

Installing and configuring PhoneGap/Cordova

# **Installing and configuring PhoneGap/Cordova**

## **EntwicklerCamp 2012**

# Installing and configuring PhoneGap/Cordova

## Table of Contents

Lab objectives.....	3
Time estimate.....	3
Prerequisites.....	3
Getting started.....	3
Setting up PhoneGap.....	3
Your first PhoneGap project.....	8
Testing your application.....	16
Summary.....	16
End of exercise.....	16



## Installing and configuring PhoneGap/Cordova

### ***Lab objectives***

The purpose of this lab is to install and configure an IDE for building hybrid applications with PhoneGap.

### ***Time estimate***

You should be able to complete the lab in around 45 minutes.

### ***Prerequisites***

#### **Appropriate Skill Level:**

- No prerequisite skills required

#### **Computer Specification:**

- Android SDK installed
- Java SDK installed
- Internet connectivity

### ***Getting started***

This lab includes the following:

- Installing and configuring the IDE
- Building and testing your first PhoneGap application

### ***Setting up PhoneGap***

In this lab, we will install the following components:

- Eclipse Indigo
- Eclipse Plugin for Android Development Tools
- Apache ANT
- Ruby
- PhoneGap

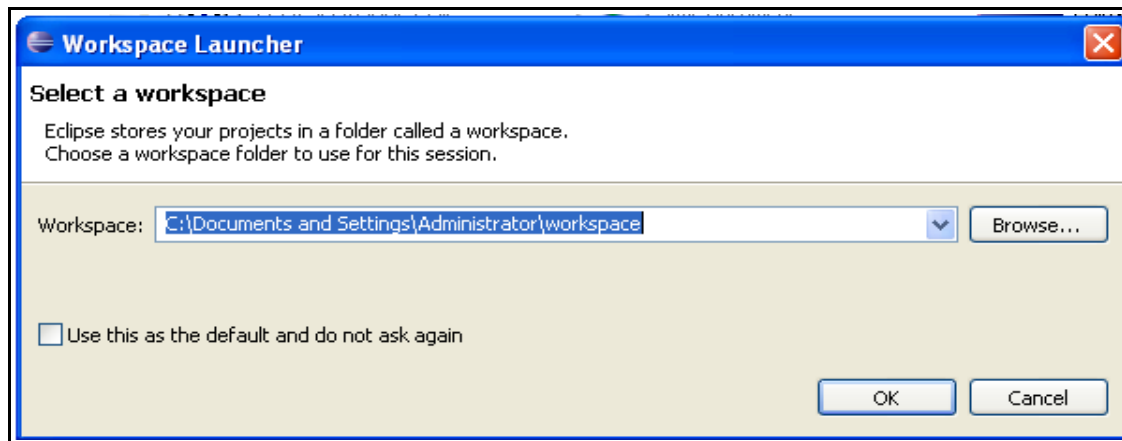
Your moderator will provide you with the required install files for all of these components.

## Installing and configuring PhoneGap/Cordova

1. Unzip the eclipse-java-indigo-SR1-win32.zip file.
2. Unzip the phonegap-phonegap-1.5.0-0-gde1960d.zip file.

Step 3 to 5 is optional – because we skip the step of compiling our own packages here and use the precompiled stuff instead.

3. Unzip the apache-ant-1.8.2-bin.zip file.
4. Run rubyinstaller-1.9.3-p125.exe to install Ruby.
5. Run Git-1.7.9-preview20120201.exe to install GIT. Do a default installation without changing anything.
6. Start Eclipse and accept the default workspace:

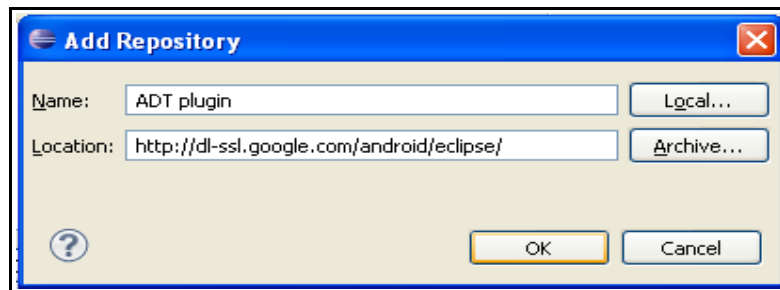


## Installing and configuring PhoneGap/Cordova

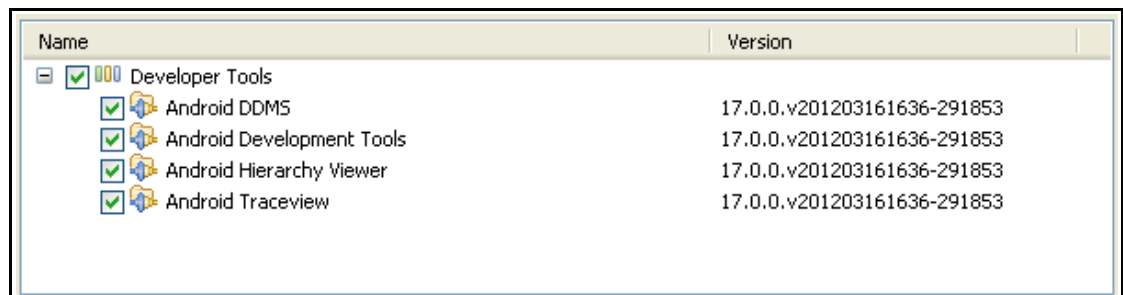
### Downloading the ADT Plugin

7. Use the Update Manager feature of your Eclipse installation to install the latest revision of ADT on your development computer:
  - 7.1. Start Eclipse, then select **Help > Install New Software....**
  - 7.2. Click **Add**, in the top-right corner.
  - 7.3. In the **Add Repository** dialog that appears, enter "**ADT Plugin**" for the Name and the following URL for the Location:

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



- 7.4. Click **OK**
- 7.5. In the **Available Software** dialog, select the checkbox next to **Developer Tools** and click **Next**.

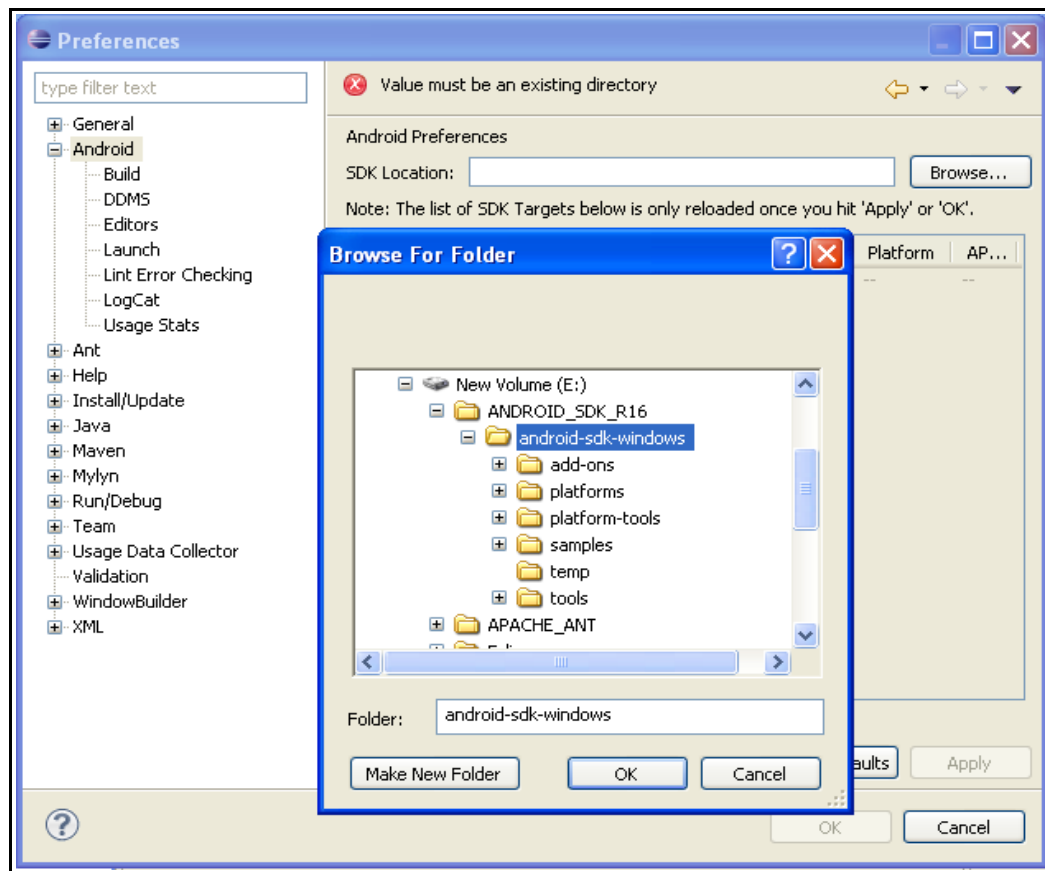


- 7.6. In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
  - 7.7. Read and accept the license agreements, then click **Finish**.
8. When the installation completes, restart Eclipse.

## Installing and configuring PhoneGap/Cordova

### Configuring the ADT Plugin

9. Now we need to modify your ADT preferences in Eclipse to point to the Android SDK directory:
  - 9.1. Select Window > Preferences... to open the Preferences panel.
  - 9.2. Select Android from the left panel.
  - 9.3. You may see a dialog asking whether you want to send usage statistics to Google. If so, make your choice and click Proceed. You cannot continue with this procedure until you click Proceed.
  - 9.4. For the SDK Location in the main panel, click Browse... and locate your downloaded SDK directory.



- 9.5. Click **Apply**, then **OK**. If finished, exit Eclipse.

## Installing and configuring PhoneGap/Cordova

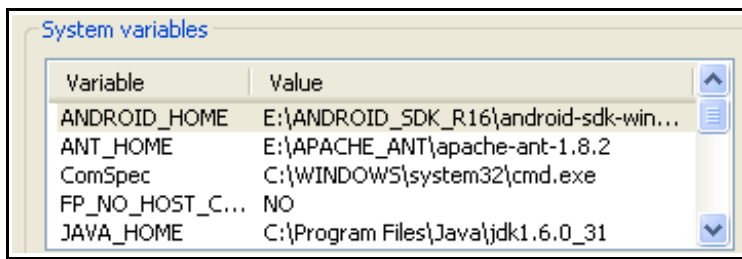
### Modifying your environment variables

Prepare your environment by extending your **PATH** by the following:

- your\_system\_path/jdk/bin
- your\_system\_path/android-sdk/tools
- your\_system\_path/ruby/bin
- your\_system\_path/apache-ant/bin

You will also need to have the following attributes defined:

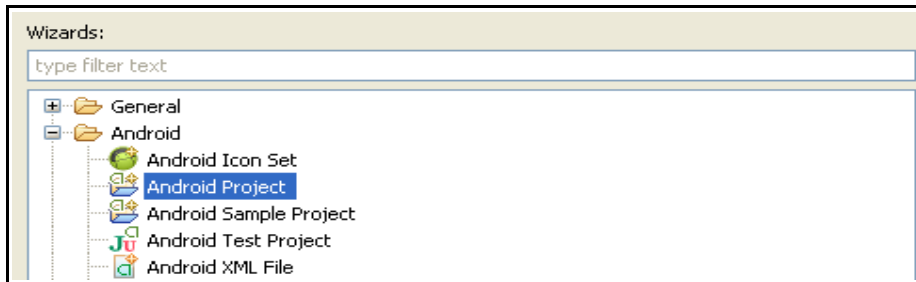
- JAVA\_HOME – path of your JDK directory
- ANT\_HOME – path of you apache-ant directory
- ANDROID\_HOME – path to your android SDK directory.



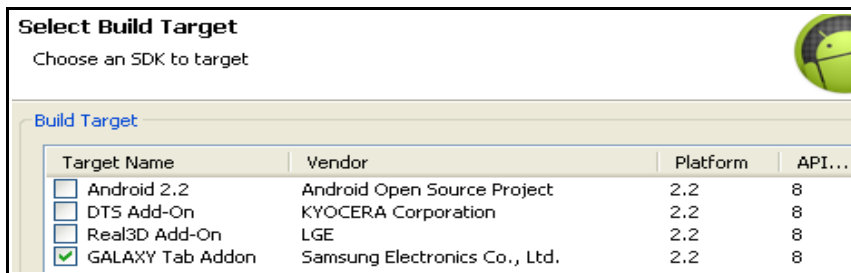
## Installing and configuring PhoneGap/Cordova

### Your first PhoneGap project

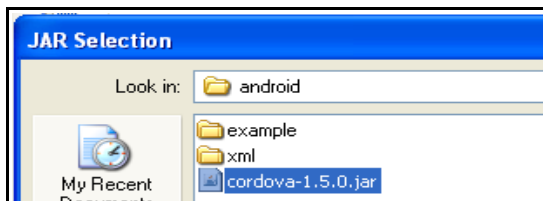
1. Start Eclipse and create a new Android project via **File > New > Other > Android > Android Project**.



2. Provide a name for the new project and click **Next**.
3. Select the **GALAXY Tab Addon** as the Build Target and click **Next**.



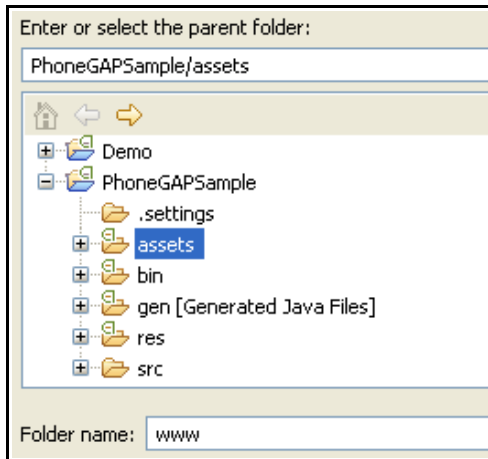
4. Provide a Package Name (e.g. demo.phonegap.sample) and click Finish.
5. Now we need to add the PhoneGap library to the project's build path. Right click on the project and select **Build Path-> Add external archive**.
6. Select the **cordova-1.5.0.jar** from your libs folder and click Open.



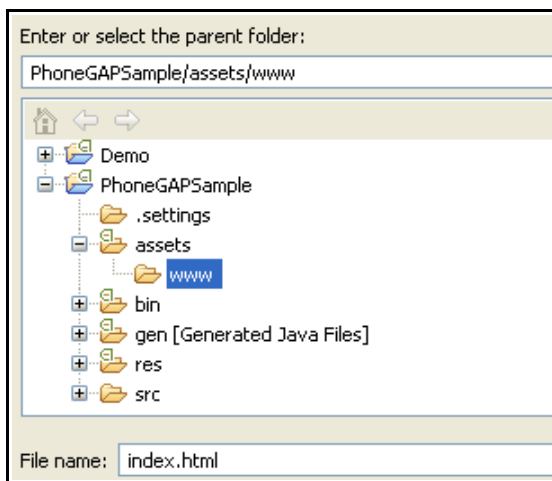


## Installing and configuring PhoneGap/Cordova

7. Right-click on the project's **assets** directory and create a new folder **www** within.



8. Within the **www** folder, create a new file **index.html**.



9. Open the **index.html** file with a text editor (**Right-click > Open with > Text Editor**).

## Installing and configuring PhoneGap/Cordova

### 10. Add the following code here:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Capture Photo</title>
    <script type="text/javascript" charset="utf-8" src="cordova-
1.5.0.js"></script>
    <script type="text/javascript" charset="utf-8">
var pictureSource; // picture source
var destinationType; // sets the format of returned value

// Wait for PhoneGap to connect with the device
//
document.addEventListener("deviceready",onDeviceReady,false);

// PhoneGap is ready to be used!
//
function onDeviceReady() {
    pictureSource=navigator.camera.PictureSourceType;
    destinationType=navigator.camera.DestinationType;
}

// Called when a photo is successfully retrieved
//
function onPhotoURISuccess(imageURI) {
    // Uncomment to view the image file URI
    // console.log(imageURI);

    // Get image handle
    //
var largeImage = document.getElementById('largeImage');

    // Unhide image elements
    //
largeImage.style.display = 'block';

    // Show the captured photo
    // The inline CSS rules are used to resize the image
```

## Installing and configuring PhoneGap/Cordova

```
//
largeImage.src = imageURI;
}

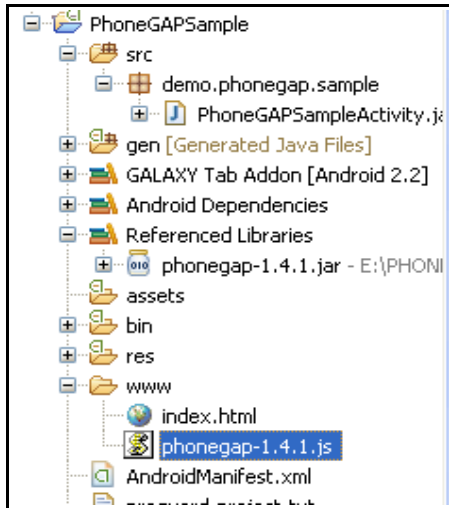
// A button will call this function
//
function getPhoto(source) {
    // Retrieve image file location from specified source
    navigator.camera.getPicture(onPhotoURISuccess, onFail, { quality: 50,
        destinationType: destinationType.FILE_URI,
        sourceType: source });
}

// Called if something bad happens.
//
function onFail(message) {
    alert('Failed because: ' + message);
}

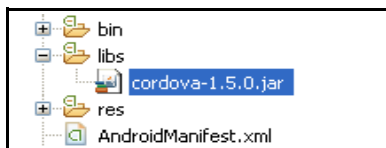
</script>
</head>
<body>
    <button onclick="getPhoto(pictureSource.PHOTOLIBRARY);">Access my Photo
    Library</button><br>
    <button onclick="getPhoto(pictureSource.SAVEDPHOTOALBUM);">Access my Photo
    Album</button><br>
    <img style="display:none;" id="largeImage" src="" />
</body>
</html>
```

## Installing and configuring PhoneGap/Cordova

11. Add the **cordova-1.5.0.js** to your **www** folder (via Right-click on the folder > **Import > General > File System**).

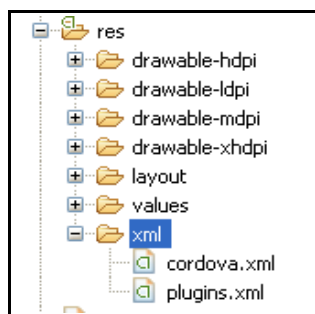
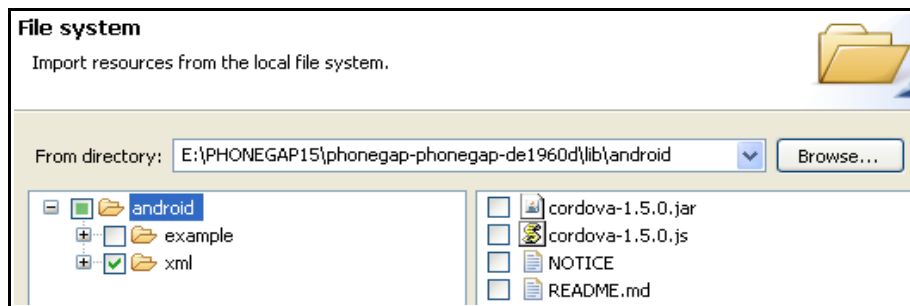


12. Add a new **libs** folder on project root level. Import the **cordova-1.5.0.jar** file into this folder.



## Installing and configuring PhoneGap/Cordova

13. Import the **xml** folder from your PhoneGap download into the **res** directory.



14. Open the Java class that has been generated on project creation.

```
package demo.phonegap.sample;

import android.app.Activity;

public class PhoneGAPSampleActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

## Installing and configuring PhoneGap/Cordova

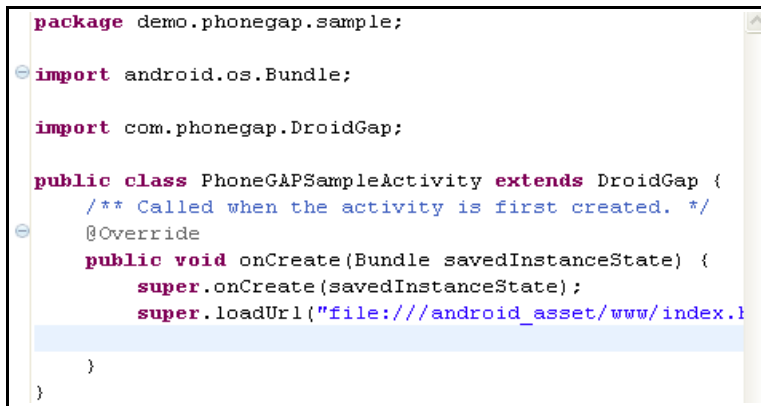
15. Change the code as follows:

Our class should extend **DroidGap** instead of **Activity**.

Remove the **setContentView** line and add the following here:

```
super.loadUrl("file:///android_asset/www/index.html");
```

16. Hit **STRG+SHIFT + O** to organize your imports and remove the issues.



```
package demo.phonegap.sample;

import android.os.Bundle;

import com.phonegap.DroidGap;

public class PhoneGAPSampleActivity extends DroidGap {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

17. **Save** your code.

18. Navigate to the **AndroidManifest.xml** and open it with a text editor.

Within this file, navigate to the

```
android:versionName="1.0" >
```

line and add the following set of permissions afterwards:

```
<supports-screens android:largeScreens="true"
android:normalScreens="true" android:smallScreens="true"
android:resizeable="true" android:anyDensity="true" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
<uses-permission
android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
```

## Installing and configuring PhoneGap/Cordova

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission
android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
```

### 19. Add the following activity declaration:

```
<activity
    android:name="com.phonegap.DroidGap"
    android:label="@string/app_name"
    android:configChanges="orientation|keyboardHidden">
    <intent-filter>
    </intent-filter>
</activity>
```



```
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity
        android:name=".PhoneGAPSampleActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.phonegap.DroidGap"
        android:label="@string/app_name" android:configChanges="orie
        <intent-filter>
        </intent-filter>
    </activity>
```

## Installing and configuring PhoneGap/Cordova

### ***Testing your application***

1. Right-click on our project and select **Run As > Android Application**.

Note that the device starts more quickly than the first time and takes you wherever you were on the device at the time you closed it down.

2. Your application should start, providing access to the local photo album & library of the device.



### **Summary**

In this lab, you installed and configured the PhoneGap development environment. You built and tested your first application

### ***End of exercise***