

phonegap: Android:Getting started with Android PhoneGap in Eclipse

Google seems to be favoring this wiki page over the [getting started page](#), and [in more detail here](#) which is more up to date and have more precompiled packages ready for you to use. The information below may still be useful for some, so it is being left as is.

This guide will help you setup your first android project with **PhoneGap 0.9.1**, if you are using the latest from GitHub then refer to the readme there.

The first thing to do is to download and install all the required dependencies.

Please contribute platform specific instructions where needed. Some cleanup wouldn't hurt either.

Prerequisite Software

PhoneGap requires you have the following software installed:

1. Eclipse Integrated Design Environment (IDE), a program to write code and manage software development
2. Java Development Kit (JDK), the Java compiler and support tools
3. The Android Software Development Kit (SDK), base code and emulators for developing Android apps
4. Android Development Toolkit (ADT), a plug-in for Eclipse that makes Android development seamless
5. Apache Ant, a program that automates software build processes (similar to 'make')
6. Ruby, a programming language
7. Git Bash, an alternative command program for Windows environments (Windows only)

If you've previously done any development in the Android environment, you probably have 1-4 already installed. If you have not, it is strongly recommended you visit <http://developer.android.com/index.html> to better understand Android development and create your own, "Hello, World" example prior to jumping into PhoneGap. A great resource exists at: <http://developer.android.com/sdk/installing.html> for new developers.

Step 1: Installing the Prerequisite Software

1A. Eclipse

Install the *Eclipse IDE for Java Developers* from <http://www.eclipse.org/downloads/>. This is a ZIP archive file, extract the folder "eclipse". Most other flavors of Eclipse (like [Aptana](#) or platform specific builds) will also work, provided that you install all the necessary Eclipse plug-ins.

1B. Java JDK

If you do not already have the Java JDK installed on your system, install the Java SE Development Kit - JDK 6 from here: <http://java.sun.com/javase/downloads/index.jsp>

If you're running Windows this is a standard installer file, just follow the on screen instructions for installation.

1C. Android SDK

Install the Android SDK Package from here <http://developer.android.com/sdk/index.html>

Extract the downloaded archive and place it somewhere you'll remember and add the tools sub directory your PATH. Additional information and instructions for adding the Android SDK to your PATH can be found at: <http://developer.android.com/sdk/installing.html>

Be sure to do the following (Windows):

1. If you install into a folder containing a space (c:\Program Files), you must refer to it by its truncated name in the command program. In this case "c:\progra~1". The truncated name will be the first six characters + ~ + a sequential number starting with 1 for each instance of the same six characters.
2. Add android SDK to your Path
3. Add android sdk tools (\android-sdk-windows\tools) to your path. This is necessary for ruby to find the command "android create project" later on.

Notice: PhoneGap 1.0 no longer supports Android 1.x (API levels 3 and 4). Make sure you download and use API level 7 (Android 2.1-u1) or newer.

1D. ADT Plug-in for Eclipse

Install the ADT Plug-in for Eclipse as described here: <http://developer.android.com/sdk/eclipse-adt.html>

To verify your installation of Eclipse, Android SDK, ADT, and Java, you can build a sample "Hello World" application using the instructions located here:

<http://developer.android.com/guide/tutorials/hello-world.html>.

1E. Apache ANT

Visit: <http://ant.apache.org/index.html>

- Windows -
 - If you install into a folder containing a space (c:\Program Files), you must refer to it by its truncated name in the command program. In this case "c:\progra~1". The truncated name will be the first six characters + ~ + a sequential number starting with 1 for each instance of the same six characters.
 - Download and unzip binaries
 - add the unzipped folder path to your PATH
- OS X - it comes with XCode otherwise "sudo port install apache-ant".

1F. Ruby

Visit: <http://rubyinstaller.org> for an automatic installation (Windows only)

- Windows - download and install Ruby from the official site <http://www.ruby-lang.org/en/downloads/> or use the RubyInstaller (**recommended**) here <http://rubyinstaller.org/>
- OS X - nothing to see here.

1G. Git Bash (Windows only)

Visit: <http://code.google.com/p/msysgit/downloads/list>

Download and install the latest full installer for official Git

Visit: <http://git-scm.com/> for additional information about Git

Note: During installation, you may select the "Use Git Bash only" option

Step 2: Confirm Environment Variables

Environment variables are data stored by your system; in this case we are interested in the file locations of the various developer tools. You should have these environment variables:

- ANDROID_HOME, location of the Android SDK (for example: D:/android-sdk-windows)
- ANT_HOME, location of Android Ant (for example: D:\apache-ant-1.8.1)
- JAVA_HOME, location of the JDK (for example: C:\Program Files\Java\jdk1.6.0_20)
- Path, folders where the operating system always looks when given a command (includes these, for example: C:\Ruby191\bin; D:\apache-ant-1.8.1\bin; C:\Program Files\Java\jdk1.6.0_20\bin; D:\android-sdk-windows\tools)

You can confirm the environment variables the following way:

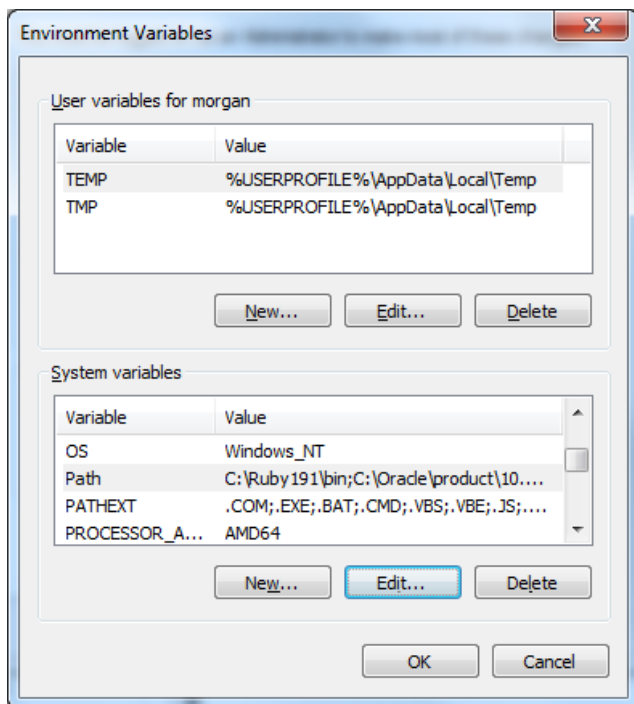
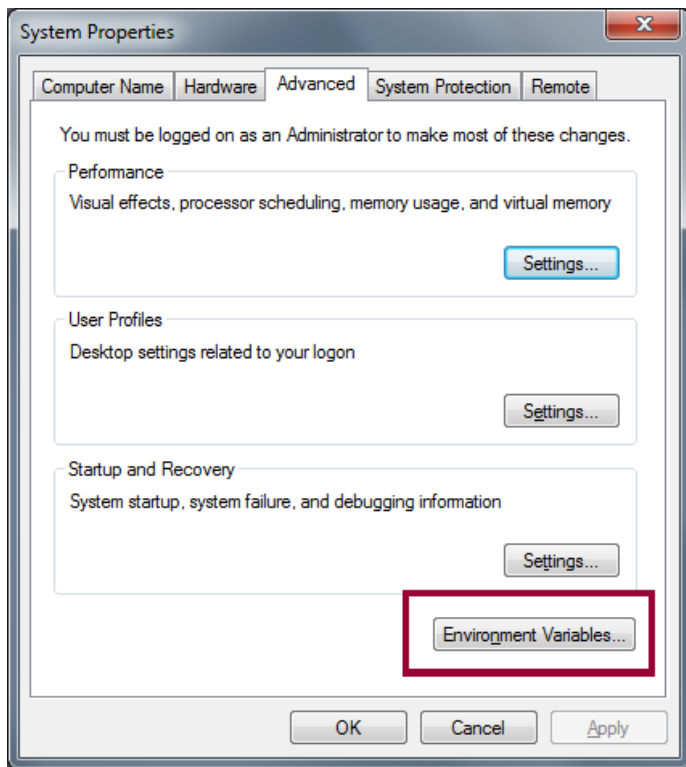
Windows:

2A. Right-click on My Computer (or Computer)

2B. Select Properties -> Advanced tab -> Environment Variables button).

2C. "Edit..." the Path to include the needed folders. Use a semicolon (;) to separate folders.

Hint: You can copy the path from the address bar in Windows Explorer.

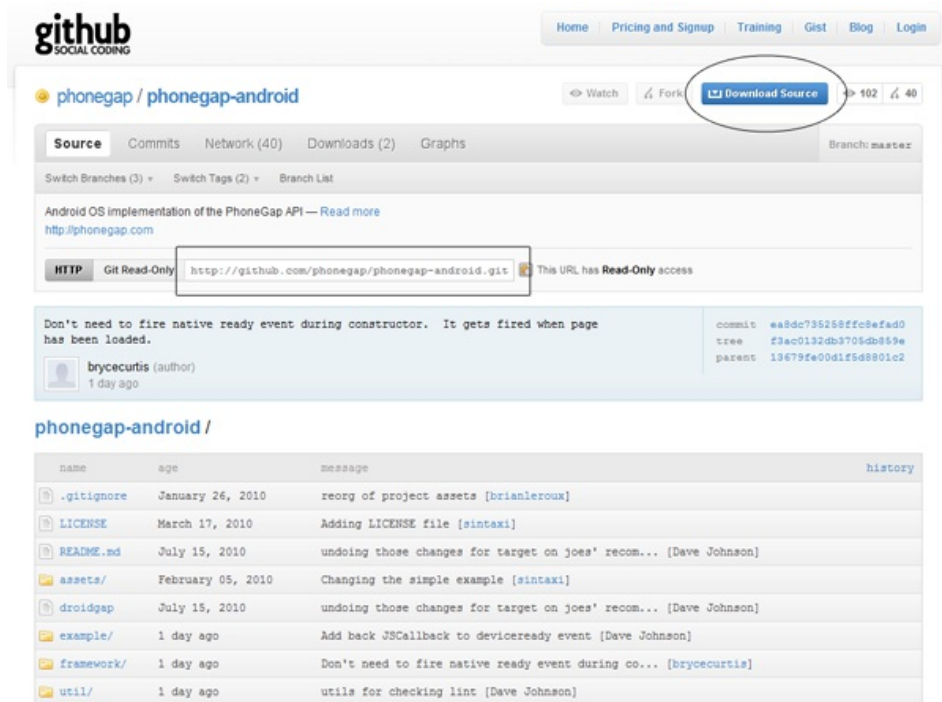


If everything looks good in the environment variables, try opening a command prompt (Start > Run > cmd) and typing in "ant", "ruby", "java", or "android". If you get the error message "x was not recognized as an internal or external command" you need to check your PATH value. If the program continues to execute you can use ctrl-c to terminate it.

Step 3. Download the PhoneGap Source Code

First you will need to download the latest PhoneGap code. It is hosted on github.com. The easier way to get the code is to download it directly from the github website.

3A. Visit: <http://github.com/callback/callback-android>



Alternatively, you may use a dedicated 'git' client to download the code or you may use Eclipse. This guide will not cover either of those methods.

3B. Unzip the files into a folder.

Step 4. Build the Example PhoneGap App

Now we will use Ruby and a script called 'droidgap' that is provided with PhoneGap to package up our example app.

This will create two outputs; a file called 'PhoneGap.jar' and an Android Eclipse project, which we will then import into Eclipse.

4A. Open a command window and change directory ('cd') to the 'phonegap-android' folder. In Windows, use git bash as the command window. In Mac OS, use Terminal.

4B. Run the following command. This will create a PhoneGap.jar file and create a basic Android Eclipse project in the output path that can then be opened in Eclipse.

NOTE: The droidgap script has been updated and now there are two methods to create the necessary files. The new script is significantly simpler to use than the old method. For those who prefer the old method, it is still available.

NEW METHOD:

4C. From the 'phonegap-android' folder, run 'ruby bin/droidgap create [path_of_application]'. For the example app, the command would be 'ruby bin/droidgap create example'. If the example folder does not exist inside 'phonegap-android', run 'ruby bin/droidgap gen' first.

The new create command in the droidgap script will generate the Eclipse project, ready to be imported. The results of the script can be found at '[path_of_application]_android'. Example: If [path_of_application] = '~/my_app', then droidgap create will place results at '~/my_app_android'. The for the example app, the results will be placed at '../phonegap-android/example_android'

Skip ahead to Step 5 for Eclipse import instructions.

Run 'droidgap help' to see other new commands including run, log, test and ship.

OLD METHOD:

Previous users of the droidgap script please note that droidgap must now be invoked with the 'classic' command to use this method.

```
ruby bin/droidgap classic "[android_sdk_path]" [name] [package_name] "[www]" "[path]"
```

android_sdk_path ... The path to your Android SDK install. (**Must** use forward slashes (/) in path)

name The name of your application.

package_name The name of your package (For example: com.nitobi.demo). Can not be just the app name, must contain at least one "."

(i.e. "com.example").

www The path to your www folder. (Where your HTML, CSS and JavaScript code is located)

path The path to generate the application. (**CANNOT** be inside **ANY** Eclipse workspace) (**Must** use forward slashes (/) in path)

You should get a message in your terminal saying completed!

Example (Windows environment):

```
$ ruby ./droidgap classic "C:/Progra~1/Androi~1/android-sdk-windows" pgtest com.example.android.pgtest example/
"c:/Progra~1/Androi~1/phonegap-android/output"
```

In the above Windows example, the following environment settings were used:

android_sdk_path = "c:\Program Files\Android SDK\android-sdk-windows"

name = pgtest

path = "c:\Program Files\Android SDK\phonegap-android\output" (This is the output location where the example files were placed)

Example (Mac environment):

```
> cd phonegap-android
> ruby ./droidgap classic android/android-sdk-mac_86/ test_app com.mayerdan.testapp example/
~/projects/test_droid_phonegap
```

Note if you get errors like:

BUILD FAILED

~/projects/phonegap-android/framework/build.xml:49: taskdef class com.android.ant.SetupTask cannot be found

followed by more errors like:

0:in `stat': No such file or directory - ~/projects/phonegap-android/framework/phonegap.jar

To fix this, just run the command with the full path to the android SDK

"sdk.dir=/andriod/android-sdk-mac_86" => "sdk.dir=/projects/andriod/android-sdk-mac_86"

Then run `ruby ./droidgap classic /Users/danmayer/projects/android/android-sdk-mac\_86 test\_app com.mayerdan.testapp example/ ~/projects/test\_droid\_phonegap` again.

Some notes:

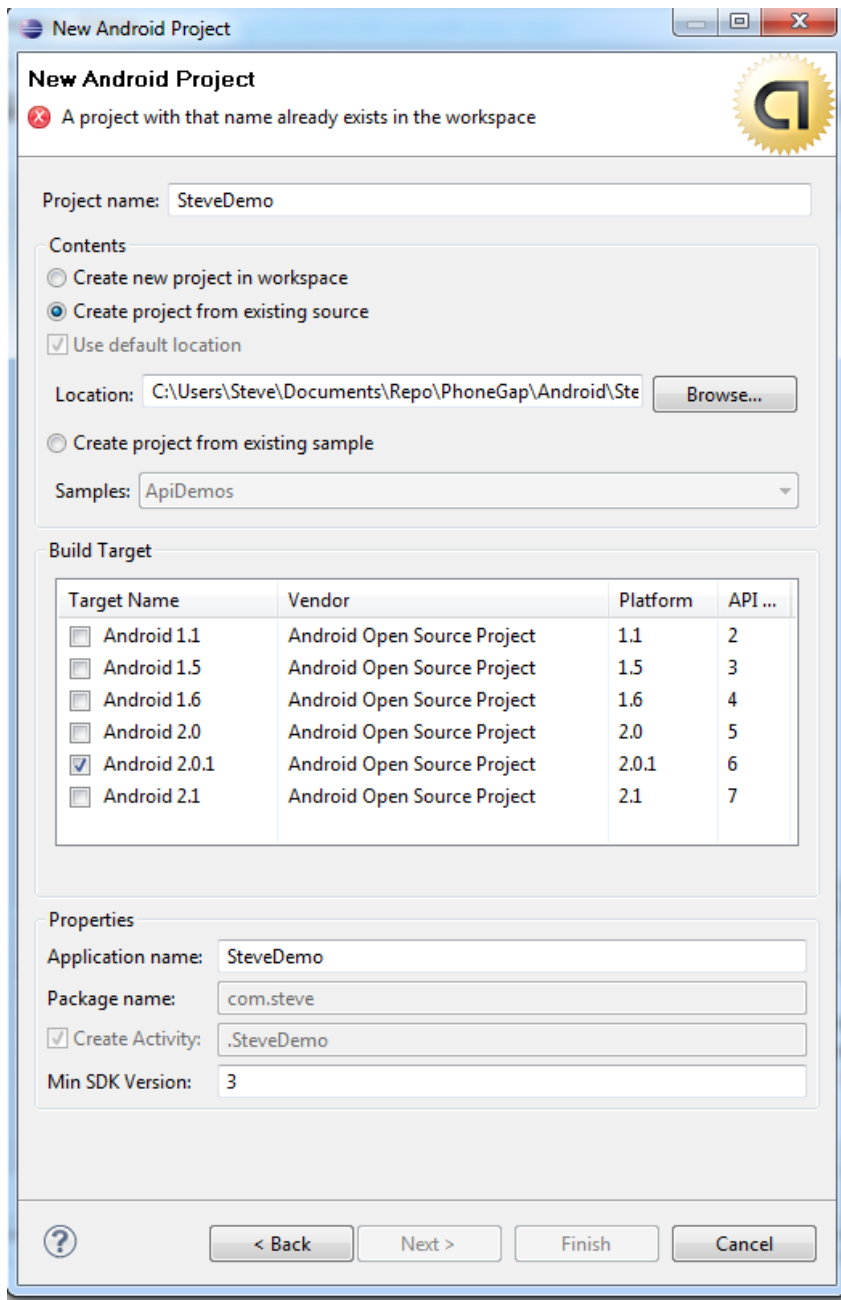
- Your output path should not be inside your Eclipse workspace. **See note in Step 5*
- Make sure the path you set to generate the application doesn't exist
- For the www folder, phonegap-android comes with a folder called example that includes some sample html, css and js.
- I found I was getting errors for not having set android or ant in my path. Make sure you followed the earlier steps.
- When pointing to a directory, you need to have quotes around the path "C:/mobile/....."
- While in the command prompt, you should manually change all the backslashes (\) to forward slashes (/). This helps the droidgap script with its syntax. **E.g. The android_sdk_path argument must use forward slashes (/).**
- If you install into a folder containing a space (c:\Program Files), you must refer to it by its truncated name in the command program. In this case "c:\progra~1". Otherwise, the script will look for "c:\Program" and fail.
 - Note: The truncated name will be the first six characters + '~' + a sequential number starting with 1 for each instance of the same six characters. For instance if you have two folders c:\PhoneGapA and c:\PhoneGapB, the former's truncated name will become, "c:\phoneg~1" and the latter's will become, "c:\phoneg~2".
- For Windows users: The droidgap script is written to expect UNIX-style syntax, which is why you need to use 'git bash' to execute these commands. 'Bash' is a shell language and the 'git bash' application can handle the expected syntax. Alternatively you may use another bash-style command application, such as 'cygwin', but if you know what that is, you know how to install and use it.
- Do not have "tools" as a folder in your path to the android sdk folder; C:\Users\Daniel\Dev\tools\android-sdk won't work.
- If at any point you need to check your PATH variables in git bash you can use the command "echo \$PATH"

Step 5. Importing Your Example App into Eclipse

Now time to create your project. In Eclipse, go to File > New project > Android > Android Project

Now select create project from existing source and navigate to the new folder that was generated in the last step. Select a build target and press finish.

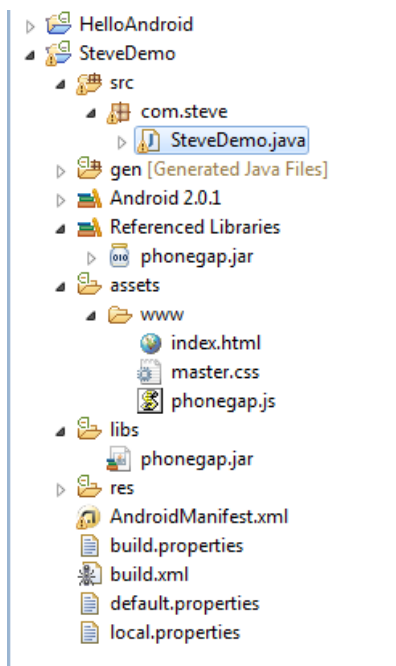
***NOTE:** The folder you specified in the Ruby script will actually be the folder that your Eclipse will save to, and compile from and NOT your current Eclipse Workspace. Saving your folder to a location inside of a Workspace will not allow you to create the project because the .metadata folder. If you have pointed your folder to one of your Eclipse Workspaces, simply cut and paste the folder to a location that is outside of any Workspace.



Now, in Eclipse you should see your project

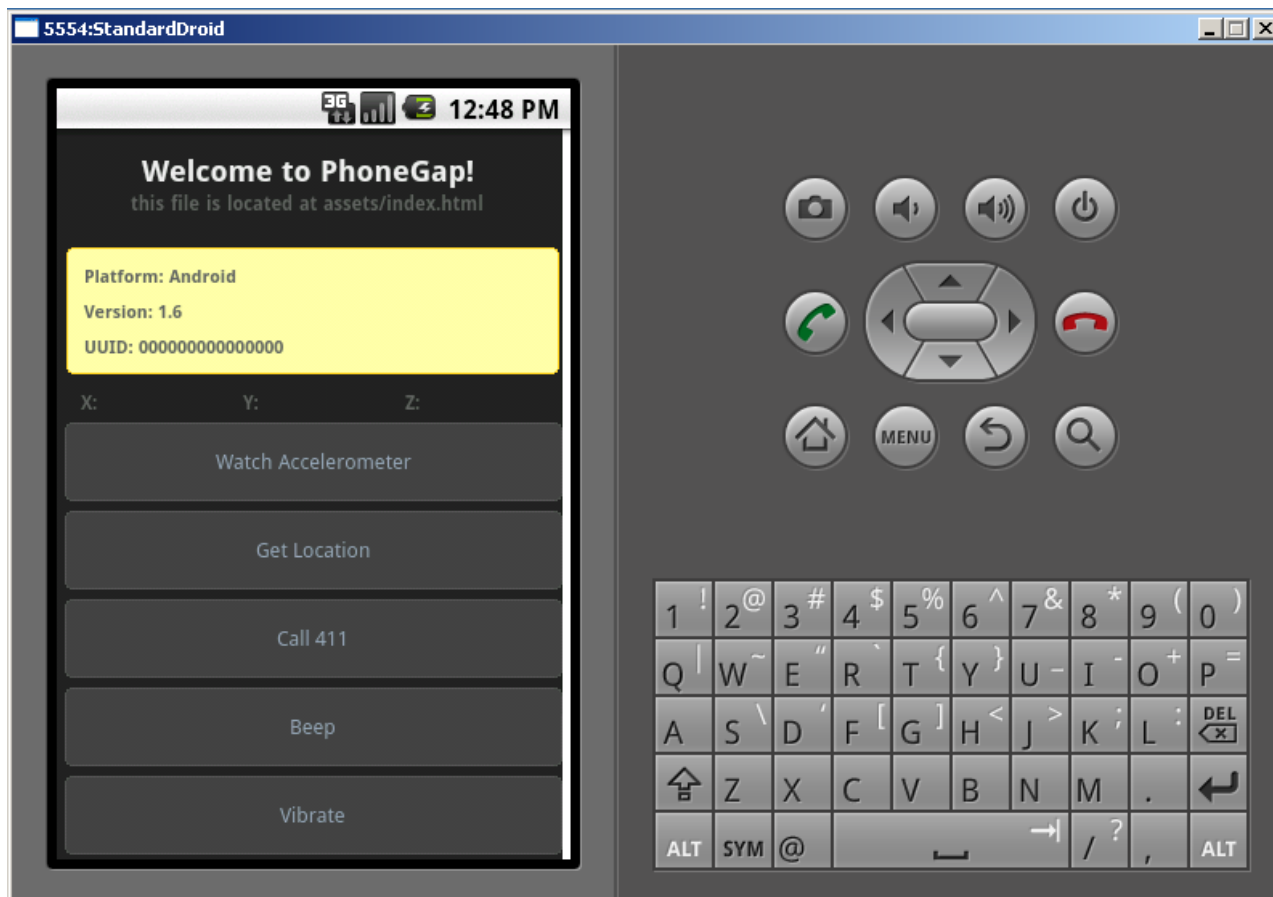
Rightclick phonegap.jar under the libs folder and go to Buildpath -> Add to build path.

Note: Now the folder structure should look like the one shown below.



Click on the project folder again and then the Run button in Eclipse, it is the button that looks like "Play". When you run this project, you should see the dialog box to Run As..., choose Android Application. You may get the unable to launch dialog, this can happen if you have the Android 1.5 or some other file highlighted when you try to run the project. Just select the project folder and click run.

If you have done everything correctly, there should be an Application that loads. It should be noted that testing with an actual device is preferable to the Emulator for many reasons, namely the fact that it's easier to test the Accelerometer, Geolocation, Beep and Vibrate Functionality. However, for testing on the actual browser browser events, and various issues with Webkit on Android the Emulator does just as well.



Notes

- Perhaps JAVA_HOME does not point to the JDK. Or maybe JAVA_HOME doesn't exist. To fix this you must right click on my computer, go to advanced system settings, click the advanced tab, press the environment variables button, Under system variables click new and add JAVA_HOME if it doesn't exist and make the path to the JDK that was installed earlier.

- ERROR:**

```
droidgap:48:in ``: No such file or directory
android create project -t 7 -k be.progone.test -a PG1 -n PG1 -p E:/baud/work/Mobile/Pr_Eclipse/PG1/build
(Errno::ENOENT)
from droidgap:48:in `create_android'
```

SOLUTION:

Add the Android SDK tools path to the system. Follow steps above to get to system environment variables.
Eg: C:\Program Files\Android SDK\tools

- ERROR:**

```
"BUILD FAILED
C:\phonegap-android\framework\build.xml:49: taskdef class com.android.ant.SetupTask cannot be found"
```

SOLUTION:

Try going to the "framework" folder and check the path in the local.properties file. See if its correct and pointing to your android sdk directory.

You **MUST** use a forward slash in the directory (Windows users) otherwise the paths will not match up.

Eg: C:/Program Files (x86)/Android SDK

- ERROR:**

"Click on various buttons on the example app, doesn't do anything" or some things are working but I can't tell what is going on, or I want more information how do I view the Android logs?

SOLUTION:

To get more information about what is going on while phonegap is running, run the Android log viewer. `adb logcat` if adb isn't found you just need to do the full path to it something like `~/projects/andriod/android-sdk-mac\_86/tools/adb logcat`. If you have both a device and a Android emulator running use -d and -e like so `adb -e logcat`. This should give you more info about what is going on. Here is a small example clipping from the android logs.

```
D/PhoneGapLog( 291): file:///android_asset/www/phonegap.js: Line 216 : TypeError: Result of expression 'accel'
[undefined] is not an object.
E/Web Console( 291): TypeError: Result of expression 'accel' [undefined] is not an object. at
file:///android_asset/www/phonegap.js:216
D/dalvikvm( 117): GC_EXPLICIT freed 1727 objects / 91840 bytes in 110ms
D/PhoneGap( 291): The provider network is disabled
D/SntpClient( 61): request time failed: java.net.SocketException: Address family not supported by protocol
```

From this we can see that the accel is not working. And some information about the location of the Error. This helps debug problems and also helps you to give more information when asking the PhoneGap community for help.