There are several steps required to get Phonegap 3.0+ working with Eclipse, and the documentation is not always clear..

Here is what worked for me:

0: Install the required Android tools and SDK

Before you try getting phonegap to work, make sure you can build a normal android application from Eclipse. Select File→new→Project and then 'Android Application Project'. Put 'test in the application name and then just keep hitting next. You should be able to then run the basic application from the toolbar.

1: Install NodeJs

Go to http://nodejs.org to install NodeJs. Make sure you are running a recent version, I had some issues when using an outdated one.

To confirm the installation worked, type

```
npm -v
```

and make sure you see a version number.



2: Install cordova (phonegap)

One of the confusing things is that there are both 'Phonega' and 'Cordova', and the documentation does not make this very clear. You can install either one and they are almost interchangable, but not 100%. I would recomment installing Cordava as that is what most of the documention uses (although ironically not the installation docs). Nota para Mac: no olvidarse de utilizar sudo para evitar problemas de permisos.

To install cordova run the following command:

```
npm install -g cordova
```

Para instalar una versión anterior de cordova:

```
npm install -g cordova@{numero_de_version}
```

Para ver las diferentes versiones de cordova:

```
npm view cordova versions
```

I had to run the command multiple times before it would succeed, but that might have just been a connection issue.

You also have to install plugman

```
npm install -g plugman
```

Instalar ant

```
npm install -g ant
```

En el path de windows introducir la ruta a la carpeta /bin de ant, y a las carpetas /tools y /platform-tools del sdk de android.

3: Create test project

Now that you have codova installed it is time to create a test project:

```
cordova create hello com.example.hello HelloWorld
```

Go to the project directory

```
cd hello
```

and then add the android platform:

```
cordova -d platform add android
```

Añadir la plataforma IOS (Es necesario hacerlo desde MAC, ya que requiere el XCODE):

```
cordova -d platform add ios
```

Añadimos el plugin para saber si hay conexión

```
plugman install --platform android --project . \ --plugin
https://git-wip-us.apache.org/repos/asf/cordova-plugin-network-information.g
it#r0.2.5
```

It might take a while the first time, if all goes well your project should be read to build:

```
cordova build
```

Gestión de errores

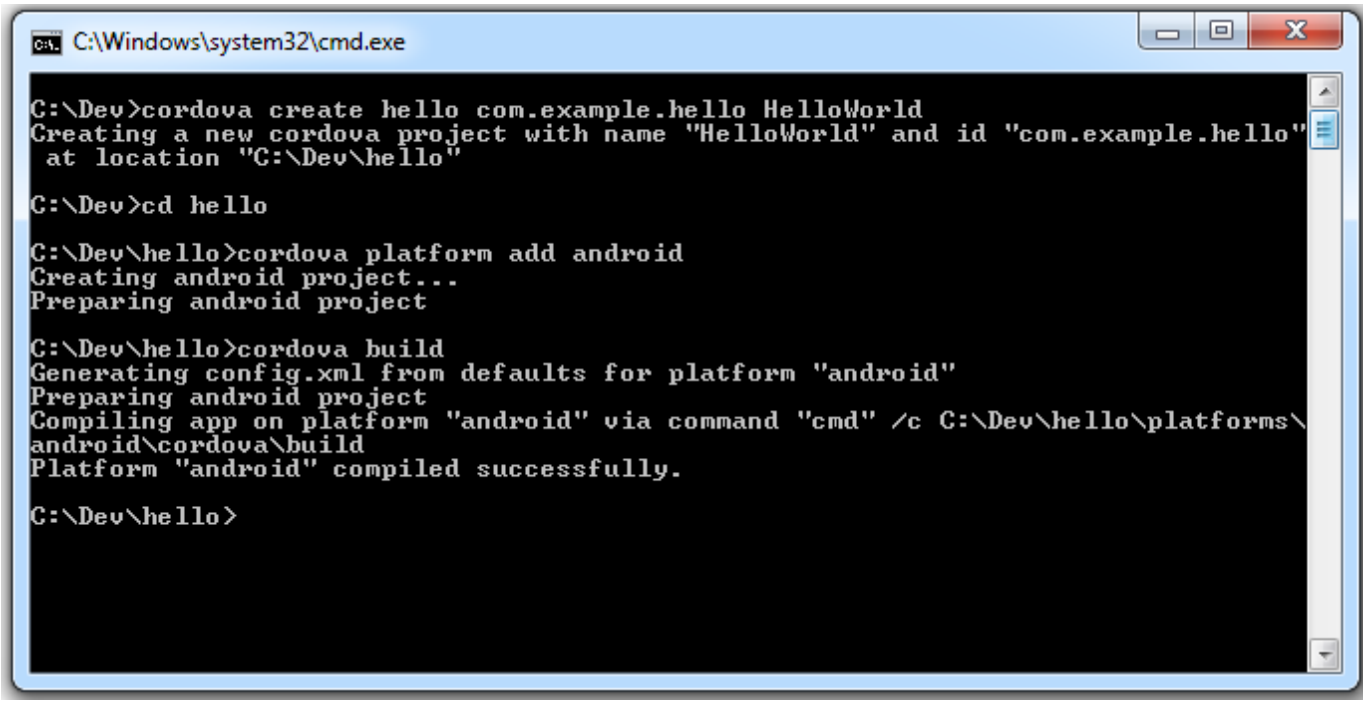
Si hubiera errores y fuera necesario rehacer le proyecto, el comando es (Atención!!!!!!! esto elimina

el proyecto, así que previamente hacer copia de seguridad):

```
cordova platform remove android
```

Y lo volvemos a crear de nuevo, de forma que los plugins que no se hayan automatizado se cargan de nuevo:

```
cordova platform add android
```

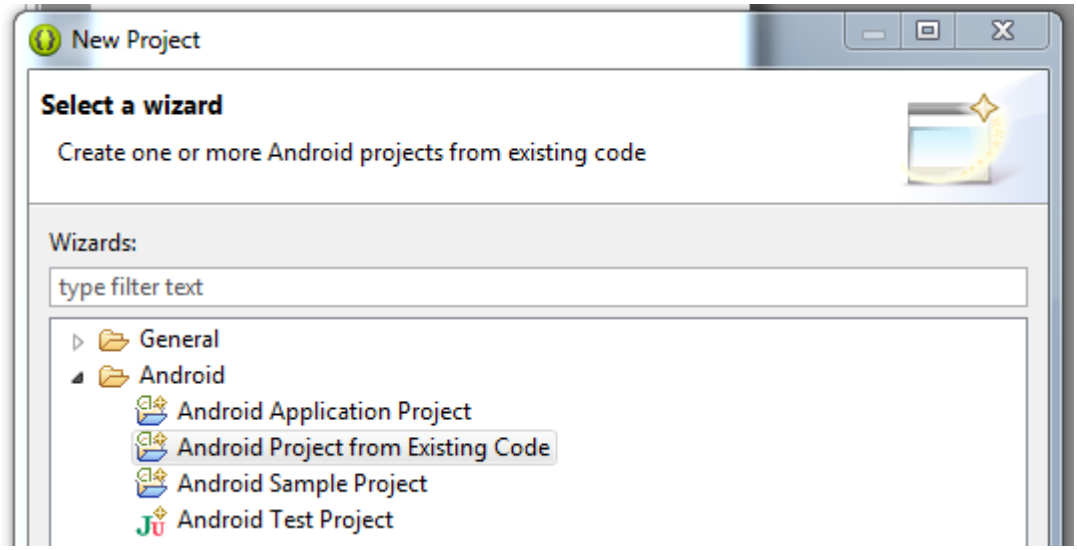


4: Import the project in Eclipse

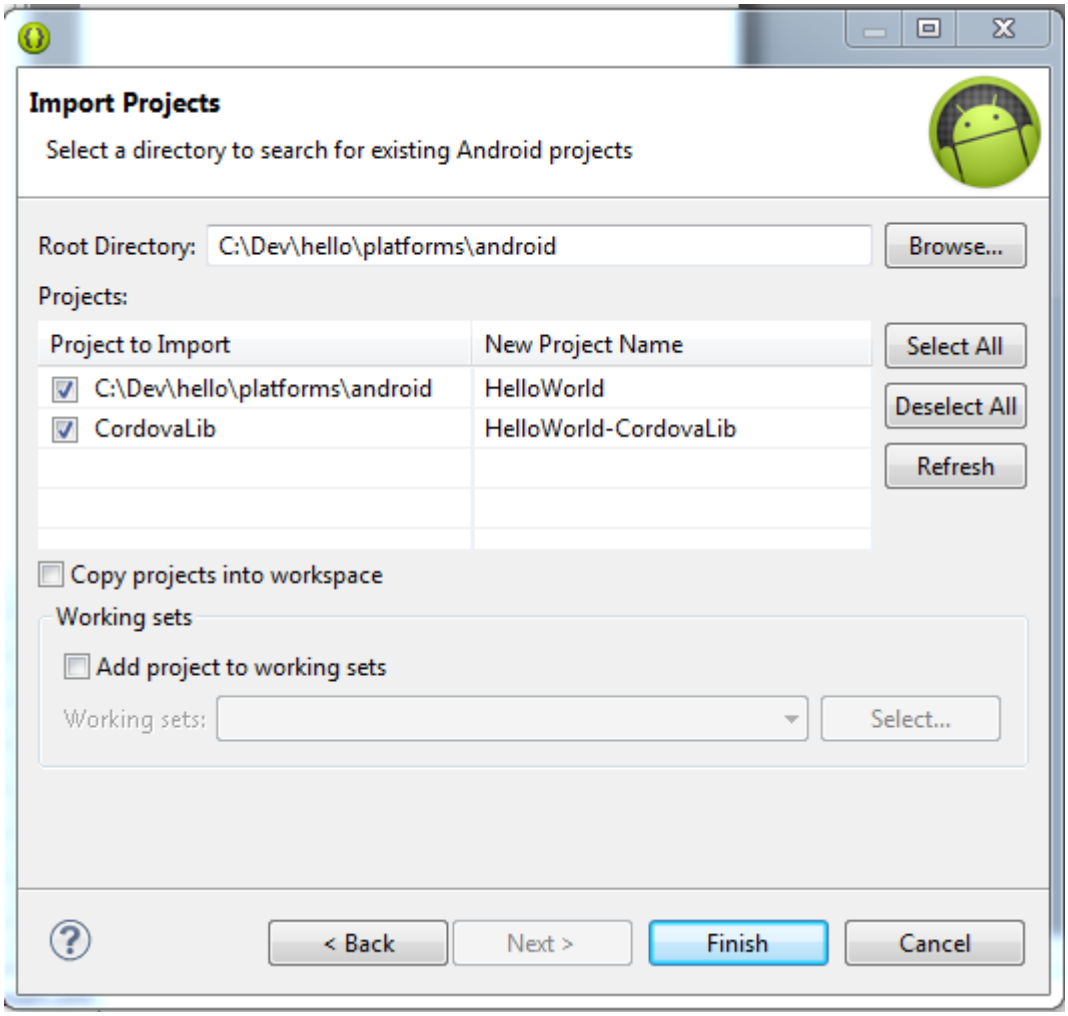
The Eclipse project files are now located in the project/platforms/android directory. It is recommended that you do not edit the assets directly in the platforms directory, but rather edit the top level www files (project/www). Otherwise if you run 'cordova prepare' it will overwrite your changes in the project directory.

To import in Eclipse:

Create a new project, and select 'AndroidProject from Existing Code'



Browse to your project directory, and select the platforms/android folder and hit finish:



The project should now open in Eclipse, but there are some issues to fix before it will compile. Most importantly it is missing the cordova-3.x.x.jar file that should be in the libs/ folder.

```
package com.example.hello;

import android.os.Bundle;
import org.apache.cordova.*;

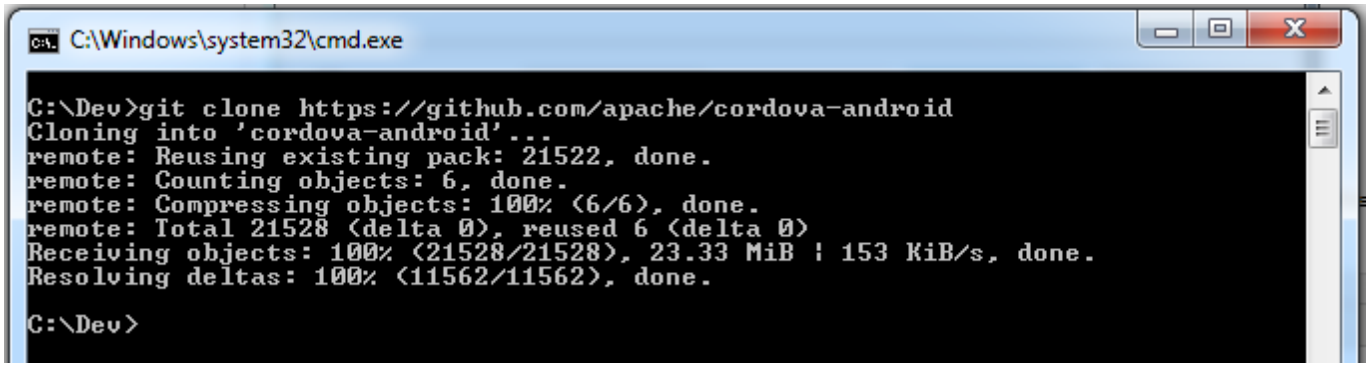
public class HelloWorld extends CordovaActivity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        super.init();
        // Set by <content src="index.html" /> in config.xml
        super.loadUrl(Config.getStartUrl());
        //super.loadUrl("file:///android_asset/www/index.html")
    }
}
```

To get the missing jar file, you need to download the Cordova-Android code from

https://github.com/apache/cordova-android

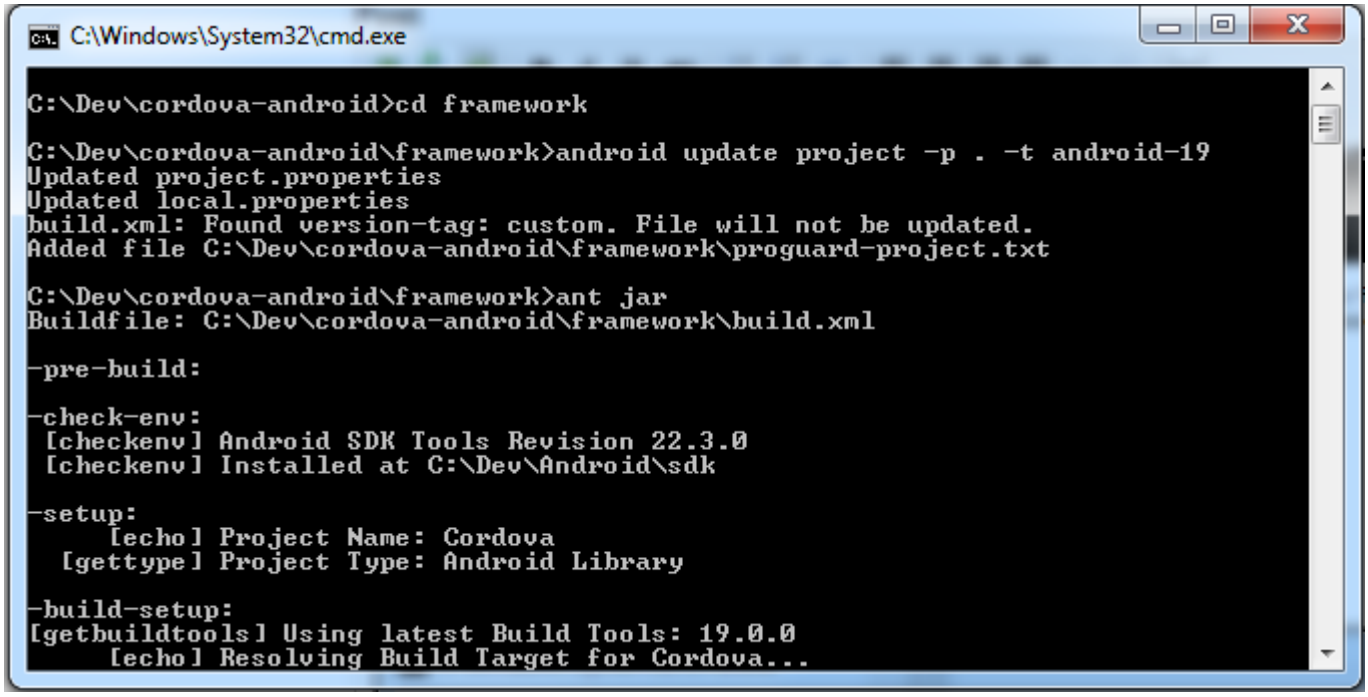
If you have git installed you can type

```
git clone https://github.com/apache/cordova-android
```



Go to the framework directory in the project, and type these commands to build the jar file (you will need to have ant installed to generate the jar file, but it should be on your system already):

```
android update project -p . -t android-19
ant jar
```



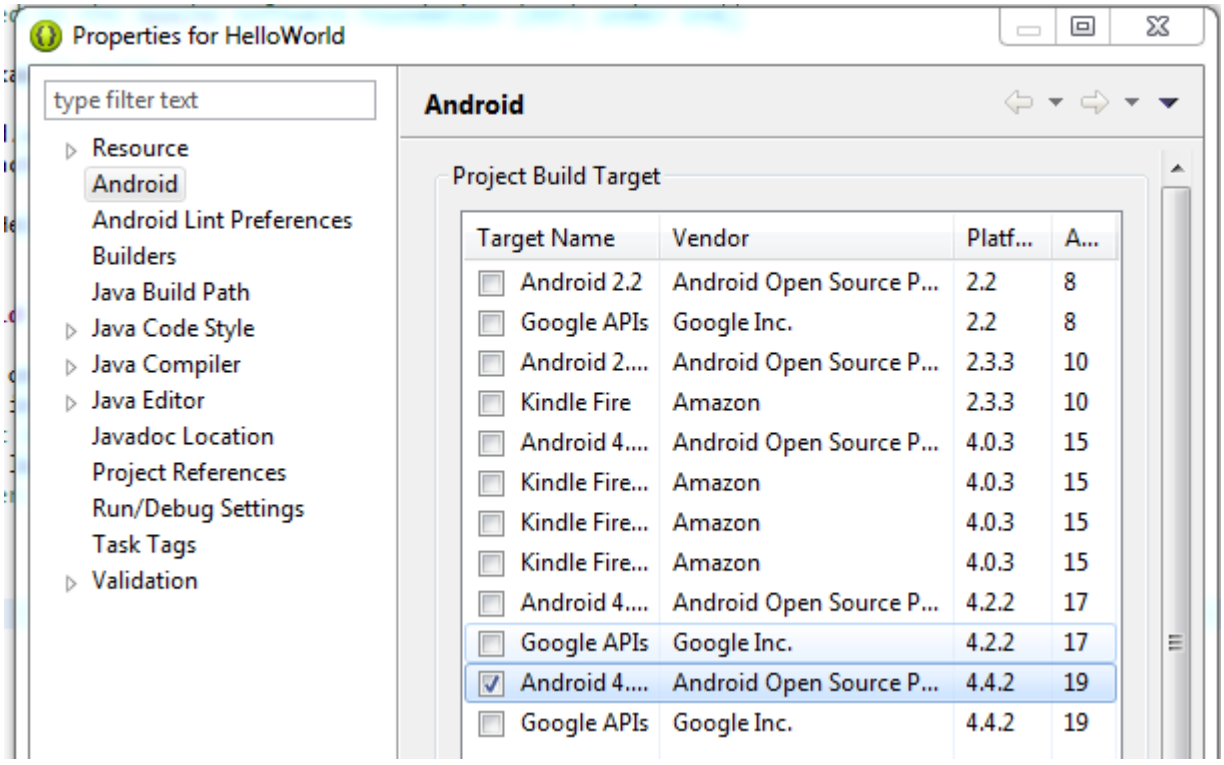
The cordova-3.3.0-dev.jar file should now be in the framework directory. Copy/Paste this file to your libs directory in Eclipse, then choose Project→Clean..

The project should now build. If you get an error like

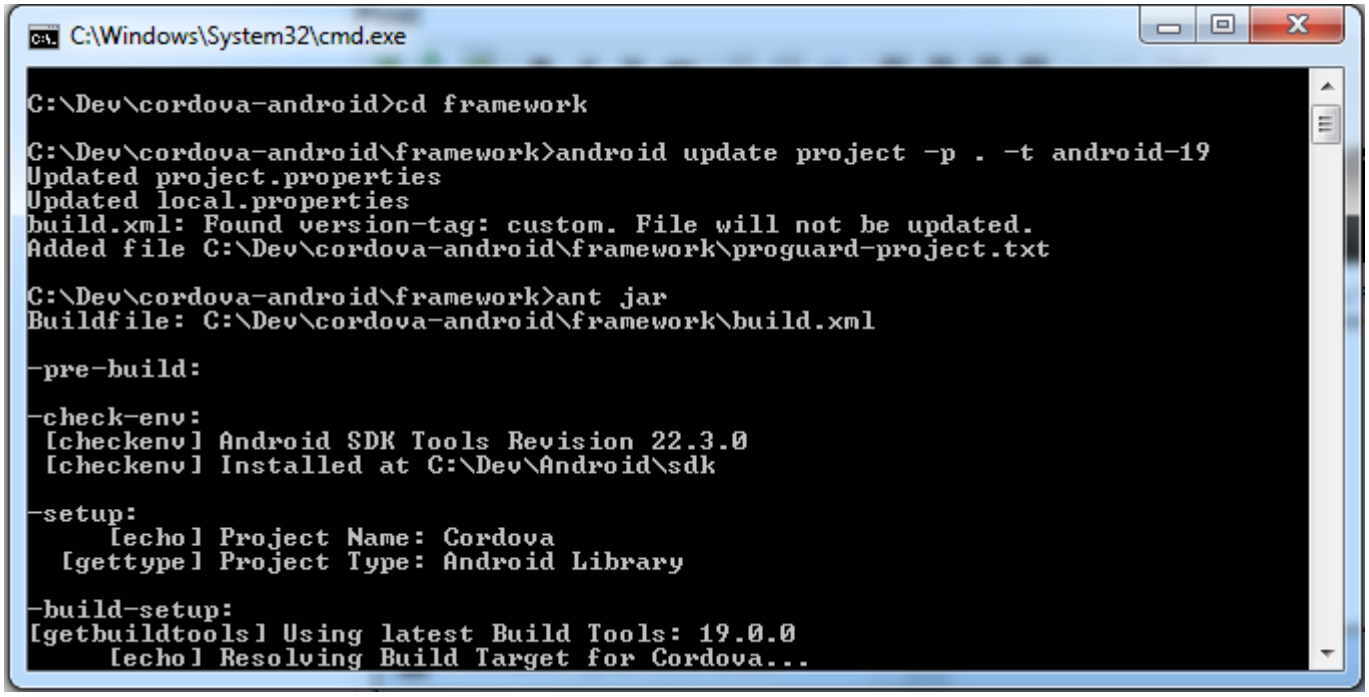
No resource identifier found for attribute 'hardwareAccelerated'

then make sure you are using the latest Android api version:

In Eclipse, right click on the project, select properties and choose the latest Android api version:



That is it, you should now be able to comple and run the phone application on your phone or in the simulator!



5: Desarrollo de la aplicación

Antes de anda, cometnar que todo lo que hacemos aquí está pensado para hacerlo con CHROME.

Lo primero es instalar la extensión ripple de google para emular mobiles.

Para poder probar el CROS (cross scripting) para acceder al webservice en otro dominio, es necesario que la página se vea con http, por lo que necesitamos arrancar el apache con la información del proyecto.

Cada vez que queramos hace un build, previamente tenemos que hacer un prepare para cargar en assests/www de cada plataforma la versión del código actualizada:

```
cordova prepare
```

Posteriormente ya podemos hacer:

```
cordova build
```

From:
https://wiki.merkatu.info/ - **Wiki de merkatu**

Permanent link:
https://wiki.merkatu.info/phonegap-cordova?rev=1416829715

Last update: **2017/03/27 17:43**