Extracted from:

Hello, Android

Introducing Google's Mobile Development Platform

This PDF file contains pages extracted from Hello, Android, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

The Pragmatic Programmers

Hello, Android

Introducing Google's Mobile Development Platform

Ed Burnette

Edited by Susannah Davidson Pfalzer

Pragmatic Bookshelf

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Printed in the United States of America.

ISBN-10: 1-934356-17-4

ISBN-13: 978-1-934356-17-3

Printed on acid-free paper.

P1.5 printing, July 21, 2009

Version: 2009-7-21

1.2 Creating Your First Program

ADT comes with a built-in example program, or template, that we're going to use to create a simple "Hello, Android" program in just a few seconds. Get your stopwatch ready. Ready? Set? Go!

Select File > New > Project... to open the New Project dialog box. Then select Android > Android Project, and click Next.

Enter the following information:

```
Project name: HelloAndroid
Build Target: Android 1.5
Application name: Hello, Android
Package name: org.example.hello
Create Activity: Hello
```

When you're done, it should look something like Figure 1.2, on the following page.

Click Finish. The Android plug-in will create the project and fill it in with some default files. Eclipse will build it and package it up so it will be ready to execute. If you get an error about missing source folders, select Project > Clean to fix it.

OK, that takes care of writing the program; now all that's left is to try running it. First we'll run it under the Android Emulator.

1.3 Running on the Emulator

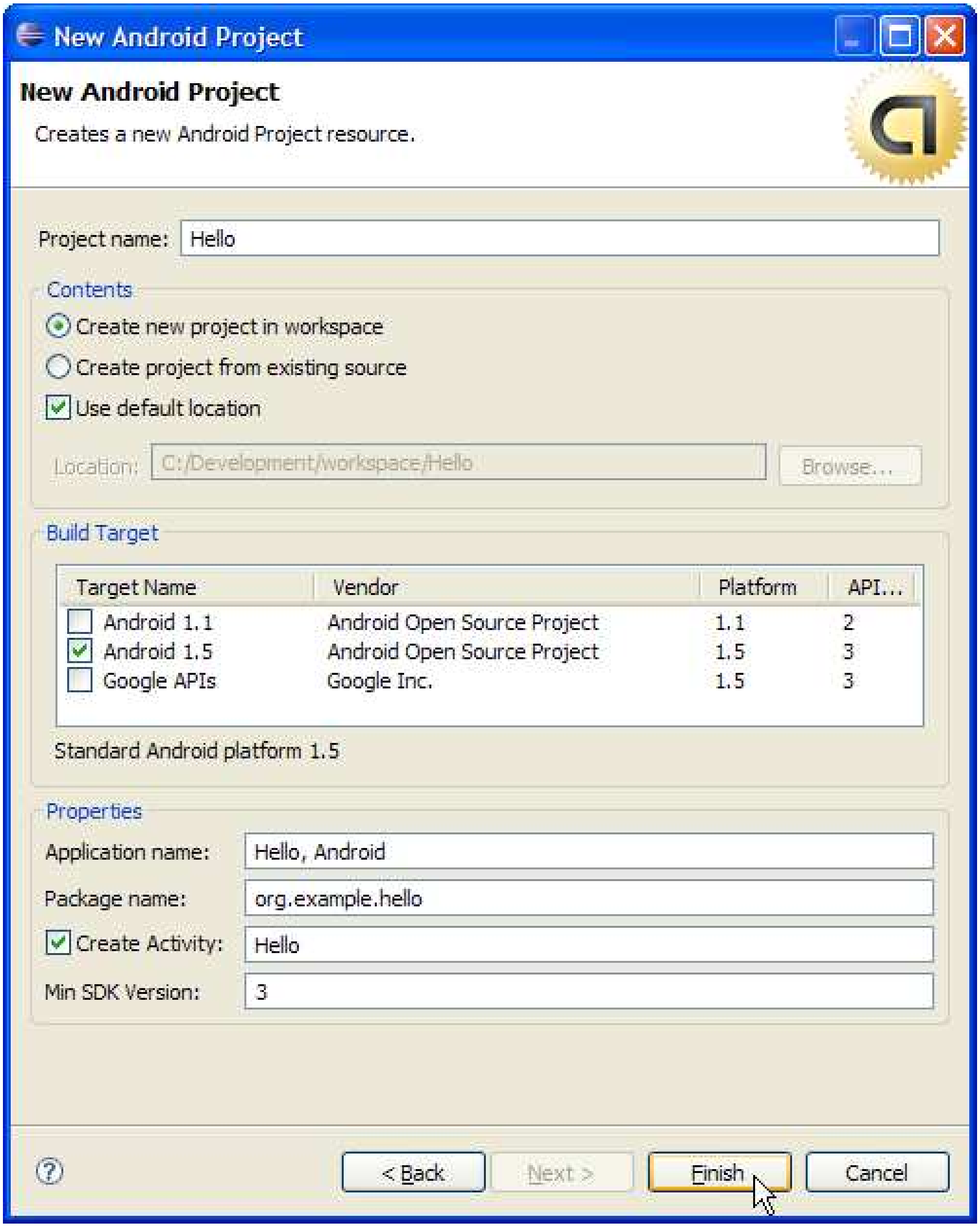
To run your Android program, go to the Package Explorer window, right-click the HelloAndroid project, and select Run As > Android Application. If you're following along in Eclipse you may see an error dialog like the one in Figure 1.3, on page 28. This indicates we haven't told the Emulator what kind of phone to emulate.

Creating an AVD

To do this, you need to create an Android Virtual Device (AVD), using either Eclipse or the android avd command.[6] It's easier to use Eclipse, so select Yes in the AVD Error dialog to open the AVD Manager. You can open the manager again later by selecting Window > Android AVD Manager.

In the AVD Manager dialog, fill out the fields for the new AVD as follows:

[6] http://d.android.com/guide/developing/tools/avd.html



New Android Project

New Android Project

Creates a new Android Project resource.

Project name: Hello

Contents

● Create new project in workspace

○ Create project from existing source

☑ Use default location

Location: C:/Development/workspace/Hello

Build Target

	Target Name	Vendor	Platform	API...
□	Android 1.1	Android Open Source Project	1.1	2
☑	Android 1.5	Android Open Source Project	1.5	3
□	Google APIs	Google Inc.	1.5	3

Standard Android platform 1.5

Properties

Application name: Hello, Android

Package name: org.example.hello

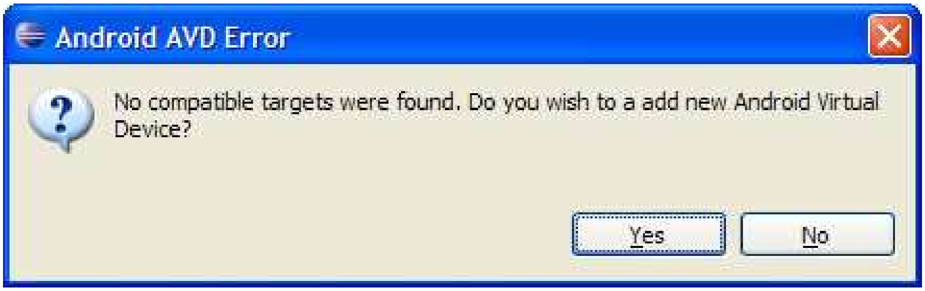
☑ Create Activity: Hello

Min SDK Version: 3

Figure 1.2: New Android project

Keeping Up with the Plug-In

The Android Eclipse plug-in is a work in progress that changes much more often than the Android SDK. The version you download may be different than the one I used when writing this book, and it may contain a few, shall we say, idiosyncrasies. I recommend you check the plug-in site monthly to pick up any new features and fixes.



Android AVD Error

No compatible targets were found. Do you wish to a add new Android Virtual Device?

Figure 1.3: Missing Android Virtual Device (AVD)

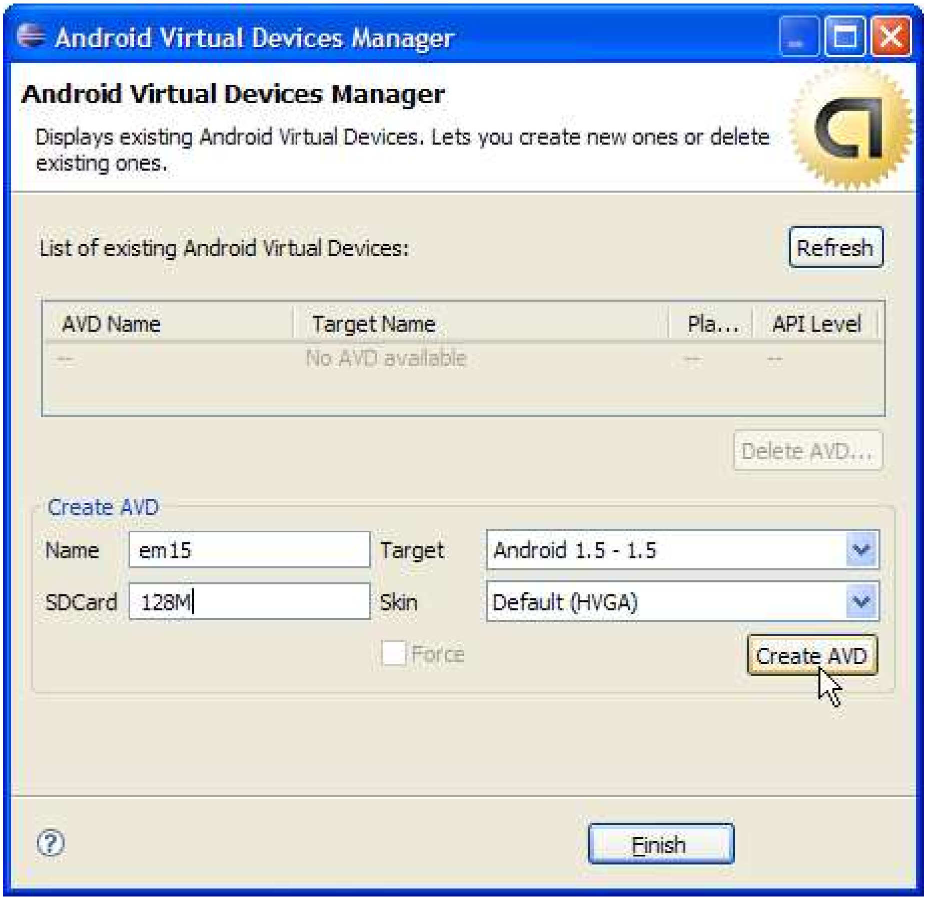
```
Name: em15
Target: Android 1.5 - 1.5
SDCard: 128M
Skin: Default (HVGA)
```

This tells Eclipse to set up a generic device called “em15” which has the Android 1.5 (Cupcake) firmware installed. A 128MB virtual Secure Digital (SD) Card will be allocated, along with a half-VGA (320x480) display.

When you’re done you should see something like Figure 1.4, on the following page. Due to updates in the plug-in since this was written your screen may look slightly different.

Click on Create AVD (not Finish) to create the virtual device. A few seconds later you should see a message that the device has been created. Click OK, and then you can click Finish.

Let’s Try That Again



Android Virtual Devices Manager

Android Virtual Devices Manager

Displays existing Android Virtual Devices. Lets you create new ones or delete existing ones.

List of existing Android Virtual Devices:

AVD Name	Target Name	Pla...	API Level
--	No AVD available	--	--

Create AVD

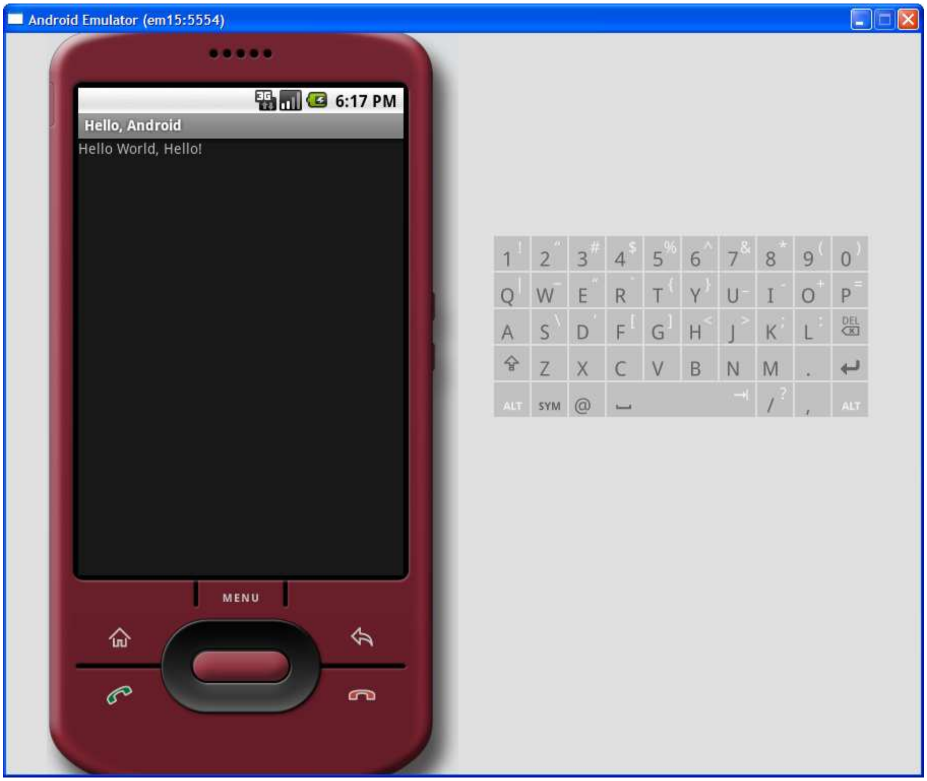
Name	em15	Target	Android 1.5 - 1.5
SDCard	128M	Skin	Default (HVGA)

□ Force

Figure 1.4: Creating an AVD in Eclipse

Once you have a valid AVD, the Android emulator window will start up and boot the Android operating system. The first time you do this, it may take a minute or two, so be patient. If you see an error message saying that the application is not responding, select the option to continue waiting.

After the emulator window starts, Eclipse will send it a copy of your program to execute. The application screen comes up, and your “Hello, Android” program is now running (see Figure 1.5, on the next page). That’s it! Congratulations on your first Android program.



Android Emulator (em15:5554)

6:17 PM

Hello, Android

Hello World, Hello!

Figure 1.5: Running the “Hello, Android” program

Shortening the Turnaround

Starting the emulator is expensive. Think about it this way—when you first turn on your phone, it needs to boot up just like any computer system. Closing the emulator is just like turning off the phone or pulling the batteries out. So, don’t turn it off!

Leave the emulator window running as long as Eclipse is running. The next time you start an Android program, Eclipse will notice the emulator is already there and will just send it the new program to run.

1.4 Running on a Real Phone

Running an Android program on a physical device such as the T-Mobile G1 during development is almost identical to running it on the emulator. All you need to do is connect your phone to the computer with a USB cable and install a special device driver.[7] Close the emulator window if it's already open. As long as the phone is plugged in, applications will be loaded and run there instead.

When you're ready to publish your application for others to use, there are a few more steps you'll need to take. Appendix C, on page 235 will cover that in more detail.

1.5 Fast-Forward >>

Thanks to the Eclipse plug-in, creating a skeletal Android program takes only a few seconds. In Chapter 3, *Designing the User Interface*, on page 46, we'll begin to flesh out that skeleton with a real application—a Sudoku game. This sample will be used in several chapters to demonstrate Android's API.

But before delving into that, you should take a few minutes to read Chapter 2, *Key Concepts*, on the following page. Once you grasp the basic concepts such as activities and life cycles, the rest will be much easier to understand.

Although the use of Eclipse to develop Android programs is optional, I highly recommend it. If you've never used Eclipse before, you may want to invest in a quick reference such as the *Eclipse IDE Pocket Guide* [Bur05].

[7] You can find the device driver and installation instructions at http://d.android.com/guide/developing/device.html.

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Hello Android's Home Page

http://pragprog.com/titles/eband

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

http://pragprog.com/updates

Be notified when updates and new books become available.

Join the Community

http://pragprog.com/community

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

http://pragprog.com/news

Check out the latest pragmatic developments in the news.

Buy the Book

If you liked this PDF, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/eband.

Contact Us

Phone Orders:	1-800-699-PROG (+1 919 847 3884)
Online Orders:	www.pragprog.com/catalog
Customer Service:	orders@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com