

Android Fat Aar

Melody

Outline

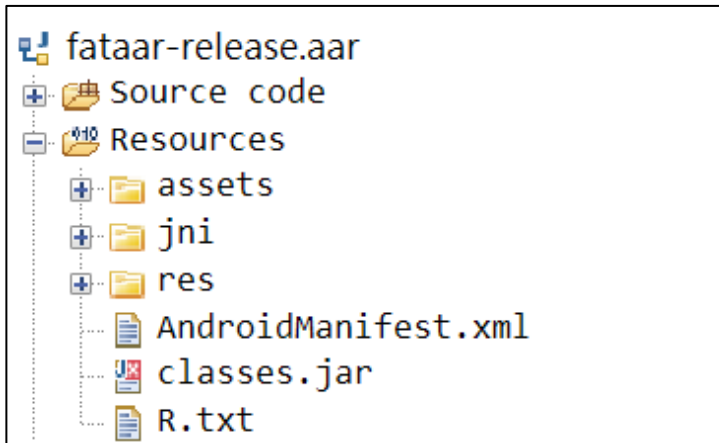
- Android aar
- Android aar content
- Android fat aar
- Gradle
- Gradle tasks
- Custom tasks
- Fat aar content
- Reference

Android aar

- one of the file formats of android sdk (other: jar)

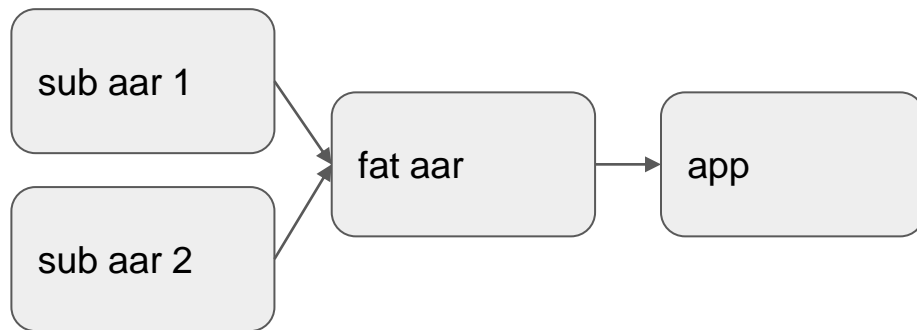
Android aar content

- assets
- jni
- res
- AndroidManifest.xml
- classes.jar
- R.txt
- libs/*.jar



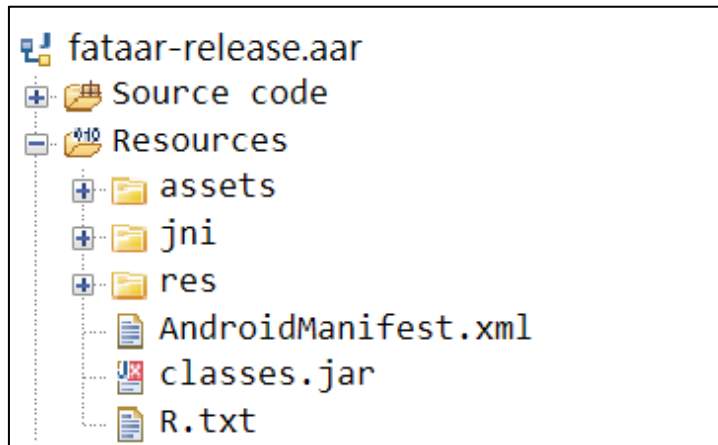
Android fat aar

- dependencies of aar contains other local aars
- not supported by current android tool
- need to merge sub aars by ourselves by using gradle



Gradle

- Android gradle plugin version to com.android.tools.build:gradle:4.1.3
- Gradle version: 6.5
- add custom tasks of gradle to generate fat aar
 - task to merge **res**
 - task to merge **classes**
 - task to merge **jni**
 - task to merge **R.txt**
 - task to merge **assets**
 - (we don't merge AndroidManifest.xml)
 - (we don't merge libs/*.jar)



Gradle tasks

- Relative tasks are shown on the right side
- notes:
 - task name may different when version of gradle tool changed
 - task name would be different after productFlavors added



Custom tasks (1)

- unzip sub aar

- unzip sub aar, **extractAar**

packageReleaseResources

generateReleaseRFile

copyReleaseJniLibsProjectAndLocalJars

generateReleaseAssets

mergeReleaseResources

compileReleaseJavaWithJavac

syncReleaseLibJars

bundleReleaseAar

Custom tasks (2)

- merge resources

- add resource dir, `embedLibraryResources`
- to shrink size of fat aar, remove resources if any dpi of resource exist

`removeDuplicateResourcesIfAnyDpiExist`



Custom tasks (3)

- merge classes

- copy classes.jar, `embedClassesAndJars`

`extractAar`

`packageReleaseResources`

`generateReleaseRFile`

`copyReleaseJniLibsProjectAndLocalJars`

`generateReleaseAssets`

`mergeReleaseResources`

`compileReleaseJavaWithJavac`

`syncReleaseLibJars`

`bundleReleaseAar`

Custom tasks (4)

- merge jni

- copy jni, `embedJniLibs`

`extractAar`

`packageReleaseResources`

`generateReleaseRFile`

`copyReleaseJniLibsProjectAndLocalJars`

`generateReleaseAssets`

`mergeReleaseResources`

`compileReleaseJavaWithJavac`

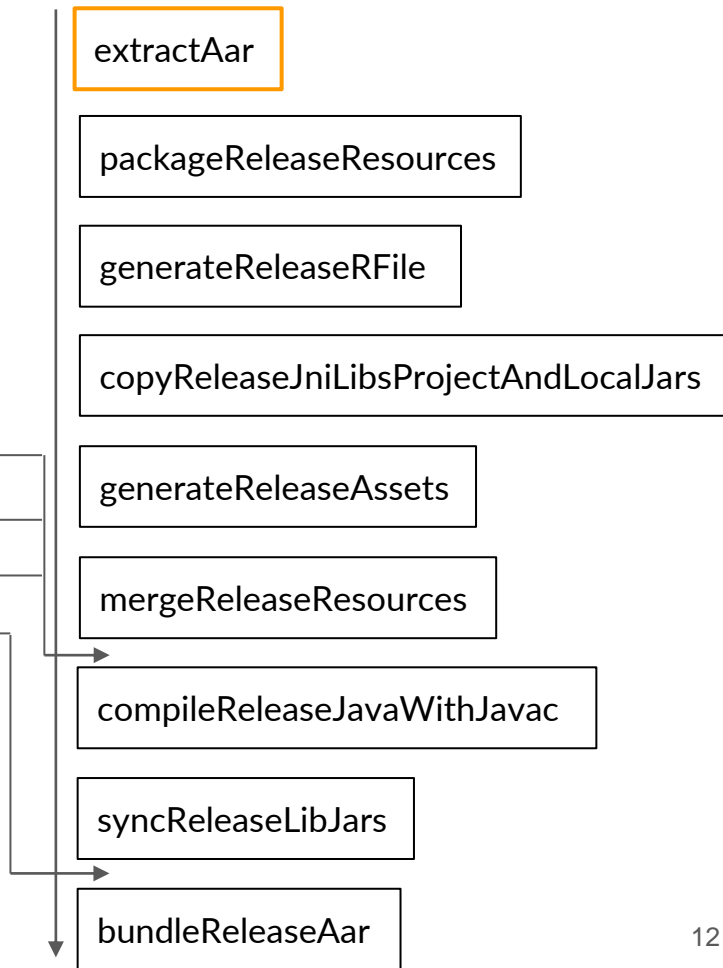
`syncReleaseLibJars`

`bundleReleaseAar`

Custom tasks (5)

- merge R.txt

- generate R.java from R.txt, **generateRJava**
- compile R.java to R.classes, **compileRJava**
- pack R.classes to embedR.jar, **embedRClass**
- copy embedR.jar after obfuscate, **copyEmbedRJar**



Custom tasks (5)

- merge R.txt

- sample of R.java for sub aar
- let resources be map to package of fat aar

```
package com.example.subaar1;

public final class R {
    public static final class string {
        public static int sub_aar_1_text = com.example.fataar.R.string.sub_aar_1_text;
    }
}
```

Custom tasks (6)

- merge assets

- add assets dir, `embedAssets`

`extractAar`

`packageReleaseResources`

`generateReleaseRFile`

`copyReleaseJniLibsProjectAndLocalJars`

`generateReleaseAssets`

`mergeReleaseResources`

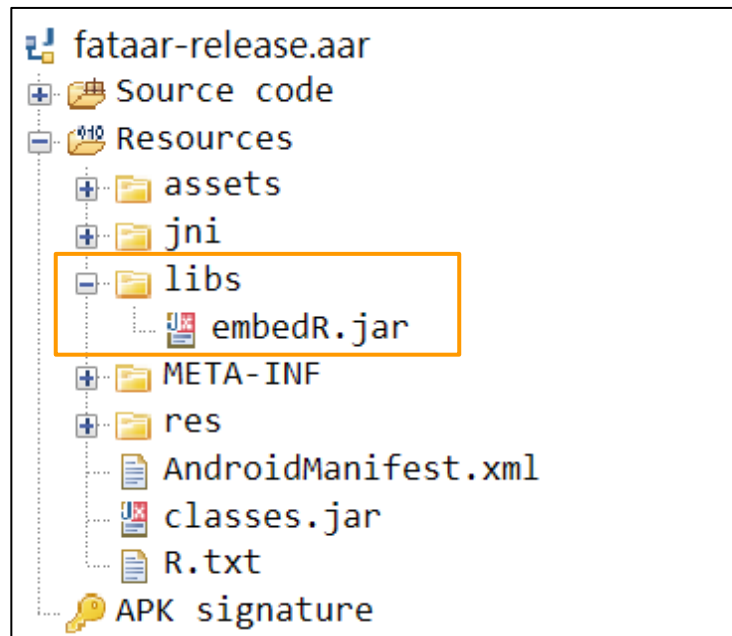
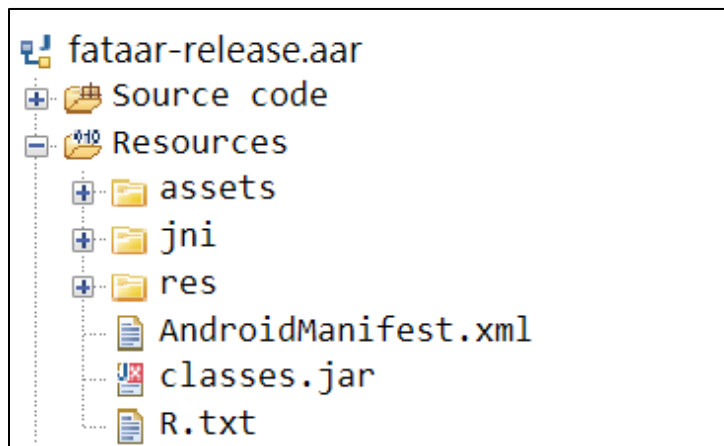
`compileReleaseJavaWithJavac`

`syncReleaseLibJars`

`bundleReleaseAar`

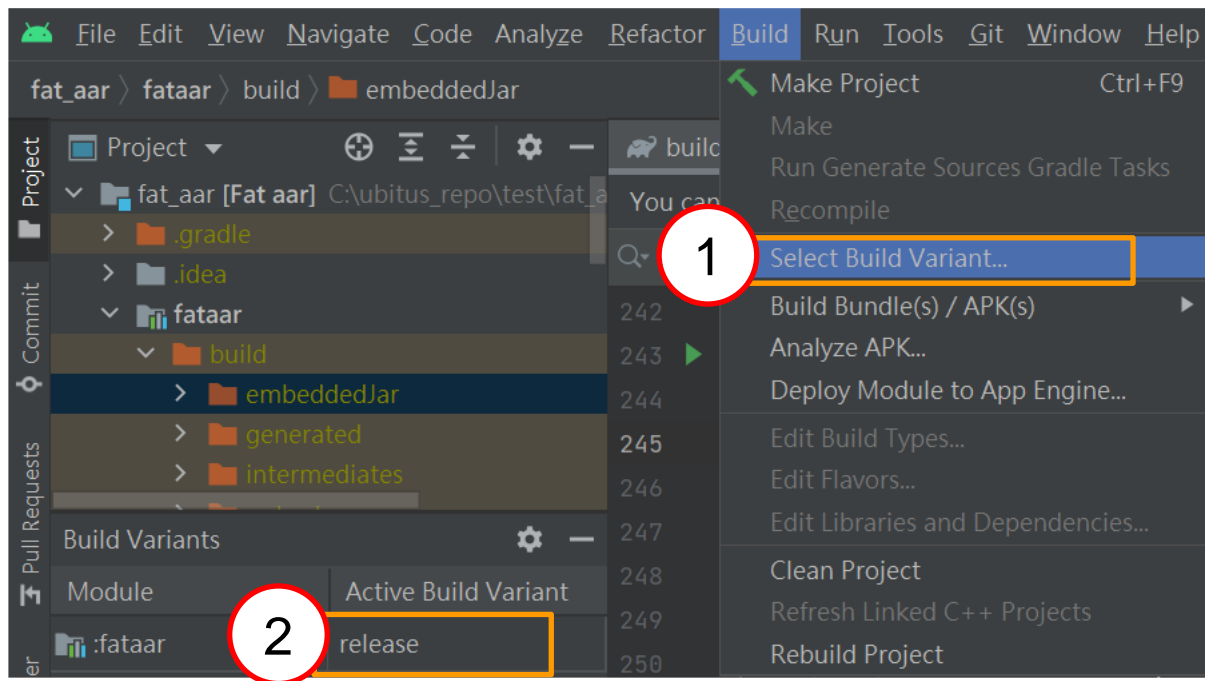
Fat aar content

- embedR.jar is been included
- other resources are all merged into fat aar



Build fat aar

- Build variant must use **release** to do obfuscation



Reference

<https://www.jianshu.com/p/57024509b1d6>

<https://www.jianshu.com/p/8f7e32015836>

[AS 3.1 多library合并打包成aar的正确方式 \(fat-aar \)](#)

[android-fat-aar/fat-aar.gradle at master · adwiv/android-fat-aar · GitHub](#)