

Integrating AndroidX in processing-android

([issue #560](#))

What is AndroidX?

First things first. AndroidX stands for *Android EXtension Libraries*. *Artifacts within the Androidx namespace comprise the Android Jetpack libraries. Like the Support Library, libraries in the Androidx namespace ship separately from the Android platform and provide backward compatibility across Android releases.*

What are support libraries and why are they Needed ?

Different Android versions may have different names for libraries, methods, and functionality that are integrated within the operating system and won't be delivered within your app APK. So to make sure the compiler understands your code and executes the same or similar functionality in all devices, support libraries take control and carry out this job for us. Support libraries are shipped with your app and are not tied to the Android version.

What is the difference between the Support Library and the new Jetpack ?

Jetpack is better organized and will be more consistent across devices. In developer terms, Jetpack is a refactor of all the support libraries. Why do I say all? Because if you look at the support libraries, they had different versions (v4, v7, v12, etc.) which made them really confusing and was becoming a big mess. With this change, you don't have to worry about the version because Jetpack knows what code to use

Migrating from support libraries to AndroidX in processing-android:

The processing-core library is dependent on Android support libraries for various ui components like Fragment, FragmentActivity, FragmentManager, FragmentTransaction, etc.

The processing-android is build in such a way that android libraries(.aar) used in the project will be converted into java archives or better known by jars using the custom AarPlugin inside buildSrc directory, and all the jars will be added into the compileclasspath of the project as well and all these useful jars will be copied from the dependencies to the path_to_project_root/mode/mode.

Note: Before starting doing the migration, first find the official CSV mapping file of AndroidX provided by Android Team for ease of Migration. And also find the corresponding dependency jars which have to be copied to mode/mode. This can be find by debugging the core's build file.

To add AndroidX in processing-core follow:

1. Upgrade **api level to 28** (at least)
2. Make **gradle.properties** in root project
3. Add - **android.UseAndroidx = true** in above file.
4. Add dependencies in project level build.gradle file
5. Add **legacy support** for support libraries in dependencies initially for smooth migration
6. Replace **android.support.*** to **androidx.*** as required package from official csv mapping file provided by google

7. Find the corresponding jars (explained above) and replace their name in the core's build,gradle.kts file
8. Build the project and test it by importing *mode and sdk updater as modules in processing repo (pde)*

Hope this guide will help you in achieving your goal with my wonderful experience working with this amazing project.

Best,

Aditya Rana

GitHub: <https://github.com/ranaaditya>

LinkedIn: <https://linkedin.com/in/adityarananitt>