PART II
IDE

Table of Contents

Configure Android....2
Create new project....3
Views....5
Run the Application....6
Context help....7
Debugging....9
Debug perspective....10

Illustration Index

Illustration 1: Preferences....2
Illustration 2: SDK Location 1....2
Illustration 3: SDK Location 2....3
Illustration 4: Create Project 1....3
Illustration 5: Create Project 2....4
Illustration 6: Select Target Platform....4
Illustration 7: Show Android Views 1....5
Illustration 8: Show Android Views 2....5
Illustration 9: Run the application....6
Illustration 10: Emulator....7
Illustration 11: Context help 1....7
Illustration 12: Context Help 2....8
Illustration 13: Context help 3....8
Illustration 14: Set breakpoint....9
Illustration 15: Debug the project....10
Illustration 16: Debug perspective....10
Illustration 17: Resume....11
Illustration 18: Emulator....11
Illustration 19: Project structure....12

Exercise 1 - IDE

Do the following as it has been shown in the Demo.

Configure Android

Go to the Eclipse folder and start Eclipse. Select `<path-to-where-you-unpacked-Android.zip>\eclipse\eclipse.exe`. Once started, Eclipse asks you for the workspace. Select `<path-to-where-you-unpacked-Android.zip>\workspace`

Next, you should import two projects you will need later on as a reference. Click on `File → Import → General → Existing Projects Into Workspace → Next`.

As **root directory** select the `<path-to-where-you-unpacked-Android.zip>\workspace` directory. Make sure the projects **Activities**, **Notes-Student,** and **OpenX** are selected. Click finish and please do not send any data to Google as you may be asked in the pop up dialogue.

If you get any **".class compatibility"** messages right click on the project and select Properties. Select **"Java Compiler"** and tick **"Enable project specific settings"**. Afterwards click on **Project in the top menu**, then **clean**, and make sure that you **clean all projects**.

Now, click on ***Window → Preferences*** *(see Illustration 1)* then ***Android → SDK Location*** *(see Illustration 2)***.** Enter, or select, the path to the Android location *(see Illustration 3)*.

Notice:
We will provide the path during the seminar.

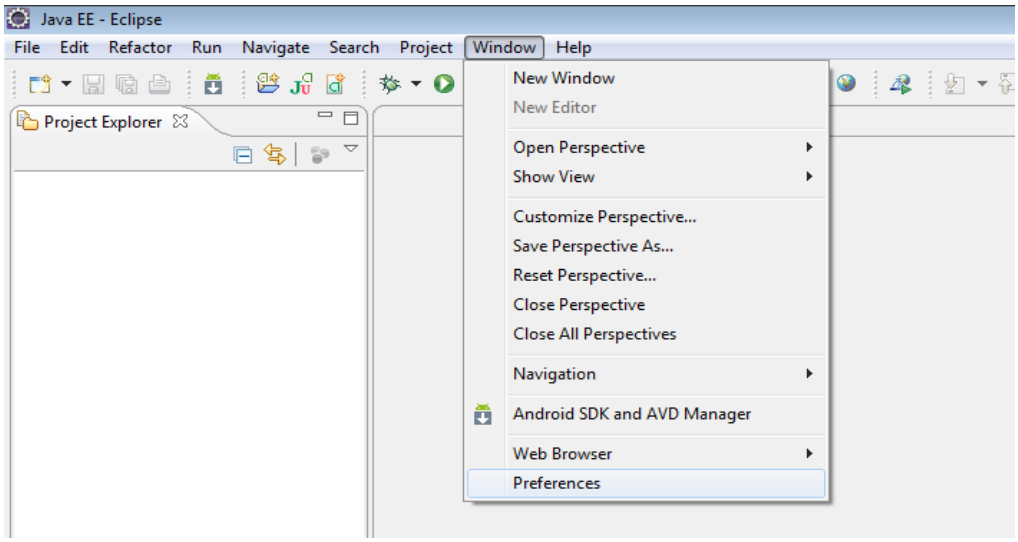

Illustration 1: Preferences

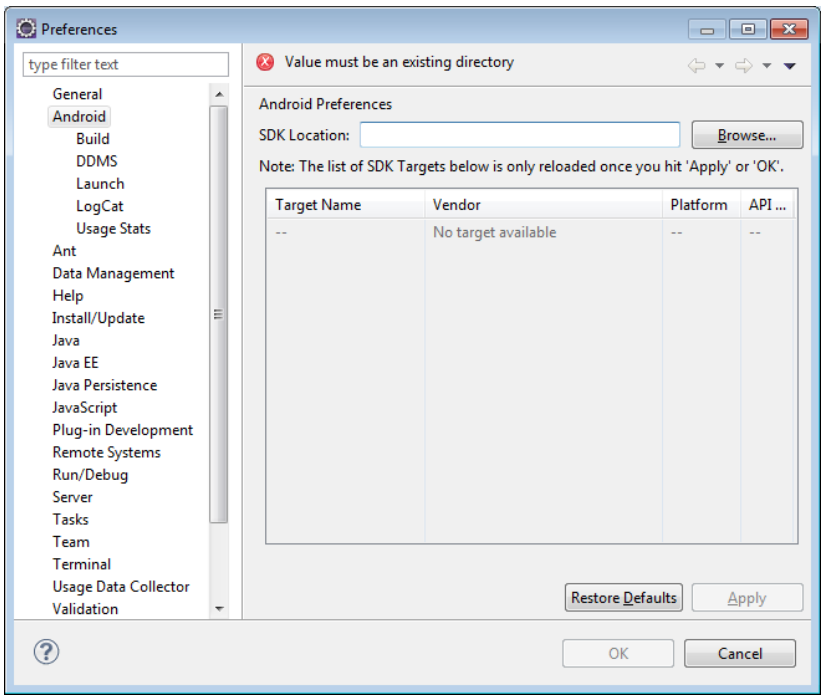

Illustration 2: SDK Location 1

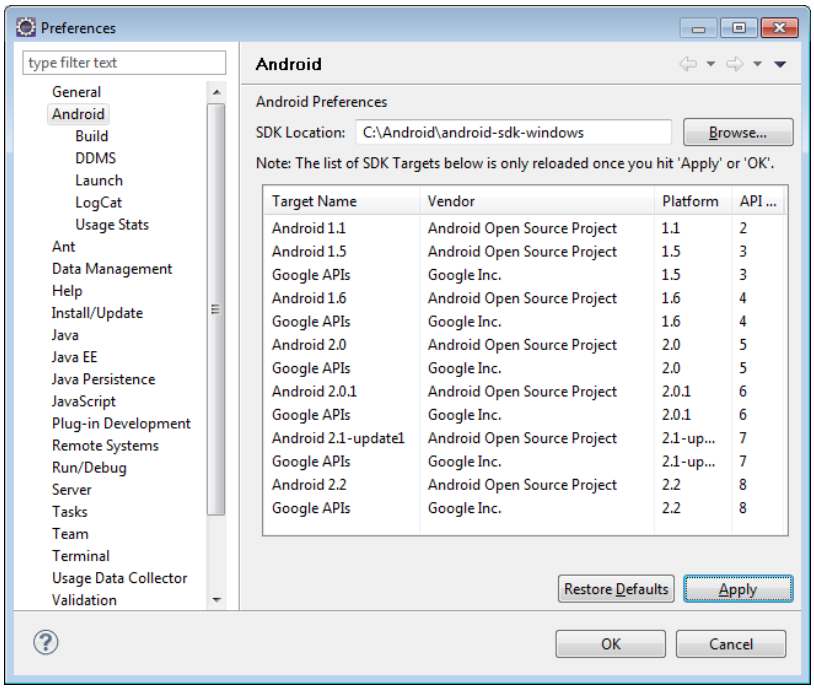

Illustration 3: SDK Location 2

Create new project

Click on ***File → Other*** *(see Illustration 4)* then ***Android → Android Project*** *(see Illustration 5)*. Please provide a *project name*, an *application name* and a *package name* of your choice. The *package name* is required to have at least two parts separated by a ".", e.g., *dbtech.example1*. And select "*Android 2.2*" as *Build Target (see Illustration 6)*.

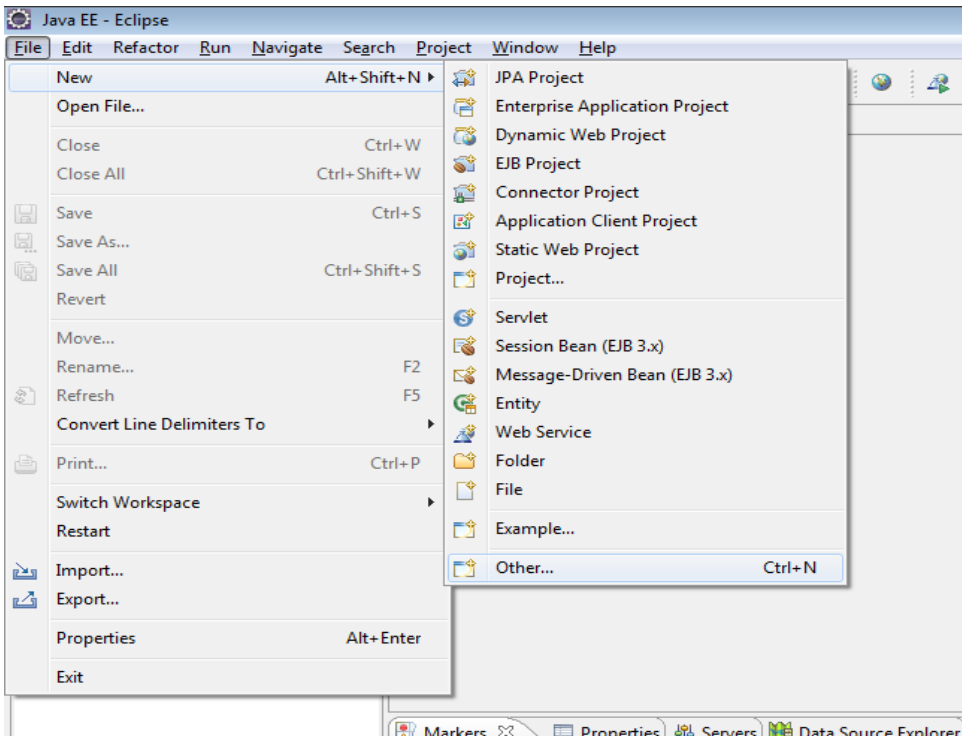

Illustration 4: Create Project 1

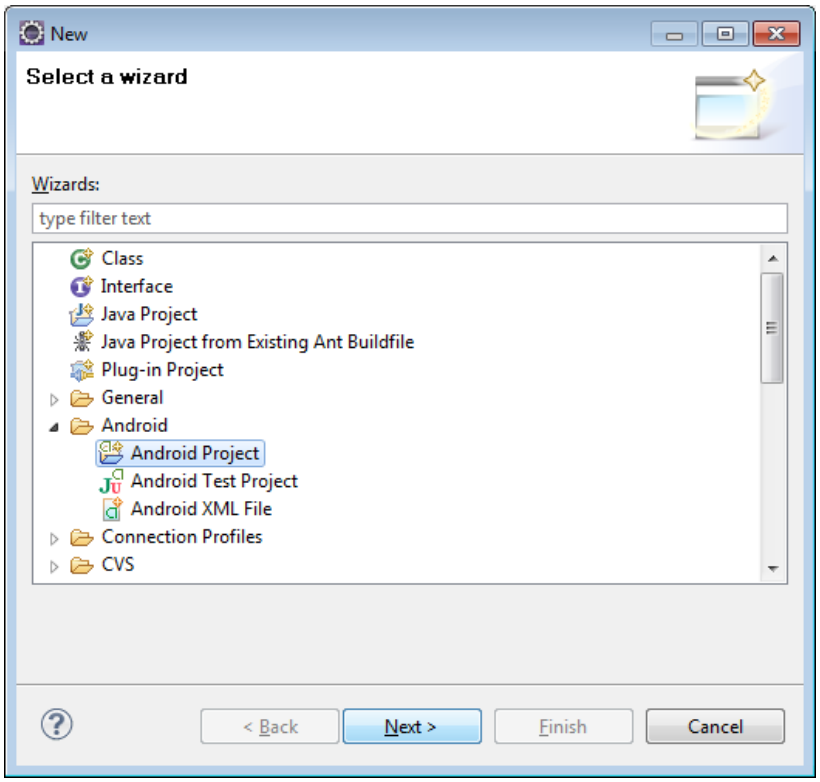

Illustration 5: Create Project 2

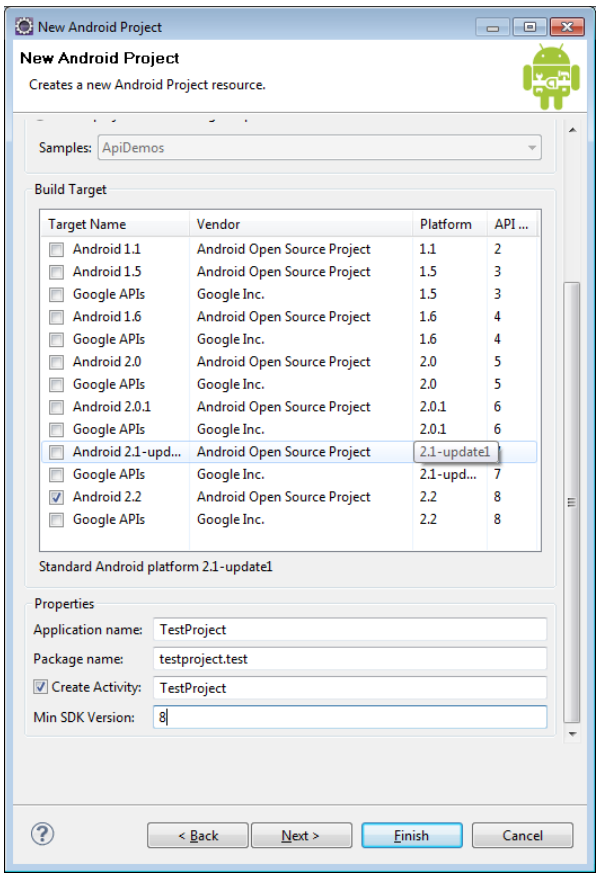

Illustration 6: Select Target Platform

Views

Eclipse applies a ***Perspective*** and ***View*** concept and each plugin, like the Android plugin, ships with an own perspective and corresponding views. To activate the Eclipse Views for Android click on ***Window → Show View → Other → Android***

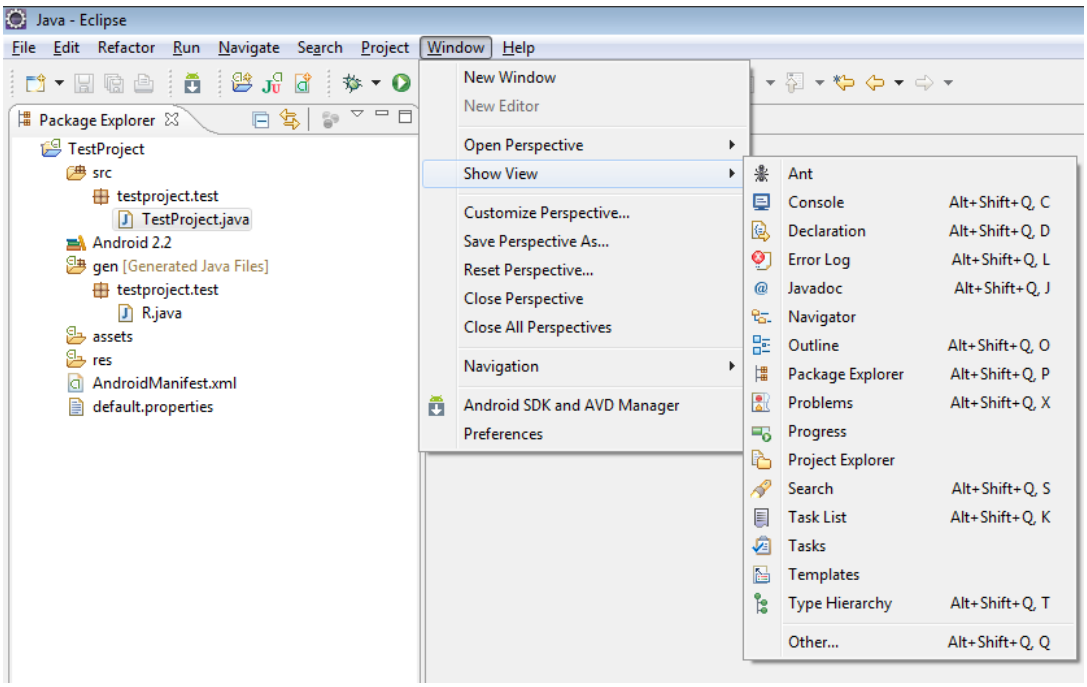

Illustration 7: Show Android Views 1

Select all entries in the Android folder.

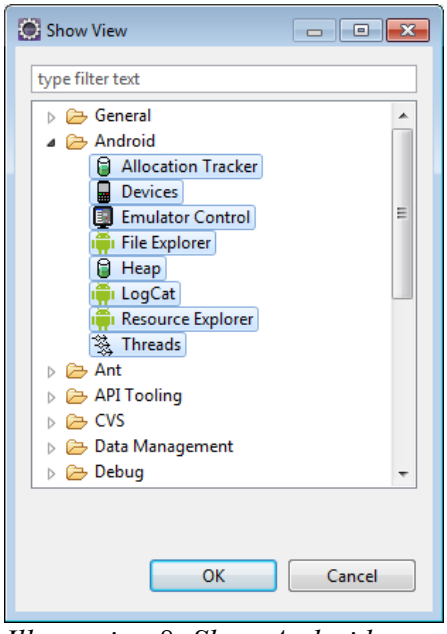

Illustration 8: Show Android Views 2

Run the Application

Right click on the ***Project → Run As → Android Application*** *(see Illustration 10).*

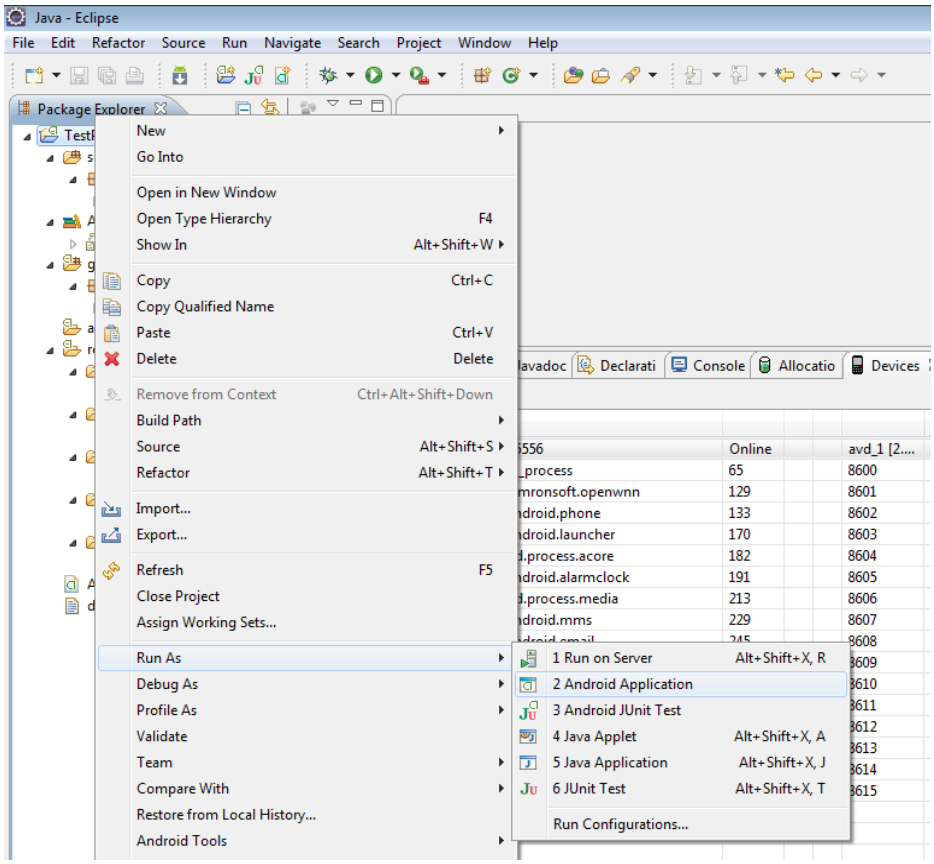

Illustration 9: Run the application

You should see the following output in the log.

```
[2010-10-04 22:33:56 - TestProject] ------------------------------
[2010-10-04 22:33:56 - TestProject] Android Launch!
[2010-10-04 22:33:56 - TestProject] adb is running normally.
[2010-10-04 22:33:56 - TestProject] Performing testproject.test.TestProject
activity launch
[2010-10-04 22:33:56 - TestProject] Automatic Target Mode: using existing
emulator 'emulator-5556' running compatible AVD 'AVD_1'
[2010-10-04 22:33:56 - TestProject] Uploading TestProject.apk onto device
'emulator-5556'
[2010-10-04 22:33:56 - TestProject] Installing TestProject.apk...
[2010-10-04 22:34:08 - TestProject] Success!
[2010-10-04 22:34:08 - TestProject] Starting activity
testproject.test.TestProject on device emulator-5556
[2010-10-04 22:34:12 - TestProject] ActivityManager: Starting: Intent
{ act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER]
cmp=testproject.test/.TestProject }
```

Furthermore, you should see the emulator. Be patient it may take a while till the application has started *(see Illustration 11)*.

Have also a look into the LogCat View. It is one of the tabs at the bottom centre part of Eclipse.

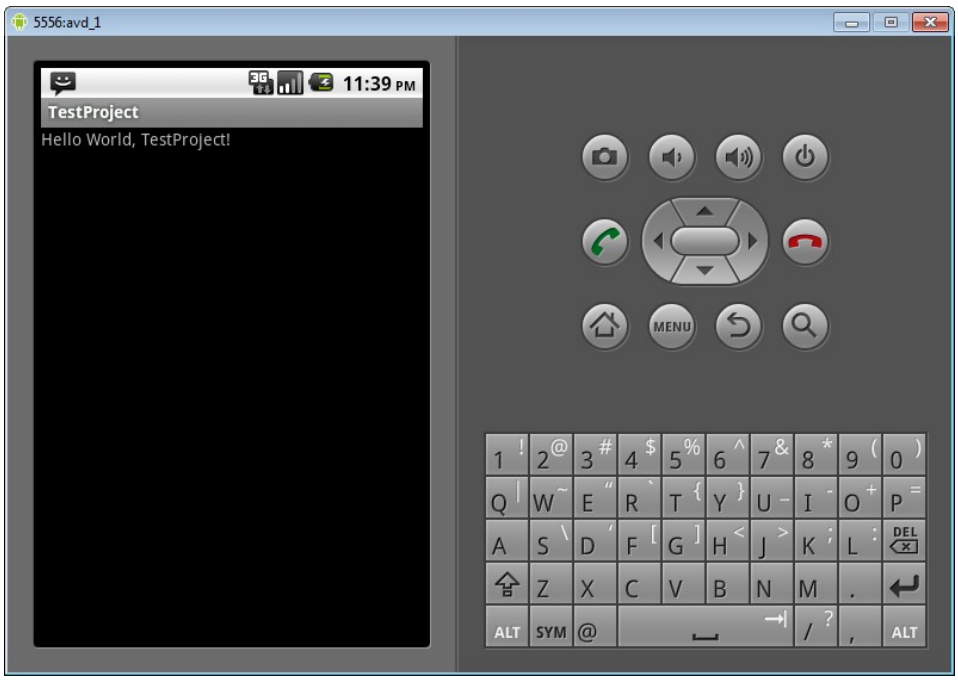

Illustration 10: Emulator

Context help

The IDE provides a context help. In order to get help, place the cursor on the class name and press ***[STRG]+[F1]*** *(see Illustration 12)*. The help menu should pop up on the left side of the IDE. Click on ***Javadoc for "android.app.Activity"*** *(see Illustration 13, 14)*.

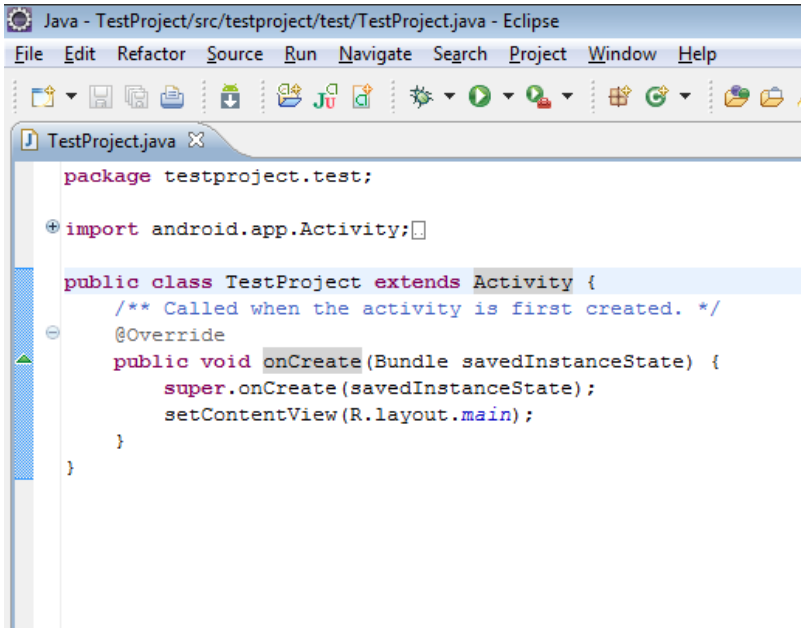

Illustration 11: Context help 1

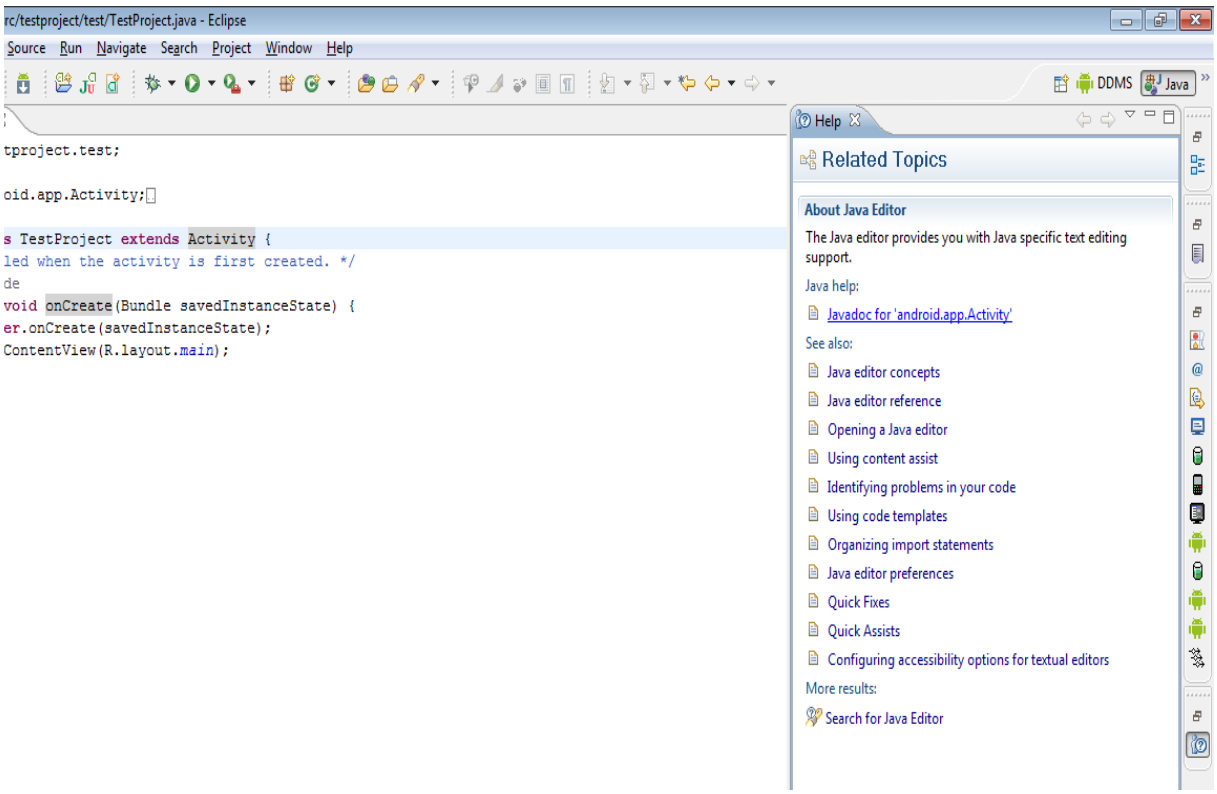

Illustration 12: Context Help 2

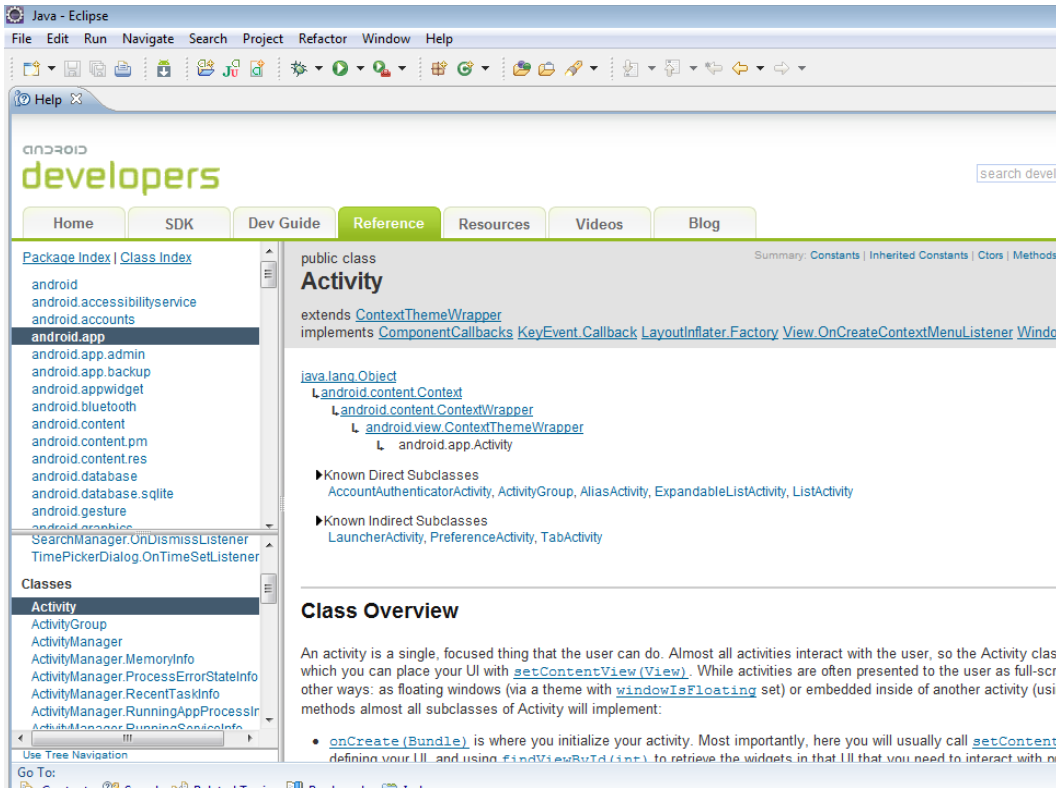

Illustration 13: Context help 3

Debugging

First of all set a breakpoint. Right click on the outer left border of the editor and click ***"Toogle Breakpoint"*** *(see Illustration 15)*. After you've set the breakpoint run the project in debug mode. Right click on ***Project → Debug as → Android Application*** *(see Illustration 16)*.

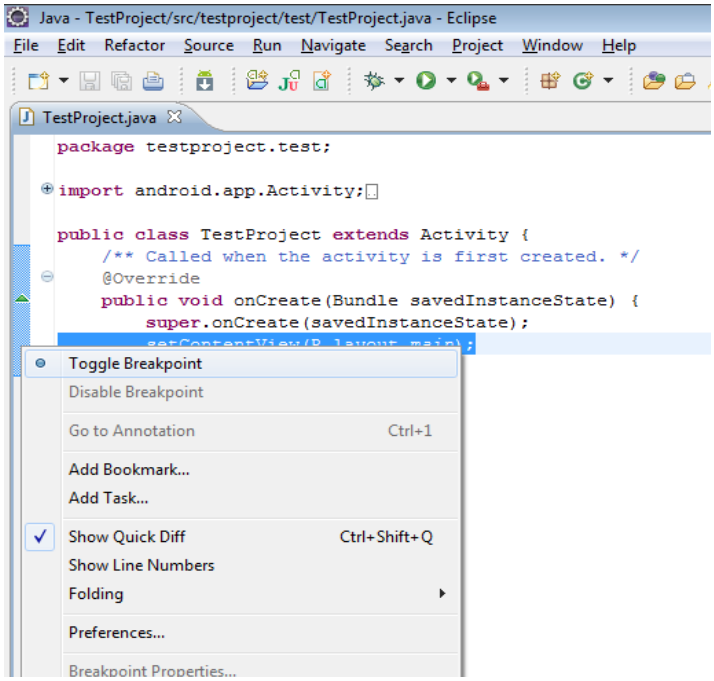

Illustration 14: Set breakpoint

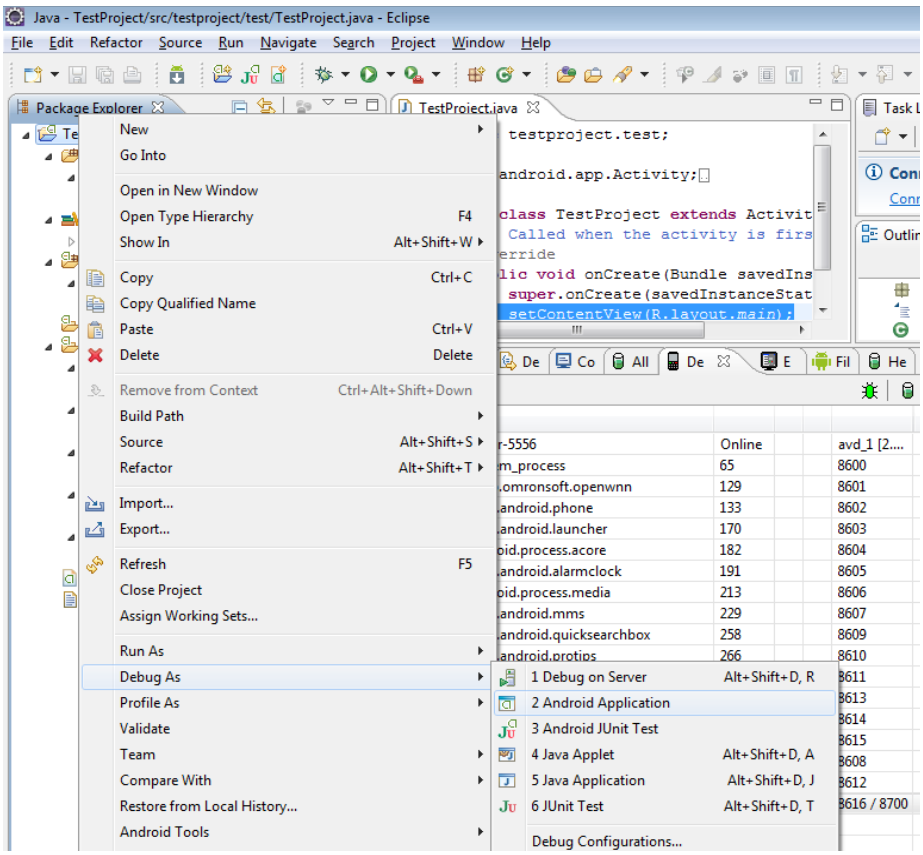

Illustration 15: Debug the project

Debug perspective

Eclipse will now switch to the debug perspective *(see Illustration 17).*

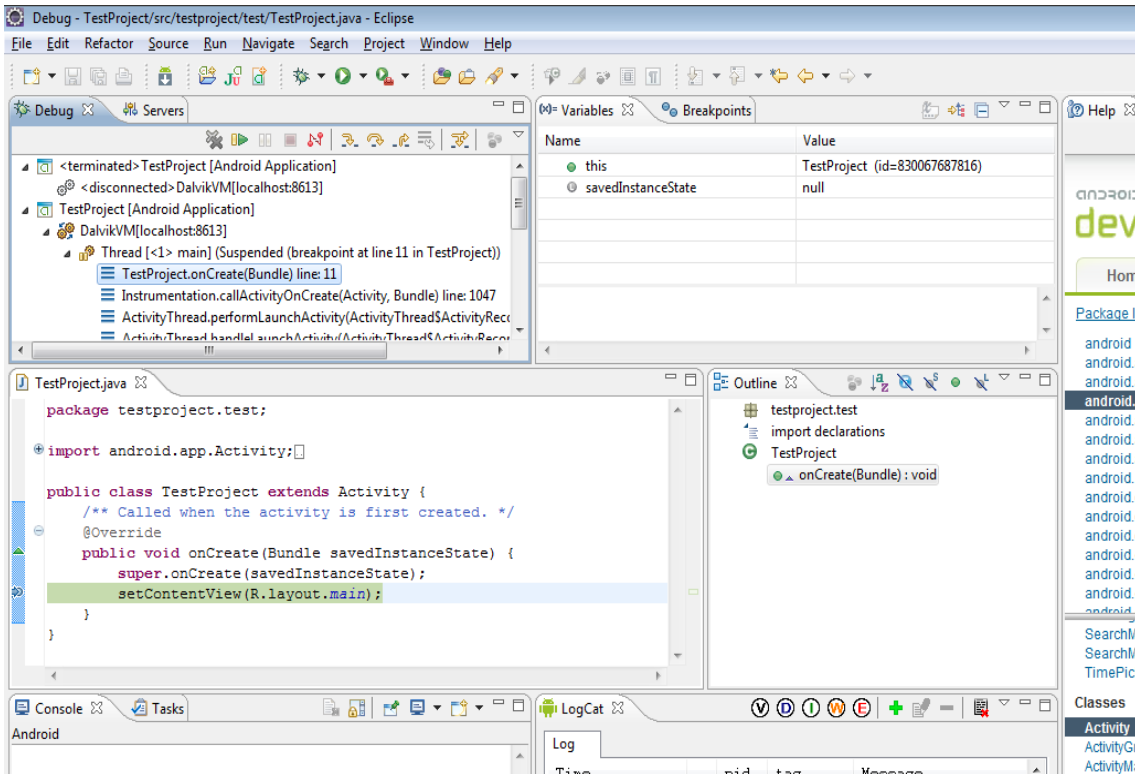

Illustration 16: Debug perspective

As soon as the line of code with the according breakpoint is reached, processing is suspended. If so, you can analyse variables, the stack and a couple of other things. We do not go into detail here and refer to http://www.eclipse.org/resources/resource.php?id=503 instead.

To continue click resume or ***[F8]*** *(see Illustration 18)*. Depending on where the breakpoint has been set you can follow the progress in the emulator.

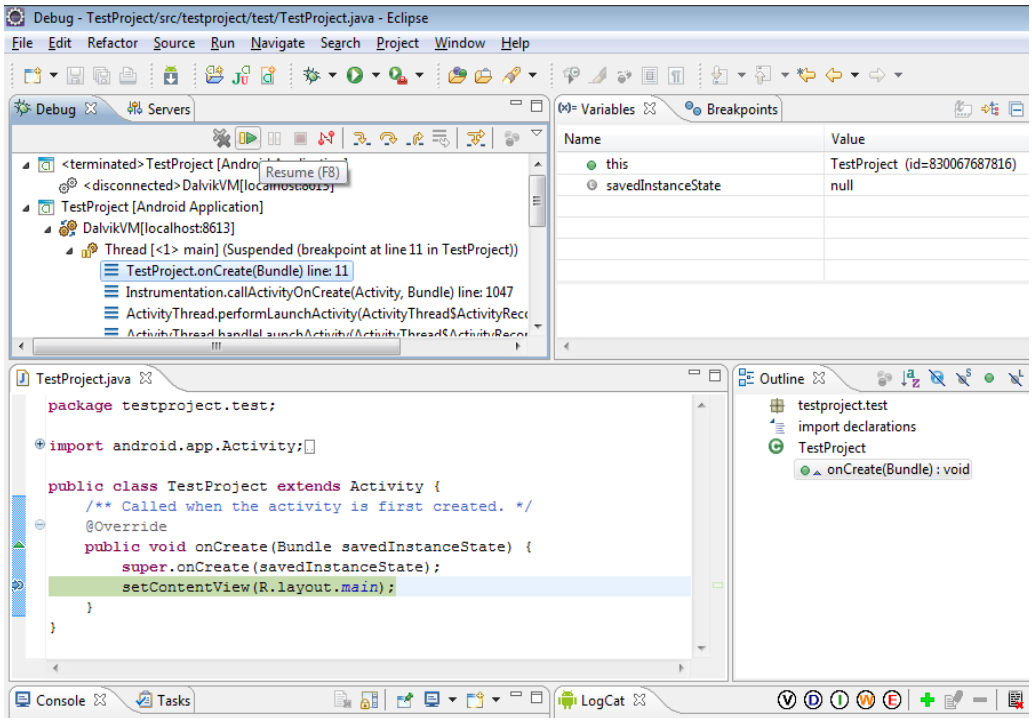

Illustration 17: Resume

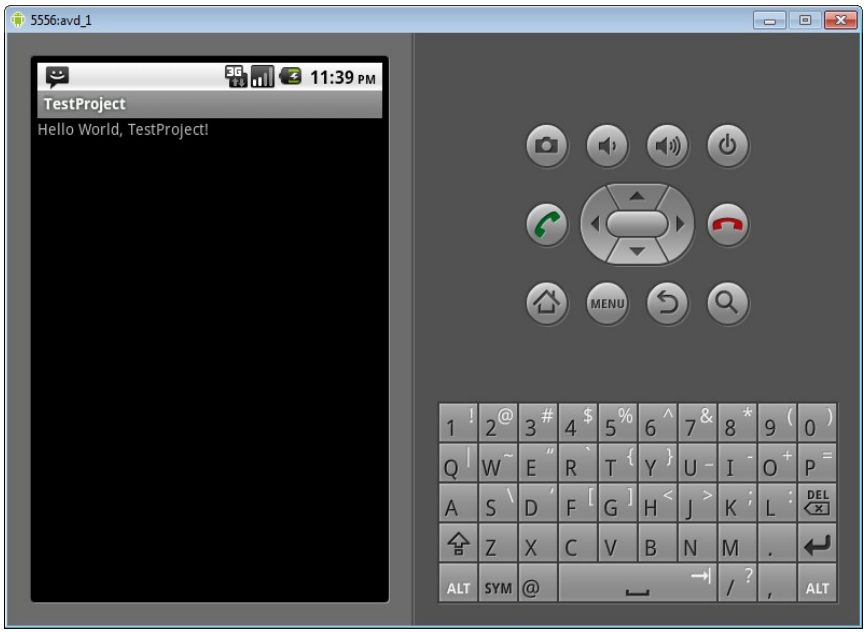

Illustration 18: Emulator

Exercise 2 - Project Structure

An Android project has a specific structure and by expanding the project tree you may notice that there are a couple of folders and files generated already. As shown in Illustration 19, the ***src*** folder, like in other Java projects, contains all the packages and the Java files. Folder ***gen*** contains a generated file named `R.java`. This file is an index that allows to access all resources like text, UI elements, etc. The folder ***assets*** is not of interest so far, however, folder ***res*** is.
To explore this folder and its content is the next exercise (see below). As you may have noticed the project and the application that has been created shows already some text on the screen, namely ***"Hello World, TestProject"***. It might slightly differ depending on your project name.

Explore the ***res*** folder and explain its purpose. Specifically find the character string ***"Hello World, TestProject"***. Discuss in your group how the dynamic binding of resources works.

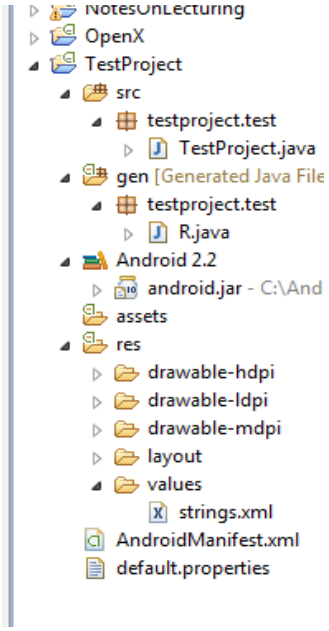

Illustration 19: Project structure