Writing Android apps using Scala

Babu Srinivasan (srinivasan.biz)

4 Dec 2010

Contents

1 Introduction

Java is typically used to develop android apps. Java files are compiled to .class files. The .class files are converted to classes.dex (Dalvik Executable) file. The zipped android package file (apk) will contain classes.dex. Dalvik vm in the Android device (or emulator) will run (interpret) classes.dex.

Instead of using Java, we can use Scala. The scala compiler converts scala files to java .class files, which can be converted to dex.

1.1 Why Scala?

Scala is a high-level language (with syntax a bit similar to java) that supports functional programming. I would call it "java-done-right". It is a language with succinct syntax that provides a high level of abstraction — functions are first class, closures are supported. Type inference (though limited) gives you all the advantages of static typing without having to enter types. More importantly, Scala takes advantage of the Java ecosystem by compiling to Java byte code. High performance JIT compilers can be used to run byte code generated by scala compiler. Scala code can utilize Java libraries. I have a blog article on the advantages of Scala.[1]

1.1.1 Reducing Code Size with ProGuard

As we use Scala to write our app, the size of the android package file (apk) is very big. ProGuard software is used to remove unused classes, fields, methods and attributes in addition to code obfuscation. Code obfuscation also reduces code size. The apk file now is in the range of 25K bytes in size for small apps.

1.2 Follow along by grabbing my code from github

You can follow along by grabbing my code from github. You need to install git. If you are running Windows, you can download git from here.

2 *Hello World* Android app using Scala

2.1 Requirements

- Install Scala
- Install android sdk
- Install sbt
- Install android plugin from git://github.com/jberkel/android-plugin.git

[1] http://blog.srinivasan.biz/software/if-you-have-to-learn-just-one-programming-language

2.2 Use wizard to create app

Run the script as follows

```
android-plugin/script/create_project excusms biz.srinivasan.android.excusms
```

This will create all the files required for a simple android app that will display "hello world"

2.3 Run update to download scala and plugins

Run sbt update command which will download scala 2.7.7 (if not already present) and required plugins.

```
cd excusms
sbt update
```

You shouldn't get any errors.

Output (some details omitted)

```
Getting Scala 2.7.7
Getting org.scala-tools.sbt sbt_2.7.7 0.7.4
[info] Recompiling plugin definition
[info] Updating plugins
[success] Plugins updated successfully.
[info]  Extracted source plugin .\lib_managed\scala_2.7.7\sbt-android-plugin-0.5.0.ja
[info] Recompiling plugin
[info] Recompiling project definition
[success] Build completed successfully.
```

2.4 Build the app and create package file (apk)

```
sbt package-debug
```

If everything goes well you will get a message

```
"[success] Build completed successfully."
```

You will see the android package file excusms_2.7.7-0.1.apk in target/scala_2.7.7 directory. File size is 5k. You have created your first android app using scala. Test app will also be created (tests/target/scala_2.7.7/tests_2.7.7-0.1.apk)

2.5 Load and run app on android device or emulator

You can now load this app in the emulator (or a real android device).

Load the app in the emulator.

```
adb -s emulator-5554 install -r target/scala_2.7.7/excusms_2.7.7-0.1.apk
```

Run it. You should see “hello,world” on the screen.

2.6 Contents of Android Package file (apk)

You can use the jar command to list the contents of apk

```
jar tf excusms_2.7.7-0.1.apk

res/drawable/app_icon.png
AndroidManifest.xml
resources.arsc
classes.dex
META-INF/MANIFEST.MF
META-INF/CERT.SF
META-INF/CERT.RSA
```

The android app (excusms_2.7.7-0.1.apk) will be signed with Android Debug certificate. Later, when you are ready to deploy the app or send to Android market, you can sign the app with your own certificate. sbt provides a “sign-release” target which will sign and zipalign the package file. We will do this after writing the application.

2.7 Activity.scala file

As mentioned earlier the android plugin (wizard) automatically creates all the files required to create “hello world app”. The starting point for Android application centers around an Activity, which is defined as *a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with setContentView(View).*

The automatically created Activity.scala file is shown below:

```
package biz.srinivasan.android.excusms

import _root_.android.app.Activity
import _root_.android.os.Bundle
import _root_.android.widget.TextView

class MainActivity extends Activity {
  override def onCreate(savedInstanceState: Bundle) {
    super.onCreate(savedInstanceState)
    setContentView(new TextView(this) {
      setText("hello, world")
    })
  }
}
```