

## 95-702 Android Getting Started

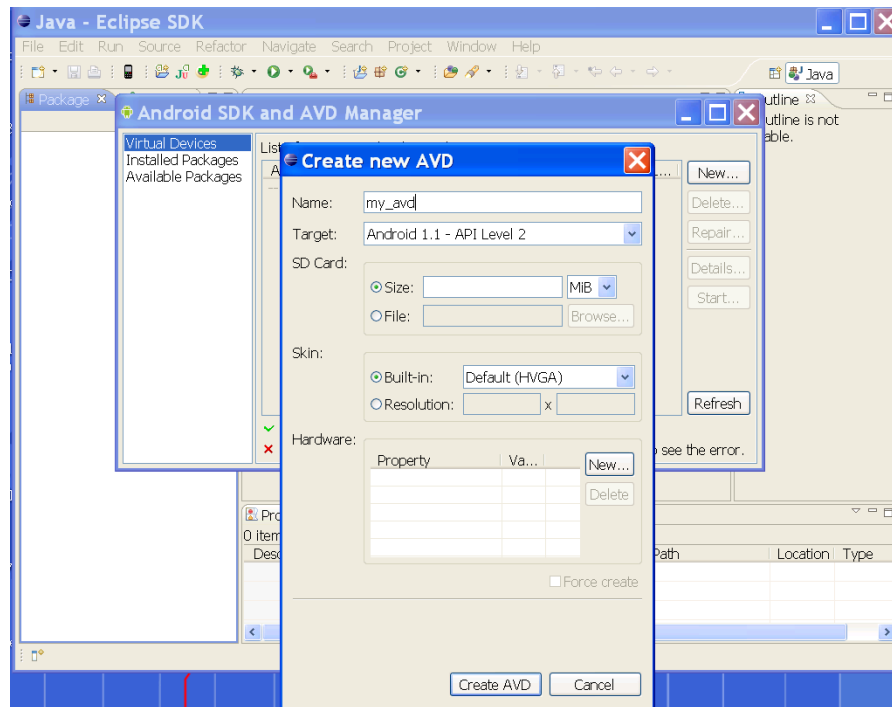
Be sure to check that you are working with a recent version of the JDK. Go to the command prompt and enter `java -version`. If JDK4 is displayed you need to download and install a new JDK and set your path variable to its bin directory.

### Installation

1. Go to <http://developer.android.com/sdk/index.html>
2. Click **System requirement** <http://developer.android.com/sdk/requirements.html>
3. Download the latest version of eclipse, for instance [Eclipse Classic 3.5.1 \(162 MB\)](#) . And decompose it into a folder, such as C:\Program Files\Java\eclipse
4. **Install the ADT Plugin for Eclipse** <http://developer.android.com/sdk/eclipse-adt.html>
5. **Download and install the SDK starter package** <http://developer.android.com/sdk/index.html>
6. In Eclipse, go to window preferences -> Android -> SDK Location. Here provide the path where you downloaded and extracted the file <http://developer.android.com/sdk/index.html>.

### HelloWorld

### Create an AVD



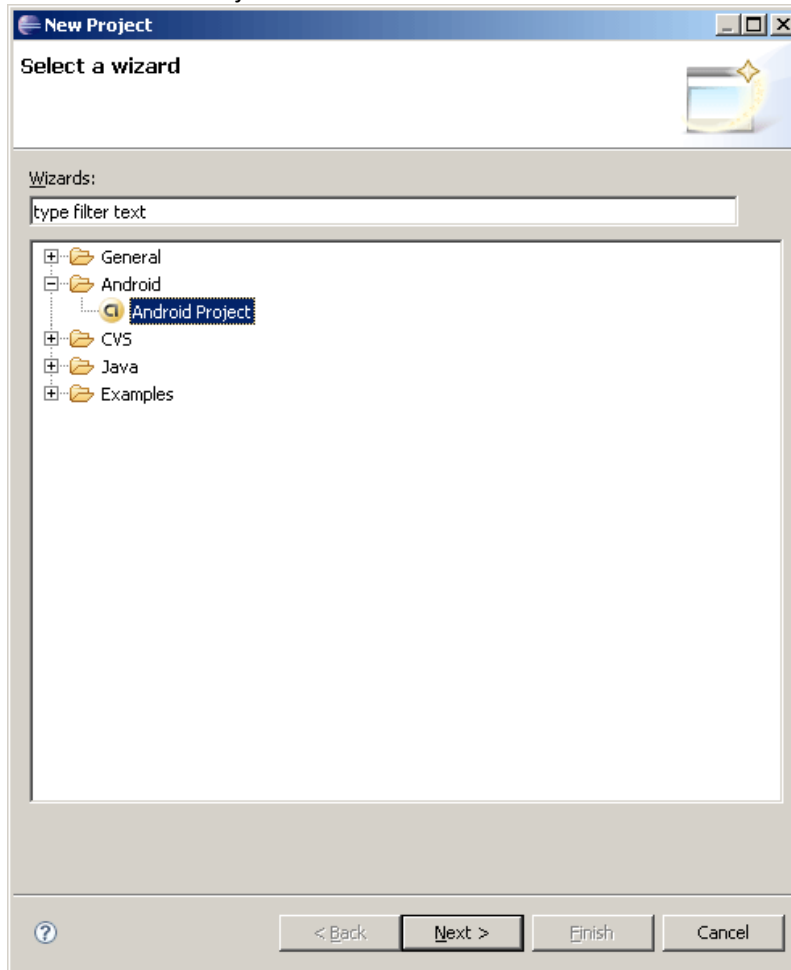
## Create a New Android Project

After you've created an AVD, the next step is to start a new Android project in Eclipse.

1. From Eclipse, select **File > New > Project**.

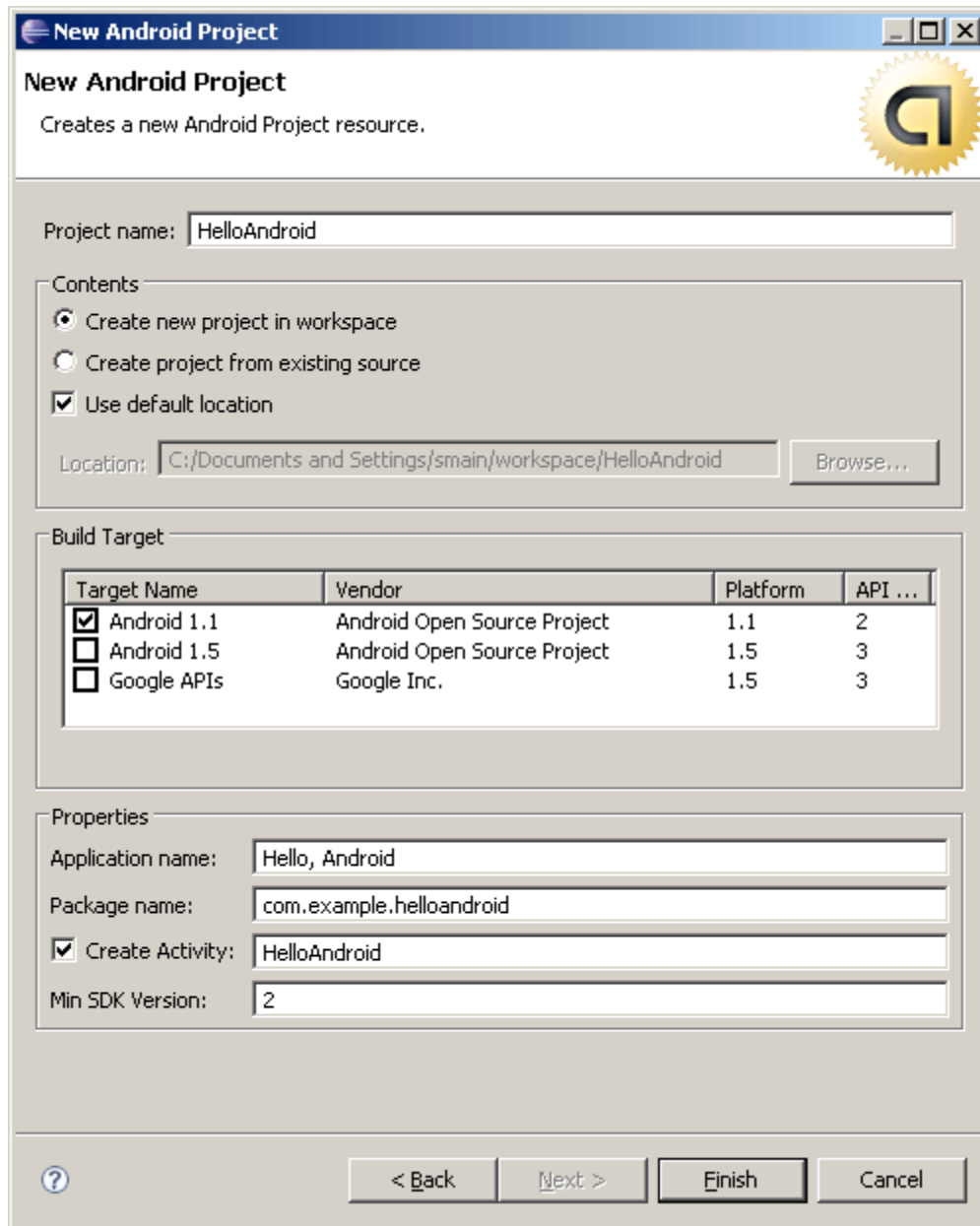
If the ADT Plugin for Eclipse has been successfully installed, the resulting dialog should have a folder labeled "Android" which should contain "Android Project". (After you create one or more Android projects, an entry for "Android XML File" will also be available.)

2. Select "Android Project" and click **Next**.



3. Fill in the project details with the following values:
  - *Project name:* HelloAndroid
  - *Application name:* Hello, Android
  - *Package name:* com.example.helloandroid (or your own private namespace)
  - *Create Activity:* HelloAndroid
  - *Min SDK Version:* 2

Click **Finish**.



Here is a description of each field:

#### *Project Name*

This is the Eclipse Project name — the name of the directory that will contain the project files.

#### *Application Name*

This is the human-readable title for your application — the name that will appear on the Android device.

### *Package Name*

This is the package namespace (following the same rules as for packages in the Java programming language) that you want all your source code to reside under. This also sets the package name under which the stub Activity will be generated.

Your package name must be unique across all packages installed on the Android system; for this reason, it's very important to use a standard domain-style package for your applications. The example above uses the "com.example" namespace, which is a namespace reserved for example documentation — when you develop your own applications, you should use a namespace that's appropriate to your organization or entity.

### *Create Activity*

This is the name for the class stub that will be generated by the plugin. This will be a subclass of Android's [Activity](#) class. An Activity is simply a class that can run and do work. It can create a UI if it chooses, but it doesn't need to. As the checkbox suggests, this is optional, but an Activity is almost always used as the basis for an application.

### *Min SDK Version*

This value specifies the minimum API Level required by your application. If the API Level entered here matches the API Level provided by one of the available targets, then that Build Target will be automatically selected (in this case, entering "2" as the API Level will select the Android 1.1 target). With each new version of the Android system image and Android SDK, there have likely been additions or changes made to the APIs. When this occurs, a new API Level is assigned to the system image to regulate which applications are allowed to be run. If an application requires an API Level that is *higher* than the level supported by the device, then the application will not be installed.

*Other fields:* The checkbox for "Use default location" allows you to change the location on disk where the project's files will be generated and stored. "Build Target" is the platform target that your application will be compiled against (this should be selected automatically, based on your Min SDK Version).

Notice that the "Build Target" you've selected uses the Android 1.1 platform. This means that your application will be compiled against the Android 1.1 platform library. If you recall, the AVD created above runs on the Android 1.5 platform. These don't have to match; Android applications are forward-compatible, so an application built against the 1.1 platform library will run normally on the 1.5 platform. The reverse is not true.

Your Android project is now ready. It should be visible in the Package Explorer on the left. Open the `HelloAndroid.java` file, located inside `HelloAndroid > src >`

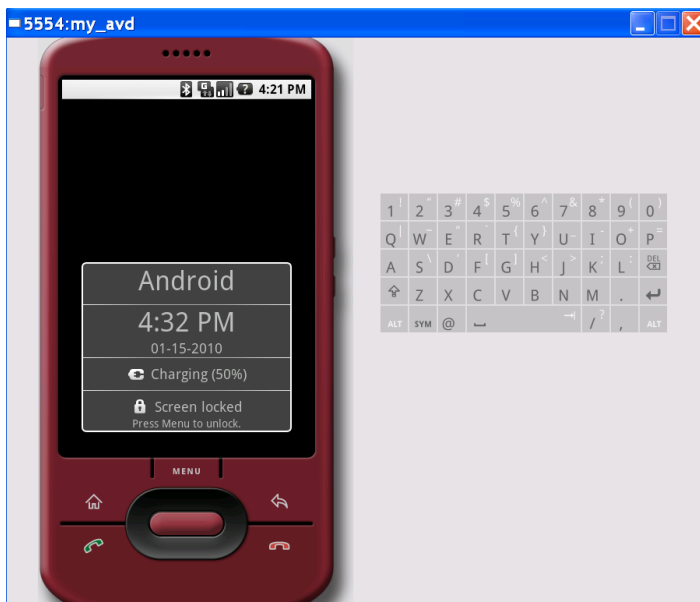
*Modify source by giving:*

```
package com.example.helloandroid;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class HelloAndroid extends Activity
{
/** Called when the activity is first created.
*/ @Override
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
TextView tv = new TextView(this);
tv.setText("Hello, Android");
setContentView(tv);
}
}
```

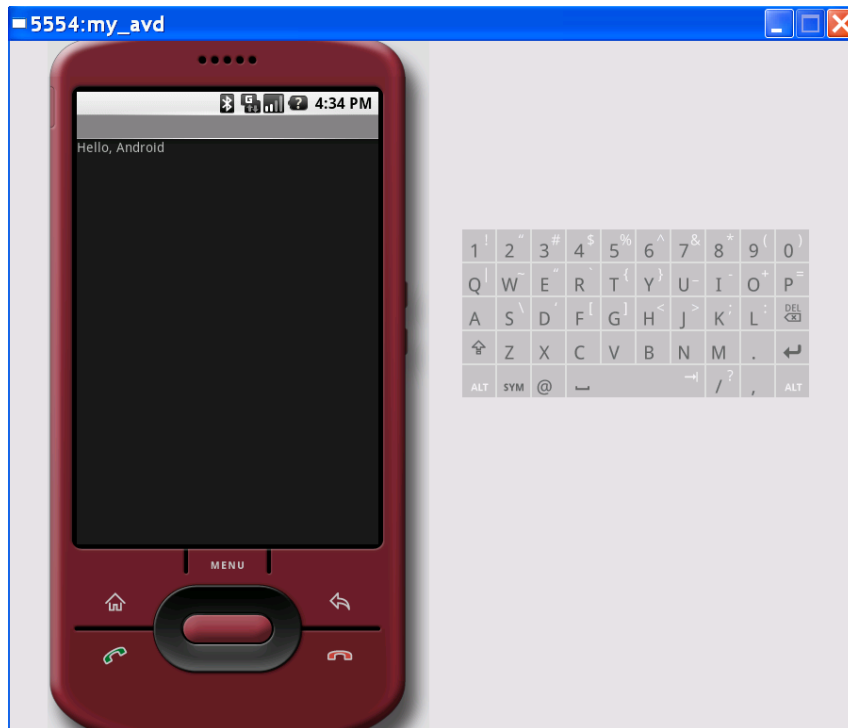
## Run the Application

The Eclipse plugin makes it very easy to run your applications:

1. Select **Run > Run**.
2. Select "Android Application".
3. **Warning: the phone takes a few minutes to load.**



Click MENU



## Upgrade the UI to an XML Layout

Change XML files as illustrated in <http://developer.android.com/resources/tutorials/hello-world.html> . And Run the project. The result looks like this.

5554:my\_avd



1 !	2 "	3 #	4 \$	5 %	6 ^	7 &	8 *	9 (	0 )
Q	W ~	E " ^	R `	T {	Y }	U -	I _	O +	P =
A S \	D ' ^	F [	G ]	H <	J >	K ;	L :	DEL	X
⇧	Z	X	C	V	B	N	M	.	↵
ALT	SYM	@	~		→	/ ?	,		ALT