Android应用的编译其实就是打包为.apk文件，这个被打包的zpk文件实质其实是一个压缩包（譬如你可以重命名后缀解压），这个压缩包至少包含编译为.class文件转换的.dex文件、一个二进制的AndroidManifest.xml文件、编译的资源文件resources.arsc、未编译的资源文件等，然后在运行前再对这个压缩包进行签名操作即可。

构建系统将依据配置的product flavors、build types和dependencies合并所有的资源，如果此时不同文件夹含有相同名称或设置的资源则会按照如下规则进行覆盖：dependencies覆盖build types，build types覆盖product flavors，product flavors覆盖main资源文件夹。下面是对上图典型构建过程的详细描述：

1. Android资源打包工具（AAPT）将应用的资源文件（譬如AndroidManifest.xml文件和Activity相关的XML文件等）进行编译，生成的一个R.java文件就是关联我们Java代码与资源文件的基础。
2. aidl工具将应用中所有.aidl的文件转换为Java接口。
3. Java编译器将应用中所有Java代码（包括R.java和aidl接口）编译输出为class类文件。
4. dex工具将应用编译输出的类文件（包括第三方jar包）转换为Dalvik字节码，.dex文件就可以最终被打包如.apk文件。
5. apkbuilder工具将所有非编译资源（譬如images）、编译资源和DEX文件打包成一个APK文件。
6. 一旦.apk文件生成以后就必须进行签名（debug或者release签名）才能在设备上运行。
7. 通过zipalign工具对齐APK，减少APP在设备上运行时的内存占用。

整个构建过程生成的东东默认都在app/build目录下（包括中间产物和最终产物）。