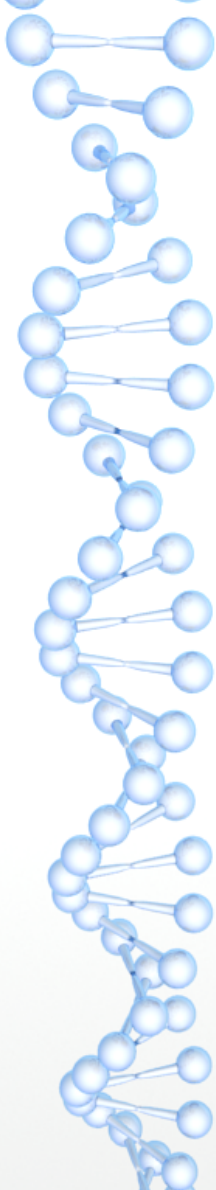
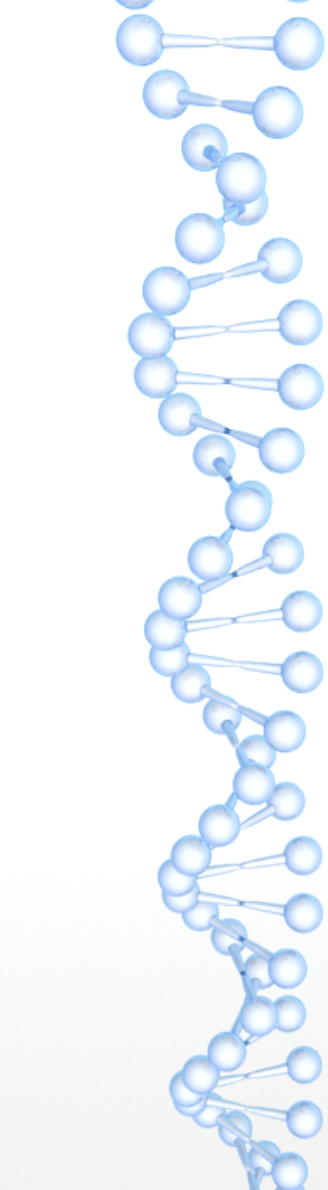
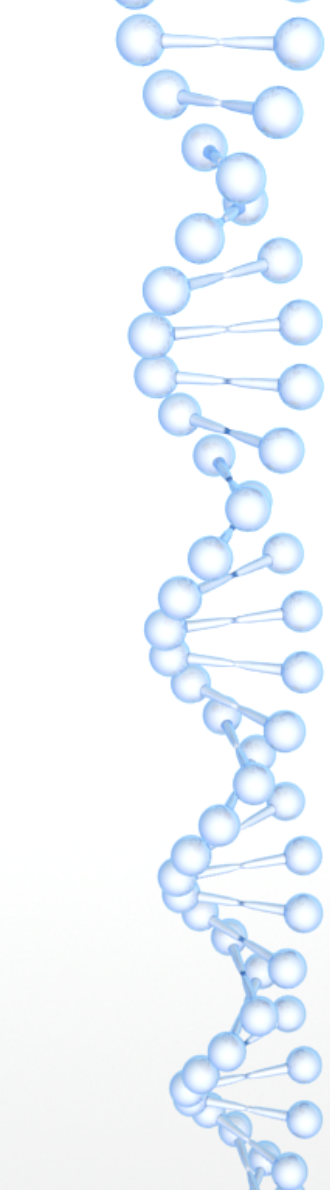


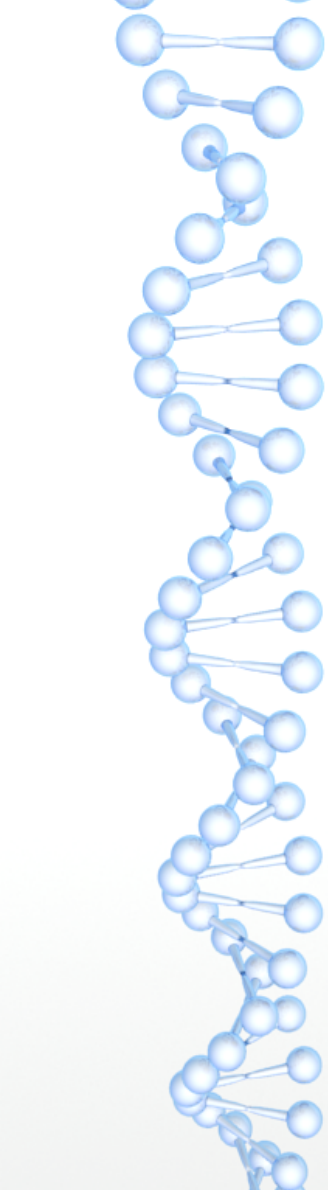
Configure Your Build



- The Android build system compiles app resources and source code, and packages them into APKs that you can test, deploy, sign, and distribute.
- Android Studio uses Gradle, an advanced build toolkit, to automate and manage the build process, while allowing you to define flexible custom build configurations.

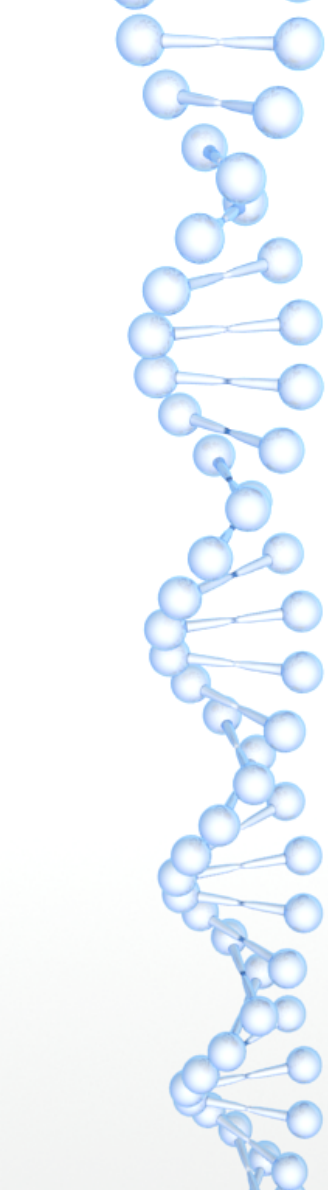


- Each build configuration can define its own set of code and resources, while reusing the parts common to all versions of your app.
- The Android plugin for Gradle works with the build toolkit to provide processes and configurable settings that are specific to building and testing Android applications.
- Gradle and the Android plugin run independent of Android Studio.
- This means that you can build your Android apps from within Android Studio, the command line on your machine, or on machines where Android Studio is not installed (such as continuous integration servers).



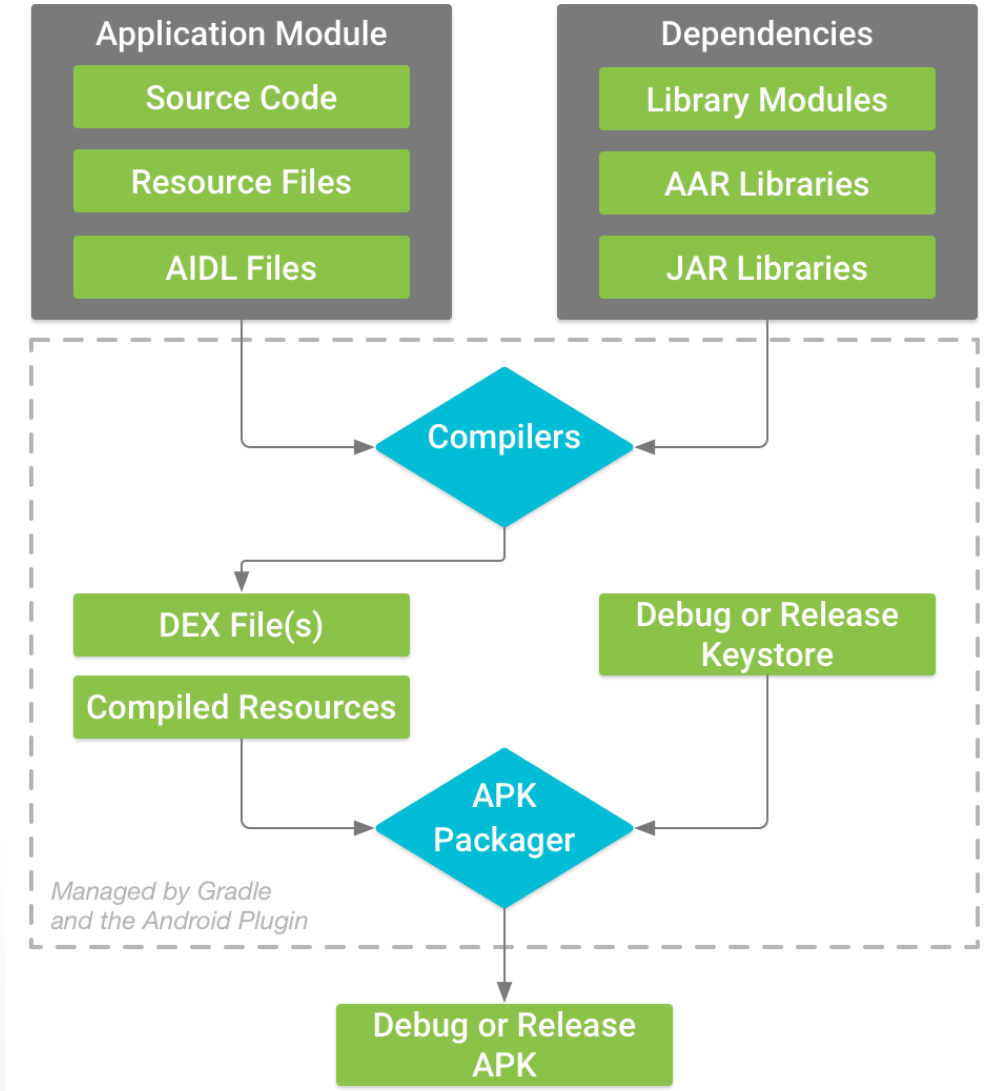
Cont ...

- The flexibility of the Android build system enables you to perform custom build configurations without modifying your app's core source files.



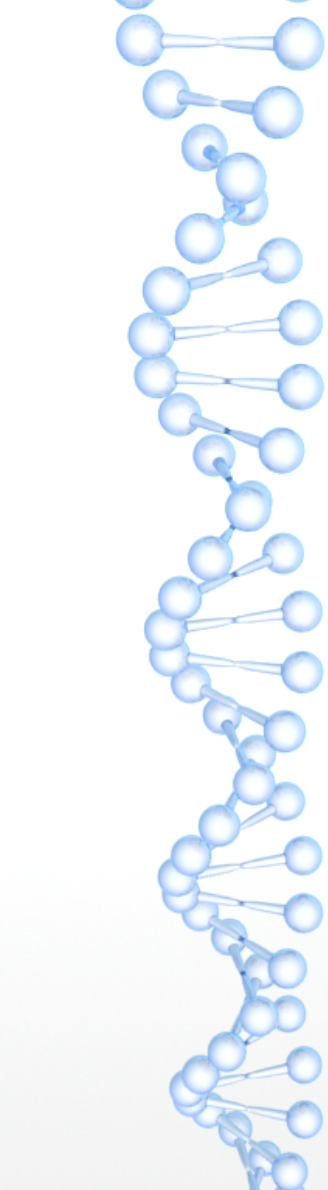
The Build Process

- The build process involves many tools and processes that convert your project into an Android Application Package (APK). T
- he build process is very flexible, so it's useful to understand some of what is happening under the hood.


Application Module
Source Code
Resource Files
AIDL Files
Dependencies
Library Modules
AAR Libraries
JAR Libraries
Compilers
DEX File(s)
Compiled Resources
Debug or Release Keystore
APK Packager
Managed by Gradle and the Android Plugin
Debug or Release APK

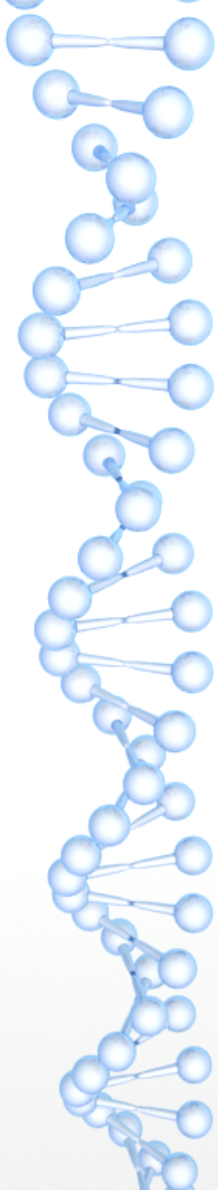
Cont ...

- The build process for a typical Android app module, as shown in figure 1, follows these general steps:
  - The compilers convert your source code into DEX (Dalvik Executable) files, which include the bytecode that runs on Android devices, and everything else into compiled resources.
  - The APK Packager combines the DEX files and compiled resources into a single APK. Before your app can be installed and deployed onto an Android device, however, the APK must be signed.



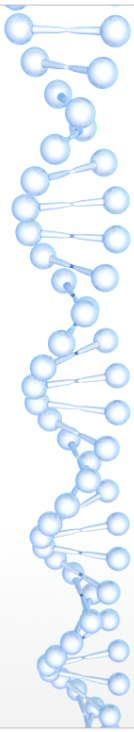
Cont ...

- The APK Packager signs your APK using either the debug or release keystore:
  - If you are building a debug version of your app, that is, an app you intend only for testing and profiling, the packager signs your app with the debug keystore. Android Studio automatically configures new projects with a debug keystore.
  - If you are building a release version of your app that you intend to release externally, the packager signs your app with the release keystore.

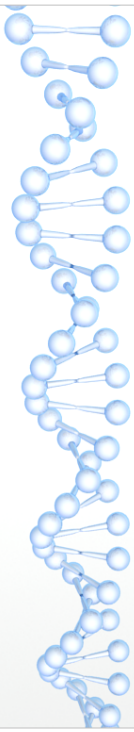


Cont ...

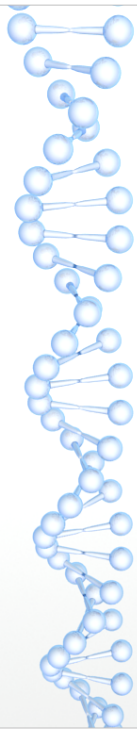
- Before generating your final APK, the packager uses the zipalign tool to optimize your app to use less memory when running on a device.



Configure Your Build



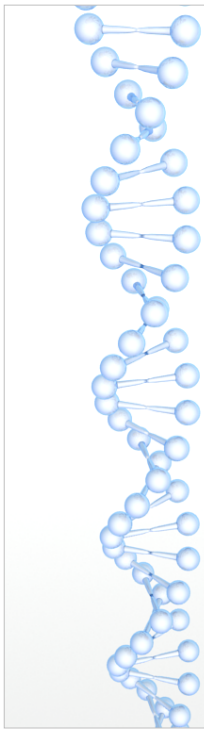
- The Android build system compiles app resources and source code, and packages them into APKs that you can test, deploy, sign, and distribute.
- Android Studio uses Gradle, an advanced build toolkit, to automate and manage the build process, while allowing you to define flexible custom build configurations.



- Each build configuration can define its own set of code and resources, while reusing the parts common to all versions of your app.
- The Android plugin for Gradle works with the build toolkit to provide processes and configurable settings that are specific to building and testing Android applications.
- Gradle and the Android plugin run independent of Android Studio.
- This means that you can build your Android apps from within Android Studio, the command line on your machine, or on machines where Android Studio is not installed (such as continuous integration servers).

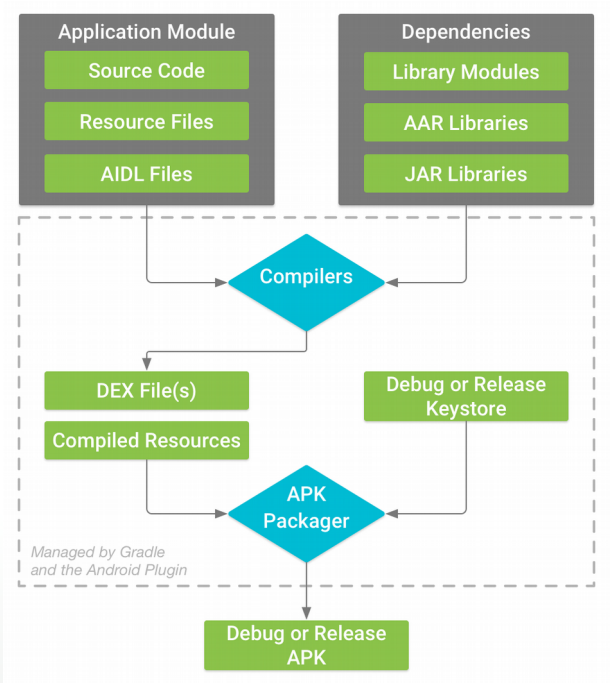
Cont ...

- The flexibility of the Android build system enables you to perform custom build configurations without modifying your app's core source files.



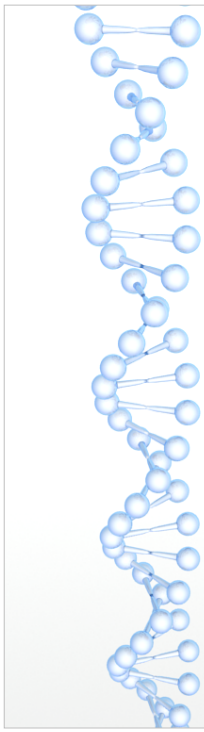
The Build Process

- The build process involves many tools and processes that convert your project into an Android Application Package (APK). T
- he build process is very flexible, so it's useful to understand some of what is happening under the hood.


Application Module
Source Code
Resource Files
AIDL Files
Dependencies
Library Modules
AAR Libraries
JAR Libraries
Compilers
DEX File(s)
Compiled Resources
Debug or Release Keystore
APK Packager
Managed by Gradle and the Android Plugin
Debug or Release APK

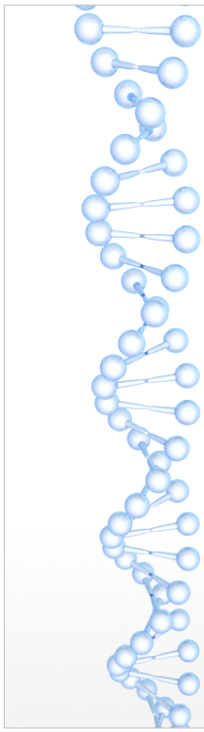
Cont ...

- The build process for a typical Android app module, as shown in figure 1, follows these general steps:
  - The compilers convert your source code into DEX (Dalvik Executable) files, which include the bytecode that runs on Android devices, and everything else into compiled resources.
  - The APK Packager combines the DEX files and compiled resources into a single APK. Before your app can be installed and deployed onto an Android device, however, the APK must be signed.



Cont ...

- The APK Packager signs your APK using either the debug or release keystore:
  - If you are building a debug version of your app, that is, an app you intend only for testing and profiling, the packager signs your app with the debug keystore. Android Studio automatically configures new projects with a debug keystore.
  - If you are building a release version of your app that you intend to release externally, the packager signs your app with the release keystore.



Cont ...

- Before generating your final APK, the packager uses the zipalign tool to optimize your app to use less memory when running on a device.