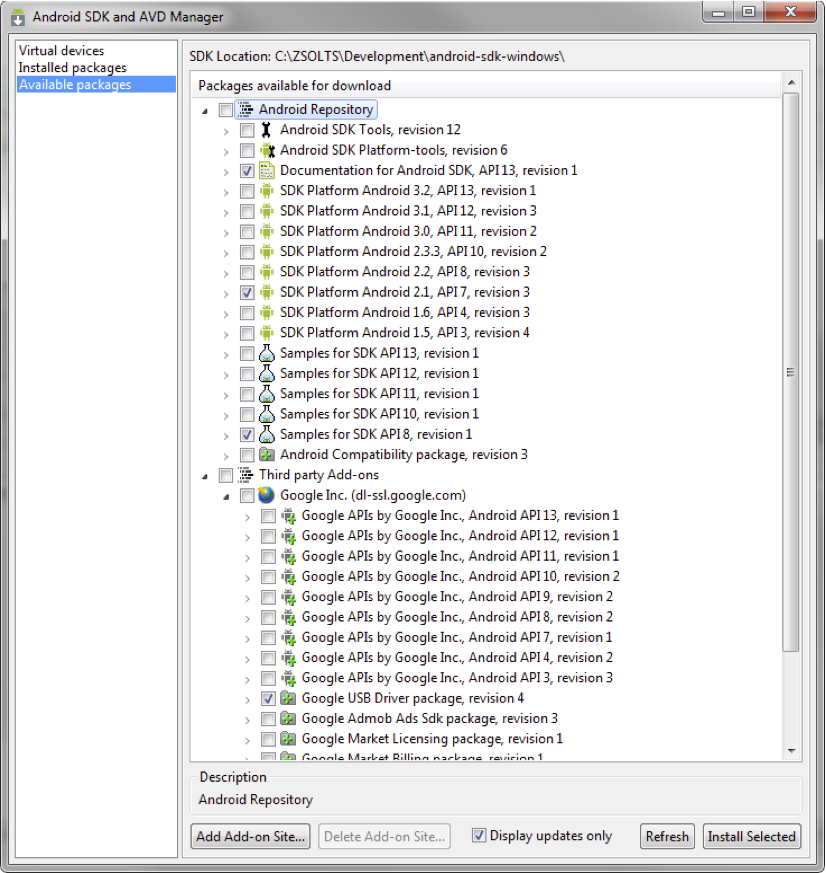
In the Manager select Available Packages and check the following boxes under the standard repository https://dl-ssl.google.com/android/repository/repository.xml :

From the Android Repository:

* Documentation for Android SDK, API 13, revision 1
* SDK Platform Android 2.1, API 7, revision 3
* Samples for SDK API 8, revision 1 (optional)

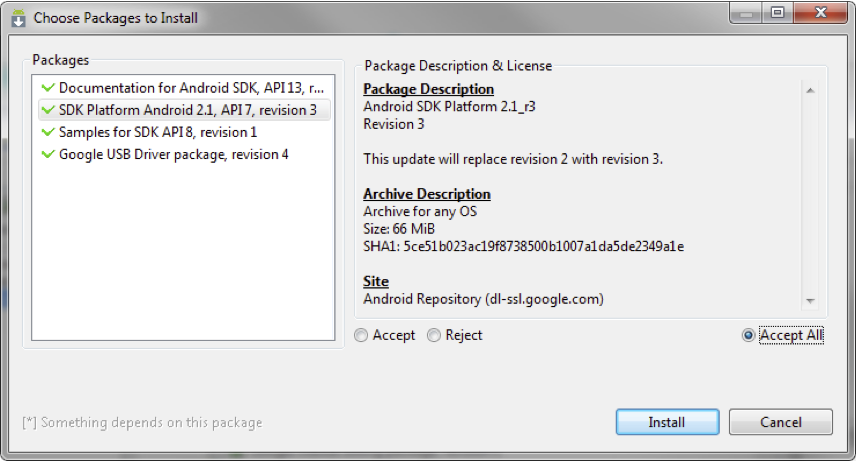
From the Third party Add-ons / Google Inc. add-ons:

* Google USB Driver package, revision 4



Android SDK component selection

To install the selected options you must hit Install Selected and accept all licenses on the next window. Use Accept All  as a shortcut and hit Install.



Android component license acceptance

The Eclipse environment is now ready for Android development.

### Cygwin Environment

**LINUX**: This section is not relevant for Linux users.

**MAC**: This section is not relevant for Mac users who have Apple Developer Tools (XCode) installed. Install XCode if necessary from <http://developer.apple.com/xcode/>.

A GNU compiler is required to compile dynamic applications as shared libraries for the Android NDK. Android make files are designed to run with gcc4. On Windows, a convenient way to have the complete environment prepared for this, is to install Cygwin - currently version 1.7.9-1.

Cygwin uses an installer helper to manage the installation process. Go to <http://www.cygwin.com/> and select "Install or update now!" below the introduction, or use the direct link<http://www.cygwin.com/setup.exe>.

Click Setup.exe and select "Install from Internet" when prompted at "Choose A Download Source" in the installer. We recommend not changing the Root Directory in the next window, and leaving it at "C:\cygwin". The "Local Package Directory" holds the downloaded packages. You may want to keep them with the downloaded Setup.exe in the same directory so as to have a Cygwin installer directory. Choose a download site with a known fast connection near you.

When the package information is downloaded you will see a hierarchical browser to select packages. Select the following package from the hierarchy for download:

All -> Devel -> "make: The GNU version of the 'make' utility"

Select the word "skip" to change it to the actual version number, which is currently 3.81-2. Finish the installation by clicking next.

Your Cygwin environment is fully set-up to work with the FastCV SDK. If you have other similar environments installed, make sure to set your Windows path variable to point to "C:\cygwin\bin" so that bash uses this version of GNU's make.exe.

### Android NDK

The Android NDK is an extension to the Android SDK that lets Android developers build performance-critical parts of their applications in native code. Download the NDK package from

<http://developer.android.com/sdk/ndk/index.html>

The current version is r6.

Unzip the archive and copy the contents into a directory. To be consistent with our previous setup we recommend that you put the contents in "C:\Development\Android\android-ndk-r6\".  Thus Android SDK and Android NDK share the same parent directory. We will later add the FastCV-SDK and your project files here.

NDK-r6 requires the above directory to be added to the Windows path! To do this right-click on 'My Computer' on the desktop and select properties. On the 'Advanced' tabs select 'Environment Variables' and look for Variable 'Path' in the 'System variables' window. After pressing 'Edit', scroll to the end of 'Variable value:' and add the full path to the directory to the end of the path, separated via semicolon from the one before. In the above example, you would have to add:

;C:\Development\Android\android-ndk-r6\

NOTE: Path has semicolon at the beginning. **Do not use pathnames with spaces.** Alternatively, you can also set a User variable with the name Path, but this is only valid for the current user. The last "\" at the end of the  Path variable has to be included.

To test your installation you can compile any of the NDK sample applications. Using a Cygwin bash shell, navigate to the root directory of any demo application (e.g. for the 'san-angeles' sample app with out installation path above):

cd /cygdrive/c/Development/Android/android-ndk-r6/samples/san-angeles

ndk-build

The compiler should produce a dynamically linked library libsanangeles.so and write it to/libs/armeabi within the application directory. NDK-r6 includes support for different architectures so you might find different subdirectories in /libs.

Now your development environment is ready to host FastCV SDK related content.

**MAC**: Update $path variable to point to the ndk directory. Modify the /Users/<login-name>/.tcshrcfile to set the path by adding /Users/<login-name>/Development/Android/android-ndk-r6. Along with the changes applied in the section 'Android SDK Downloader' you will have:

set path = ( $path /Developer/usr/bin /Users//Development/Android/android-sdk-mac\_x86/tools /Users//Development/Android/android-ndk-r6 )

**LINUX**: Update your PATH to point to the ndk directory. If you use bash shell modify the ~/.bashrcfile to set the path by adding /opt/android-ndk-r6. Along with the changes applied in the section 'Android SDK Downloader' you will have:

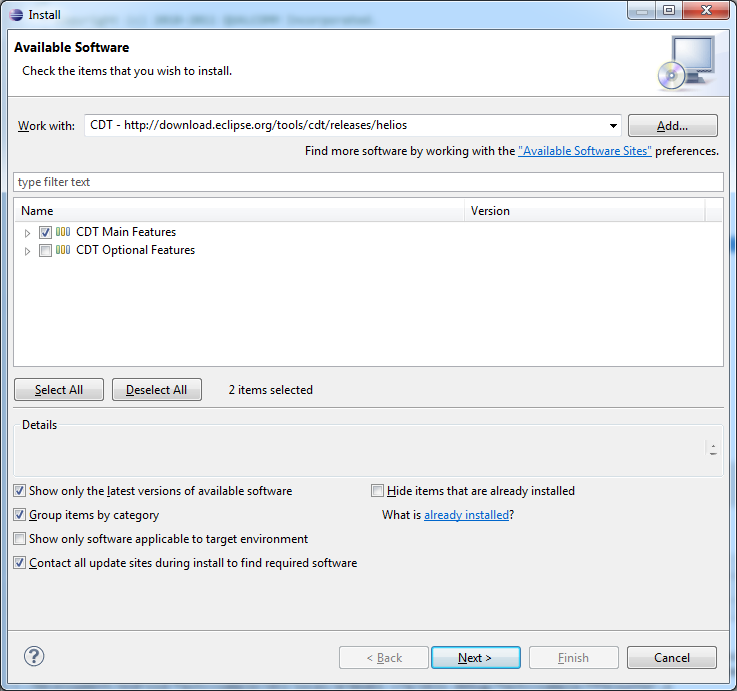
export PATH=~/bin:/opt/android-sdk-linux\_x86/tools/:/opt/android-ndk-r6:$PATH

### C/C++ Development Toolkit (CDT)

To compile the C/C++ portion of FastCV through Eclipse, not only android-ndk, we also need to install CDT. To install CDT, please go to Eclipse -> Help -> Install New Software -> Add

Based on your Eclipse version (can be checked at Help -> About Eclipse SDK or when Eclipse start), use the appropriate url listed below:

* For Eclipse Galileo:<http://download.eclipse.org/tools/cdt/releases/galileo/>
* For Eclipse Helios: <http://download.eclipse.org/tools/cdt/releases/helios/>



# Step 2: Installing the FastCV SDK

## Download

FastCV SDK is distributed through the [Qualcomm Computer Vision (FastCVTM) Portal on Qualcomm’s Devloper Network site.](http://fastcv.qualcomm.com) Follow the instructions below to access the installer.

**NOTE**: Although we distribute installers for the platforms listed below, support for FastCV SDK development is limited to the Win 7 32/64 bit platform.

### Windows

Although we recommend that you develop on Windows 7 32/64bit or Windows XP, the FastCV SDK has also been successfully run on Windows XP 64bit environments. However, we currently do not offer specific support for this platforms.

1. Download installer EXE-file from the Downloads page.
2. Run installer

### Mac OS X

Installation has been tested on Mac OS X 10.6.

1. Download archive file from the Downloads page.
2. Unarchive and run installer

### Linux

Installation has been tested on Ubuntu 10.10.

1. Requires JVM to be installed
2. Download the installer file from the Downloads page.
3. Open terminal window and change directory to the installer file location
4. Execute the installer file on the command line

## Clean Installation

To start developing with the FastCV SDK you will need to:

* Download the install package from fstcv.qualcomm.com/download
* Accept the license agreement
* The installer will ask you which directory you wish to install the FastCV zip file.
* Install the FastCV-SDK in your  <DEVELOPMENT\_ROOT>
* **Note:** *In section 3: Setting up the Development Environment <DEVELOPMENT\_ROOT> = C:/Program Files/Development.*
* Following the recommended installation location the development environment should result in the <DEVELOPMENT\_ROOT> directory structure list below. Here xx-yy-zz and xx.yy.zz stand for the version number of the FastCV SDK.

<DEVELOPMENT\_ROOT>

Android-ndk-r6\

Android-sdk-windows\

Fastcv-android-xx-yy-zz\

inc\

lib\

*readme.txt /\*Starting Readme documentation\*/*

*readme\_License.txt /\*License information\*/*

*Fastcv-android-sample-app-xx-yy-zz\*

*inc\ /\*Commented Header Files \*/*

*jni\ /\*Native Source Code\*/*

*lib\ /\*Static Library\*/*

*res\ /\*Additional assets required within the FastCV sample application\*/*

*src\ /\*Java Source Code\*/*

*readme.txt /\*Starting Readme documentation\*/*

This above directory structure will ensure that sample apps can be easily built and deployed using the Android NDK and the Eclipse Java developer environment. This directory structure keeps the FastCV binary file and the FastCV Sample application in separate trees. This will ensure easier updates to the SDK, leaving the sample application tree, and additional application trees untouched.

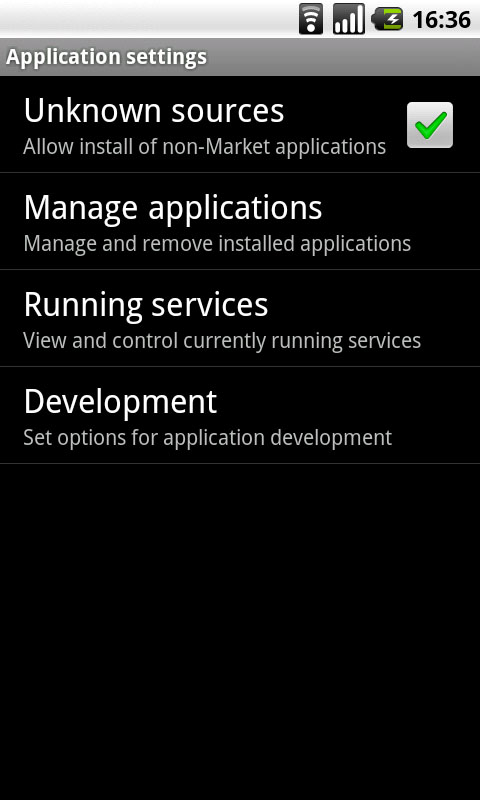
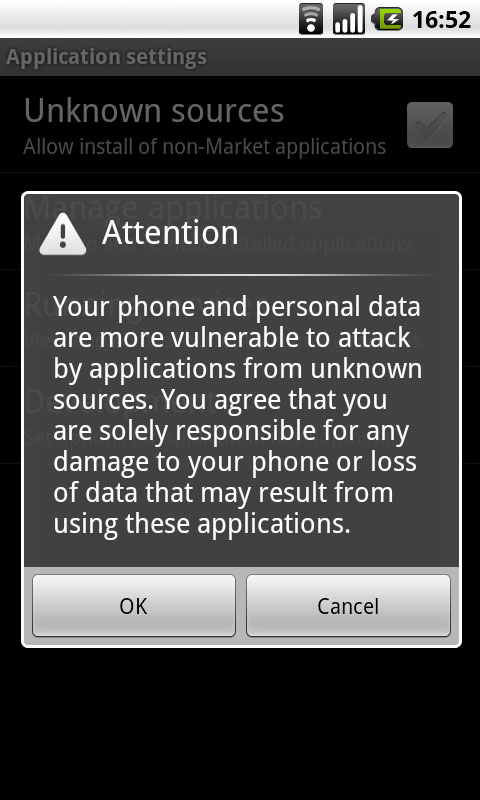
## Prepare Test Device for Development

### Developer Settings on the Phone

Android phones require special settings for development. In this step we will

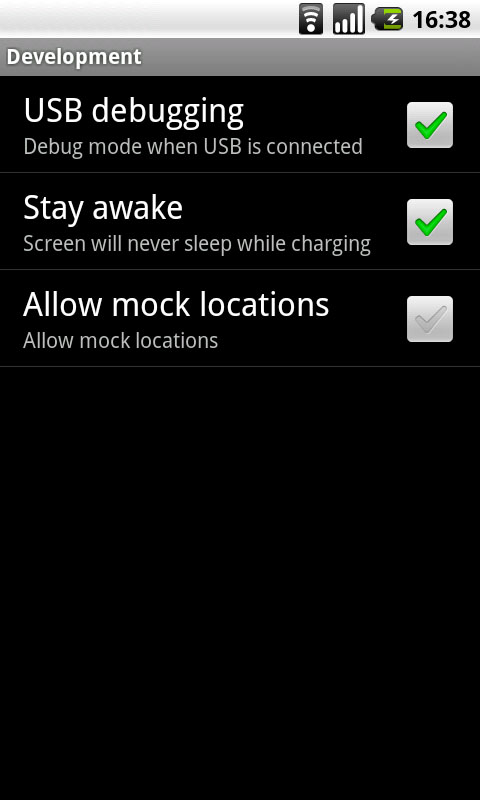
* Allow apps to be installed from unknown sources
* Enable USB debugging

On the phone, go to Settings->Applications and choose "Unknown sources" as shown below. Accept the warning shown on the right. This setting allows the direct installation of unsigned APKs from within Eclipse.

Allowing unknown sources for software installation

Go to the Development dialog on the above screen and choose the two settings as shown below:



USB debugging is mandatory. The stay awake setting will help with development.

### Install the USB Driver

Connect your phone to the development PC via the USB cable.

On the first connection, Windows will recognize the phone as a new device and ask for a driver. The Android SDK already includes the appropriate USB drivers - compatible with Nexus One. In the Windows device driver dialog, select manual install without Internet connection and browse to the following directory

<DEVELOPMENT\_ROOT>\android-sdk-windows\usb\_driver

When the device installation finishes you are ready to use your test device.

On some machines the USB driver installation will ask you to reboot the machine. You can skip this step and the phone should be accessible without a reboot.

**NOTE**: You should contact the phone manufacturer to get the correct USB driver.

# Step 3: Compiling and Running a FastCV Sample App

## Create the Android APK & Compiling Shared object

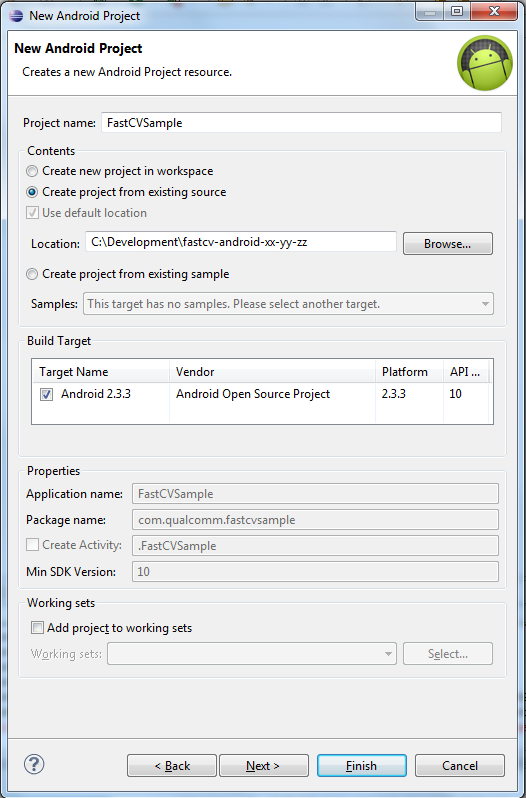
We will now compile and build the Android application APK using the Eclipse IDE environment. The sample applications have several Java classes to create the provide GUI functionality, read the camera frame, allow user interaction, and provide performance data to the user

### Building the FastCVSample application

Create a new project in Eclipse by selecting File->New->Project.... Choose Android->Android Project from the Wizard selection.

Select 'Create project from existing source' and browse to the <DEVELOPMENT\_ROOT>\fastcv-android-xx-yy-zz\ directory.

Click Finish.



Eclipse "New Android Project" panel

Adjust the project name and click Finish

To compile the native part, we’ll need to add C/C++ attributes to the project.

To do so, use clock on **File > New > Other > C/C++ > Convert to a C++ Project** menu option to add C/C++ attributes to the project.

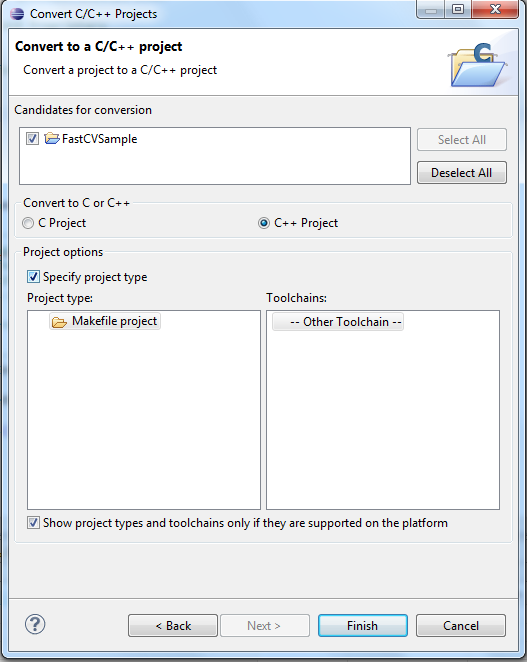
**NOTE:**

Contrary to what the text of the menu option implies, this does not replace the Android application attributes with C/C++ application attributes, nor does it create a new project in place of, or side-by-side with, your Android application. It only adds additional attributes for C/C++ to the project, alongside the Android attributes.

The wizard invoked to add C/C++ attributes for your project will ask you to configure the project type and toolchains to use for the project. By selecting **Makefile project** for the project type, and **Other Toolchain** for the toolchains, you configure the project to invoke [GNU Make](http://www.gnu.org/software/make/) to build the C/C++ portion of your project.

This is where you will be integrating the two build systems (the Java part using the ADT, with the C/C++ part using the NDK).

The makefile you will create will invoke the **ndk-build** script from the NDK to take care of building the native shared object for your application, and then the ADT will automatically include that shared object when it constructs your application package.



The standard installation of Eclipse has automatic compilation turned on. The very first time you compile the app, you may receive some error messages from Eclipse. Project->Clean...->Clean All Projects should fix these errors as Eclipse has to build some standard directories. This also creates the APK package for deployment.

Copy lib/libfastcv.a to your ndk lib directory (e.g <Android-NDK-Root/platforms/<Android API>/arch-arm/usr/lib)

Copy the contents of inc directory to the ndk include directory (e.g <Android-NDK-Root/platforms/<Android API>/arch-arm/usr/include)

Now when you click on buildall, you should see something like follows in the Console Log:

\*\*\*\* Build of configuration Default for project FastCVSample \*\*\*\*

make all

Host OS was auto-detected: windows

make[1]: Entering directory `/cygdrive/c/Development/fastcv-android-xx-yy-zz'

Gdbserver : [arm-linux-androideabi-4.4.3] libs/armeabi/gdbserver

Gdbsetup : libs/armeabi/gdb.setup

Compile++ thumb : fastcvsample <= fastcvsample.cpp

Compile++ thumb : fastcvsample <= FPSCounter.cpp

Compile++ thumb : fastcvsample <= CameraRendererRGB565GL2.cpp

Compile++ thumb : fastcvsample <= CameraUtil.cpp

StaticLibrary : libstdc++.a

SharedLibrary : libfastcvsample.so

Install : libfastcvsample.so => libs/armeabi/libfastcvsample.so

make[1]: Leaving directory `/cygdrive/c/Development/fastcv-android-xx-yy-zz'

The apk should be inside the \bin\ directory. The app has been successfully built.

## Run the Sample Application

To install on your device, run the following:

Adb install fastcvsample.apk

This should install the sample app on the device, and you should be able to run the app.

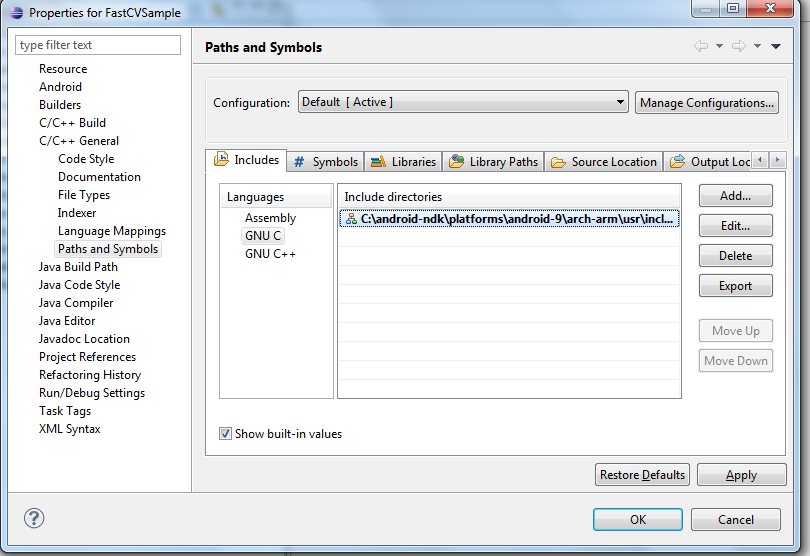
# Tips/Tricks:

## Tip 1:

If you get an error saying C/C++ primitives not found, you might have to include the include folder of your ndk directory under project properties.

To do so, right click on Project->Properties->C/C++ General->Paths and Symbols.

Under GNU and GNU C++, add the Android NDK include directory



## Tip 2:

If you get an error saying ndk-build not found, make sure you have GNU make 3.8.1 or higher installed. You can check version of GNU make by typing in make –v

## Tip3:

If you get an error saying ndk-build not found

make all   
make: /ndk-build: Command not found  
make: \*\*\* [all] Error 127

Add ANDROID\_NDK\_ROOT at your environment variable. To do this, at Eclipse IDE, goes to Project -> Properties -> C/C++ Build -> Environment, then add

Variable: ANDROID\_NDK\_ROOT

Value: <path to android ndk directory>

## Tip4:

Please check each functions detailed requirements in the Doxygen comments at:

<http://Fastcv.qualcomm.com/FastCVAPI>