Building the To Do application

for Android

by Trevin Beattie

Table of Contents

1 Building from the Source

When making any changes to the code, you also need to change the release date in the "About..." dialog (`res/values/strings.xml` / `InfoPopupText`) as well as the application's internal version number (`AndroidManafest.xml` / `android:versionCode` and `android:versionName`).

Building the application `.apk` was originally done up until 2014 using:

- A Java 1.6 compiler
- Eclipse integrated development environment (IDE)
- The Android software development kit (SDK) release 12.

To build the old code on a newer system requires a few different tools, but not *too* modern. The following development environment was set up and tested in 2025 on Fedora Linux 40 with only minor bug fixes to the code:

- Java (1.)8 *(Do* **not** *use a newer version, as Gradle 3.5 is not compatible with Java 11 or higher.)*
- Android Studio 2.3.3, downloaded from the archive (`https://developer.android.com/studio/archive`).
- Gradle 3.5, installed by setting up the Gradle Wrapper for the project and then modifying `gradle/wrapper/gradle-wrapper.properties` to set `distributionUrl` to `https\://services.gradle.org/distributions/gradle-3.5-all.zip`.
- The Android Gradle Plugin version 2.3.3, which had to be downloaded along with most of its dependencies from a 3rd-party mirror `https://repository.axelor.com/nexus/service/rest/repository/browse/maven-public/` as it does not exist in Maven Central nor in Google's Maven repository.

See `http://developer.android.com/sdk/index.html` for information on how to set up a project using the Android SDK.

1.1 Generating Icons

Some of the application icons are generated from 3-D image description files using Persistence of Vision (`http://www.povray.org/`). The image files are included with the source code (in `app/src/main/res/drawable*/`, but in case you want to tweak or change any image you can use `povray` to generate new ones. The `.pov` sources are under `app/src/main/graphics/`.

Because some of the icons contain small details that may get lost if rendered at a low resolution, it is recommended that you render the initial image at a size which is the least common multiple of all icon sizes — 288×288 — and then use a raster graphics program such as the GIMP (`http://www.gimp.org/`), `convert` (`http://www.imagemagick.org/`), or `pamscale` (`http://netpbm.sourceforge.net/`). For each image file

`foo.png`, you need to create five icons: a 16×16 icon in `res/drawable/foo_16.png`, a 24×24 icon in `res/drawable-mdpi/foo.png`, a 32×32 icon in `res/drawable/foo_32.png`, a 36×36 icon in `res/drawable/foo_36.png`, and a 48×48 icon in `res/drawable-hdpi/foo.png`.

The command for generating an image from one of the `.pov` descriptions is:

```
povray +FN +AM3 +A0.3 +UA +W288 +H288 foo.pov
```

The main application icon was drawn by hand in the GIMP, and can be found in `IconMaster.xcf`. This is scaled down to 48×48 for `res/drawable-mdpi/icon.png`, and to 72×72 for `res/drawable-hdpi/icon.png`.

1.2 Generating the Application Package (APK)

In Android Studio, click on the Build menu then "Build APK". If there were no errors, this should produce `app/app-release.apk`. You should rename this file to a more descriptive name like `todo-1.2.0.apk`.

1.3 Generating Documentation

Lastly, if you need to generate a new edition of this manual, you will need `texinfo` (`http://www.gnu.org/software/texinfo/`) and `texi2pdf` (`https://www.gnu.org/software/texinfo/manual/texinfo/html_node/Format-with-texi2dvi-or-texi2pdf.html`).

To generate the manual in PDF, simply run:

```
texi2pdf ToDo.texinfo
```

2 To Do

- Document how the code works
- Document the data structures
- Document the encryption algorithms