Libraries which migrate to KMP should specify default target architecture in POM files

Fixed Feature Request P1 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION az...@lyft.com created issue #1

Jetpack Compose version: 1.5.0-beta01

Steps to Reproduce or Code Sample to Reproduce:

1. Try to upgrade from Compose 1.5.0-alpha04 to 1.5.0-beta01
2. Observe dependency resolution failures with Bazel and rules_jvm_external, can't download .aar files

Stack trace (if applicable):

```
https://maven.google.com/androidx/compose/foundation/foundation-layout/1.5.0-beta01/foundation-layout-1.5.0-beta01.aar:
not found
```

Looks like now instead of publishing Android `.aar` files directly under Compose Maven artifacts, there is now additional `-android` artifact published which should be used to get `.aar` files.

However the `.pom` files don't point to `-android` artifacts correctly! While Gradle Module Metadata `.module` files do! This keeps working with Gradle, but breaks any other build tool that tries to r
`type` should be resolved normally.

Example: `androidx.compose.animation:animation:1.5.0-beta01` pom file doesn't point to -android artifact of `foundation-layout` correctly:

```
<dependency>
    <groupId>androidx.compose.foundation</groupId>
    <artifactId>foundation-layout</artifactId> <!-- Should be `-android` -->
    <version>1.5.0-beta01</version>
    <scope>runtime</scope>
    <type>aar</type>
</dependency>
```

I think:

1. `.pom` files should be fixed and point to `-android` artifacts since they're clearly trying to resolve `aar` on the artifact that doesn't have `aar` anymore
2. Such changes to packaging and publishing should be announced and confirmed with user community rather than just being pushed in an alpha-to-beta minor update :)

- Cross-linking with the issue in rules_jvm_external: https://github.com/bazelbuild/rules_jvm_external/issues/909
- Another issue where missing .aar files break Compose Layout Inspector in Android Studio https://issuetracker.google.com/issues/284998242

✓ Mentioned issues (3) ✓ Links (32)

Mentioned issues (3)

P1 Compose Layout Inspector breaks due to non-existent Compose UI AAR "https://issuetracker.google.com/284998242"

P1 Incorrect addition of <type>aar</type> to KMP artifact dependencies. "https://issuetracker.google.com/285600312"

-- -- "1.6.0-alpha01 of ui-geometry was cut from b/10325923 . Both that and the head build are missing the -android entry. Reassigning back to myself for investigation."

Links (32)

"Cross-linking with the issue in rules_jvm_external: https://github.com/bazelbuild/rules_jvm_external/issues/909"
"This breaking change doesn't seem to be announced anywhere in Compose release notes 1.5.0-beta01 (I've checked individual artifact release notes too) https://developer.android.com/jetpack/andr
"Great point, created http://issuetracker.google.com/issues/285400399 to add a release note."
"It is somewhat implicit. Created http://issuetracker.google.com/285401322 to make that explicit"
"Generally, we do try to do that if the cost to maintain that compatibility is low. An example for that is androidx rewriting of POM files to add <type>aar</type> as out of the box Gradle generated aa

See all related links

COMMENTS

au...@google.com <au...@google.com> #2

Status: Won't Fix (Intended Behavior)

We have started publishing artifacts in a Kotlin multiplatform (KMP) way as you spotted. By default, KMP artifacts are only fully working through Gradle (via gradle module files) and it falls b

We currently only support Gradle consumers of AndroidX artifacts, so we do not intend to add any workarounds on our publishing side to keep non-gradle systems working. KMP is a big shift
your build system will have to build proper support of KMP artifact consumption.

az...@lyft.com <az...@lyft.com> #3

Hi, thank you for quick reply!

1. This breaking change doesn't seem to be announced anywhere in Compose release notes `1.5.0-beta01` (I've checked individual artifact release notes too) https://developer.android.co
2. We never saw a statement in the docs that AndroidX artifacts only support Gradle consumers. Since you publish Maven artifacts it would be great to keep as much compatibility with M maybe POM files should be fixed to point to Android artifacts as they used to be otherwise POMs are completely useless/broken and should not be published?
3. While Gradle-compatible KMP consumers will work with `.module` files I think it is foundational for *AndroidX* artifacts to work by default for Android-compatible build tools like Bazel wit point, it is definitely nowhere near widely adopted in the industry as Maven POMs/BOMs).
4. *Our* build system is Google's Bazel :)

al...@gmail.com <al...@gmail.com> #4

To be clear, the build system we are using is Bazel (developed by Google). So I'm more curious why these decisions to only support gradle are being made without coordinating with the Baze with the Bazel team about this and is there an internal understanding of what the next steps would be to support this sudden change? It's one thing to make a decision to only support Gradle

be...@ben.cm <be...@ben.cm> #5

Can you elaborate on why pushing broken POM files that point at non-existent artifacts is the intended behavior?

For context, we use both okio and wire in our codebase both of which are fetched using rules_jvm_external and both of which work perfectly fine without Gradle.

au...@google.com <au...@google.com> #6

Re #3

> This breaking change doesn't seem to be announced anywhere in Compose release notes 1.5.0-beta01 (I've checked individual artifact release notes too) https://developer.android.com/jet

Great point, created http://issuetracker.google.com/issues/285400399 to add a release note.

> We never saw a statement in the docs that AndroidX artifacts only support Gradle consumers.

It is somewhat implicit. Created http://issuetracker.google.com/285401322 to make that explicit

> Since you publish Maven artifacts it would be great to keep as much compatibility with Maven POMs as possible

Generally, we do try to do that if the cost to maintain that compatibility is low. An example for that is androidx rewriting of POM files to add <type>aar</type> as out of the box Gradle g

We already have several features that are *not* working at all when not using Gradle module files, for example dependency constraints to align compatible versions of libraries. There is no way `android` artifacts within non-android artifacts, but that is fragile and costly to maintain over time. The real solution is for other build systems to start reading gradle module as it is a richer fo

> If your publishing falls back to -jvm variant but in Compose you don't publish JVM variant, then maybe POM files should be fixed to point to Android artifacts as they used to be otherwise P

It is possible, but fragile and comes with a maintenance cost that at the moment we are not staffed for.

> While Gradle-compatible KMP consumers will work with .module files I think it is foundational for AndroidX artifacts to work by default for Android-compatible build tools like Bazel

As noted above, we are not staffed to support other build systems, so the fact it happened to work before, does not imply continuing support. We do generally try to do our best to keep our to systems.

> Our build system is Google's Bazel :)

Google's external build system offering for Android developers is Gradle. I sadly have no powers to change that.

Re #4

> To be clear, the build system we are using is Bazel (developed by Google). So I'm more curious why these decisions to only support gradle are being made without coordinating with the Baz

As stated above, Google's external offering for Android devs is Gradle. Android Studio is built on top of that, including most of the other Android development tools we offer. Bazel is its own t

> Has the Android X team coordinated with the Bazel team about this and is there an internal understanding of what the next steps would be to support this sudden change?

I am not sure what exactly we'd be coordinating about. My team does help the internal google app builds to work with updated artifacts using Blaze. As noted above, KMP artifacts are becom

> It's one thing to make a decision to only support Gradle but it feels weird for the change to be such a surprise without any external communication regarding Bazel support.

We never supported Bazel, created http://issuetracker.google.com/285401322 to make it clear in our documentation.

Re #5

> Can you elaborate on why pushing broken POM files that point at non-existent artifacts is the intended behavior?

It is not non-existent artifacts, they do exist, for `androidx.compose.material:material:1.5.0` POM files points you to `https://dl.google.com/android/maven2/androidx/compose/ma` fallback to be android instead of this file.

> For context, we use both okio and wire in our codebase both of which are fetched using rules_jvm_external and both of which work perfectly fine without Gradle.

okio does not have android artifacts, you are getting JVM platform artifacts. If you look at their pom file you'll see that they implicitly point to https://repo1.maven.org/maven2/com/squar

```
<dependency>
  <groupId>com.squareup.okio</groupId>
  <artifactId>okio-jvm</artifactId>
  <version>3.3.0</version>
  <scope>compile</scope>
</dependency>
```

au...@google.com <au...@google.com> #7

One additional clarification from discussion offline, it seems that -android artifact pom files have dependencies that add `<type>aar</type>` incorrectly as it points to KMP artifacts.

example from `foundation-layout` pom file has the following

```
    <dependency>
      <groupId>androidx.compose.animation</groupId>
      <artifactId>animation-core</artifactId>
      <version>1.2.1</version>
      <scope>runtime</scope>
      <exclusions>
        <exclusion>
          <artifactId>kotlin-stdlib-common</artifactId>
          <groupId>org.jetbrains.kotlin</groupId>
        </exclusion>
        <exclusion>
          <artifactId>kotlin-test-common</artifactId>
          <groupId>org.jetbrains.kotlin</groupId>
        </exclusion>
        <exclusion>
          <artifactId>kotlin-test-annotations-common</artifactId>
          <groupId>org.jetbrains.kotlin</groupId>
        </exclusion>
      </exclusions>
      <type>aar</type>
    </dependency>
```

This is caused by our pom rewrite This is a somewhat tangential issue changing what platform we default to in pom files.

au...@google.com <au...@google.com> #8

I've created https://github.com/bazelbuild/rules_android/issues/92 to track rules_android having proper support of consuming Android variant of KMP artifacts.

ar...@grabtaxi.com <ar...@grabtaxi.com> #9

I think it should be done via rules_jvm_external https://github.com/bazelbuild/rules_jvm_external/issues/909

au...@google.com <au...@google.com> #10

Status: Assigned (reopened)

We are investigating what a workaround would look like for defaulting to the Android variant would look like.

As for #7, I split that into https://issuetracker.google.com/285600312

ar...@grabtaxi.com <ar...@grabtaxi.com> #11

EDIT: I saw the status update only after posting this below comment

Hi regardless of official Bazel support or not, the expectation here is to at least have POM files working for Android builds. Since Gradle Module Metadata is exclusively used for multi-platfor is need to make JVM as the default platform for a POM graph containing android artifact (aar) no?

This won't be a breaking change since Gradle would continue using Module Metadata while others can use POM files to resolve android dependencies. That was my understanding from G

> It explains to Gradle that a better module metadata file exists and that it should use it instead. It doesn't mean that consumption from Maven or Ivy would be broken either, only that it work

Guess what I am proposing is already mentioned in the report i.e to have

```
<dependency>
    <groupId>androidx.compose.foundation</groupId>
    <artifactId>foundation-layout</artifactId> <!-- Should be `-android` -->
    <version>1.5.0-beta01</version>
    <scope>runtime</scope>
    <type>aar</type>
</dependency>
```

Since the correct path to aar is and also `type` is indicating it to be `aar`.

https://dl.google.com/android/maven2/androidx/compose/foundation/foundation-layout-android/1.5.0-beta01/foundation-layout-android-1.5.0-beta01.aar

Message last modified on Jun 3, 2023 04:14AM

al...@google.com <al...@google.com> #12

Yeah, it'll be something along the lines of comment#11. We won't have something ready for the next Compose `beta`, but I'll try to hold `rc` until we can get this addressed.

yb...@google.com <yb...@google.com> #13

Btw, we might need something more complex than just defaulting to android since we do have some java/native KMP projects in datastore that do not need any android specific code.

au...@google.com <au...@google.com> #14

I think we need to apply this to compose artifacts that shipped as android libraries only

al...@google.com <al...@google.com> #15

Still need to put together a design doc, but tentatively this'll be configurable in the `androidx` block.

```
androidXMultiplatform {
  defaultPomPlatform = KotlinPlatformType.jvm
}
```

All of the naming there is TBD so please don't paint the shed.

Message last modified on Jun 3, 2023 04:47AM

al...@google.com <al...@google.com>

Accepted by al...@google.com.

al...@google.com <al...@google.com> #16

aosp/2612118 pending, just need to finish writing the tests.

ap...@google.com <ap...@google.com> #17

Project: platform/frameworks/support
Branch: androidx-main

commit 95899324a20f05b5ba53b7da279bbb2e8ddabdd0
Author: Alan Viverette <alanv@google.com>
Date: Thu Jun 08 14:34:21 2023

Refactor src/test/kotlin to src/test/java for interop

Otherwise we can't have Java files in the source set.

Bug: 285353844
Test: buildSrc-tests suite
Change-Id: I7554a338664354b19e9325e95f6a2451a7760c06

M buildSrc-tests/src/test/java/androidx/build/AndroidXImplPluginTest.kt
M buildSrc-tests/src/test/java/androidx/build/KmpPlatformsTest.kt
M buildSrc-tests/src/test/java/androidx/build/LibraryVersionsServiceTest.kt
M buildSrc-tests/src/test/java/androidx/build/MavenUploadHelperTest.kt
M buildSrc-tests/src/test/java/androidx/build/SdkResourceGeneratorTest.kt
M buildSrc-tests/src/test/java/androidx/build/SettingsParserTest.kt
M buildSrc-tests/src/test/java/androidx/build/VersionTest.kt
M buildSrc-tests/src/test/java/androidx/build/buildInfo/CreateLibraryBuildInfoFileTaskTest.kt
M buildSrc-tests/src/test/java/androidx/build/checkapi/CheckApiTest.kt
M buildSrc-tests/src/test/java/androidx/build/dependencyTracker/AffectedModuleDetectorImplTest.kt
M buildSrc-tests/src/test/java/androidx/build/dependencyTracker/AttachLogsTestRule.kt
M buildSrc-tests/src/test/java/androidx/build/dependencyTracker/BuildPropParserTest.kt
M buildSrc-tests/src/test/java/androidx/build/dependencyTracker/ChangeInfoGitClientTest.kt
M buildSrc-tests/src/test/java/androidx/build/dependencyTracker/GitRunnerGitClientTest.kt
M buildSrc-tests/src/test/java/androidx/build/dependencyTracker/ProjectGraphTest.kt
M buildSrc-tests/src/test/java/androidx/build/dependencyallowlist/DependencyAllowlistTest.kt
M buildSrc-tests/src/test/java/androidx/build/metalava/CheckApiCompatibilityTaskTest.kt
M buildSrc-tests/src/test/java/androidx/build/metalava/UpdateApiTaskTest.kt
M buildSrc-tests/src/test/java/androidx/build/playground/VerifyPlaygroundGradleConfigurationTaskTest.kt
M buildSrc-tests/src/test/java/androidx/build/testConfiguration/AndroidTestConfigBuilderTest.kt

https://android-review.googlesource.com/2619771

ap...@google.com <ap...@google.com> #18

Project: platform/frameworks/support
Branch: androidx-main

commit 5b2d285fe8a604f046ea355aa14d461130d75280
Author: Alan Viverette <alanv@google.com>
Date: Fri Jun 02 16:51:58 2023

Insert default platform dependencies for KMP anchor publications

Specifies default for projects with multiple platforms.

Bug: 285353844
Test: MavenUploadHelperTest
Change-Id: I1ea37fa55db8089bfc597b6408fcf13e41ba0c8d

M annotation/annotation/build.gradle
M buildSrc-tests/src/test/java/androidx/build/MavenUploadHelperTest.kt
A buildSrc-tests/src/test/java/androidx/build/testutils/PomTestData.kt
A buildSrc-tests/src/test/java/androidx/build/testutils/XmlProviderImpl.java
M buildSrc/private/src/main/kotlin/androidx/build/AndroidXImplPlugin.kt
M buildSrc/private/src/main/kotlin/androidx/build/AndroidXMultiplatformExtension.kt
M buildSrc/private/src/main/kotlin/androidx/build/MavenUploadHelper.kt
M buildSrc/public/src/main/kotlin/androidx/build/KmpPlatforms.kt
M collection/collection/build.gradle
M compose/animation/animation-core/build.gradle
M compose/animation/animation-graphics/build.gradle
M compose/animation/animation/build.gradle
M compose/foundation/foundation-layout/build.gradle
M compose/foundation/foundation/build.gradle
M compose/material/material-icons-core/build.gradle
M compose/material/material-icons-extended/build.gradle
M compose/material/material-ripple/build.gradle
M compose/material/material/build.gradle
M compose/material3/material3-adaptive/build.gradle
M compose/material3/material3-window-size-class/build.gradle
M compose/material3/material3/build.gradle
M compose/runtime/runtime-saveable/build.gradle
M compose/runtime/runtime/build.gradle
M compose/ui/ui-geometry/build.gradle
M compose/ui/ui-graphics/build.gradle
M compose/ui/ui-test-junit4/build.gradle
M compose/ui/ui-test/build.gradle
M compose/ui/ui-text/build.gradle
M compose/ui/ui-tooling-data/build.gradle
M compose/ui/ui-tooling-preview/build.gradle
M compose/ui/ui-tooling/build.gradle
M compose/ui/ui-unit/build.gradle
M compose/ui/ui-util/build.gradle
M compose/ui/ui/build.gradle
M datastore/datastore-core-okio/build.gradle
M datastore/datastore-core/build.gradle
M datastore/datastore-preferences-core/build.gradle
M datastore/datastore-preferences/build.gradle
M datastore/datastore/build.gradle
M development/project-creator/compose-template/groupId/artifactId/build.gradle

https://android-review.googlesource.com/2612118

al...@google.com <al...@google.com> #19

Marked as fixed.

And of course, I forgot to put a `Relnote` so this bug won't get updates when this lands in releases.

Build cut-off isn't until tomorrow for the next batch of releases, so this will get picked up by all of the Compose `1.6.0-alpha` releases.

al...@google.com <al...@google.com> #20

Status: Assigned (reopened)

Juan, do you want to pick these up as cherry-picks for Compose beta? We'd need aosp/2619771 and aosp/2612118, which should be a clean merge (fingers crossed).

ju...@google.com <ju...@google.com> #21

we can run a build and cherrypick them into the beta

au...@google.com <au...@google.com> #22

alanv@ can we test to make sure it resolves when using poms only? (should be able to emulate by deleting all the .module files)

al...@google.com <al...@google.com> #23

> (should be able to emulate by deleting all the .module files)

I was planning on creating a standalone project with a local Maven repository and deleting the `.module` files, which is going to be a pain. Any chance you had an approach that might be mor

yb...@google.com <yb...@google.com> #24

androidx.dev snapshots should already be serving these new versions.

https://androidx.dev/snapshots/builds https://androidx.dev/snapshots/builds/10321535/artifacts/repository/androidx/compose/foundation/foundation-layout/1.6.0-SNAPSHOT/maven-me

al...@google.com <al...@google.com> #25

> That one is still showing jar as default though :/

The POM correctly depends on the platform-specific artifact. We still ought to remove the empty JARs, though -- I am not sure why those are there in the first place, but I don't think they are h

al...@google.com <al...@google.com> #26

> androidx.dev snapshots should already be serving these new versions.

Yes but these include `.module` files and I'm not going to test using a Bazel build.

al...@google.com <al...@google.com> #27

Verified locally using anchor artifacts for libraries with `-jvm` and `-android` default platforms.

al...@google.com <al...@google.com> #28

> we can run a build and cherrypick them into the beta

It looks like this didn't happen? Neither of aosp/2619771 or aosp/2612118 appear in the `androidx-compose-beta-release` branch.

ap...@google.com <ap...@google.com> #29

Project: platform/frameworks/support
Branch: androidx-main

commit 328693f8e983178c30c4efe54d876b5f4a9cb05b
Author: Alan Viverette <alanv@google.com>
Date: Fri Jun 09 16:05:49 2023

Clean up KmpPlatforms implementation

- Remove unused constants
- Change naming to reflect platform groups versus platforms
- Cache parsed enabled platforms flag

Retains the unused mapping of platform to group, since that seems like
useful information.

I'm not enthusiastic about the naming of the enableXxx() functions, since
they are actually isXxxEnabled(), but let's not rock the boat too much.

Bug: 285353844
Test: ./gradlew :collection:collection:publish
Change-Id: Id2fe7c77d4d83b04bc0a932a72a1518ce0bf0491

M buildSrc-tests/src/test/java/androidx/build/KmpPlatformsTest.kt
M buildSrc/public/src/main/kotlin/androidx/build/KmpPlatforms.kt

https://android-review.googlesource.com/2621291

az...@lyft.com <az...@lyft.com> #30

Just reporting: we don't see fix in just released `1.6.0-alpha01`

androidx.compose.animation:animation:1.6.0-alpha01 still incorrectly depends on androidx.compose.foundation:foundation-layout:1.6.0-alpha01 `jar` rather than `aar` or `-android`

Should we expect a cherry-pick or new release cut?

al...@google.com <al...@google.com> #31

Change landed in ab/10313259, `1.6.0-alpha01` was cut from ab/10325923 -- should be included.

POM looks correct, as well:

```
<dependencies>
  <dependency>
    <groupId>androidx.compose.animation</groupId>
    <artifactId>animation-android</artifactId>
    <version>1.6.0-alpha01</version>
    <scope>compile</scope>
  </dependency>
  ...
```

Similarly, the POM for `androidx.compose.foundation:foundation-layout:1.6.0-alpha01` specifies a dependency on the `-android` artifact:

```
<dependencies>
  <dependency>
    <groupId>androidx.compose.foundation</groupId>
    <artifactId>foundation-layout-android</artifactId>
    <version>1.6.0-alpha01</version>
    <scope>compile</scope>
  </dependency>
  ...
```

This all looks correct. Can you clarify what issue you're seeing?

az...@lyft.com <az...@lyft.com> #32

So is the fix for *all* KMP androidx deps to depend on their `-android` versions in their regular `pom`?

Because I'm seeing some number of `1.6.0-alpha01` deps that don't have such a connection in their `pom` and they're now `.jar` while they were `.aar` before:

- androidx.compose.runtime:runtime does *not* depend on androidx.compose.runtime:runtime-android
- androidx.compose.ui:ui-geometry does *not* depend on androidx.compose.ui:ui-geometry-android

I haven't checked all of androidx artifacts, but I was able to quickly find these ^

al...@google.com <al...@google.com> #33

Reassigned to al...@google.com.

`1.6.0-alpha01` of `ui-geometry` was cut from b/10325923. Both that and the head build are missing the `-android` entry. Reassigning back to myself for investigation.

> and they're now .jar while they were .aar before:

The `<type>aar</type>` entry was incorrect. This was fixed by aosp/2613770.

al...@google.com <al...@google.com> #34

Accepted by al...@google.com.

The affected libraries have no dependencies, but inserting a dependency on the default platform is predicated on finding the `<dependencies>` element. Quick fix.

al...@google.com <al...@google.com> #35

Also, potentially more concerning, the Groovy XML parser that's used by Gradle strips comments -- including the comment that's supposed to tell Gradle to use Gradle Metadata instead of PC

ap...@google.com <ap...@google.com> #36

Marked as fixed.

Project: platform/frameworks/support
Branch: androidx-main

commit 7aa4e307db78e68bdc78fa42f0537c6e5c941a1a
Author: Alan Viverette <alanv@google.com>
Date: Tue Jun 27 10:28:37 2023

> Insert dependencies element containing default platform dependency
>
> Previously insertion would fail silently if there were no other
> dependencies. This CL also updates the other failure modes to throw
> exceptions.

Fixes: 285353844
Test: MavenUploadHelperTest
Change-Id: I6e88800f3c8358809856af1c72861217bf067866

M buildSrc-tests/src/test/java/androidx/build/MavenUploadHelperTest.kt
M buildSrc-tests/src/test/java/androidx/build/testutils/PomTestData.kt
M buildSrc/private/src/main/kotlin/androidx/build/MavenUploadHelper.kt

https://android-review.googlesource.com/2640771

---

al...@gmail.com <al...@gmail.com> #37

Do you have an idea of which versions of Compose will ship with the fixed artifacts?

---

na...@google.com <na...@google.com> #38

Compose beta-03 (released yesterday) has the fixes merged from above

---

al...@gmail.com <al...@gmail.com> #39

Some of the libraries still seem to be missing packaging, while others have it.

- Missing packaging: https://maven.google.com/web/index.html?q=foundation-lay#androidx.compose.foundation:foundation-layout:1.5.0-beta03
- Packaging specified: https://maven.google.com/web/index.html?q=ui-android#androidx.compose.ui:ui-android:1.5.0-beta03

Would specifying `<packaging>jar</packaging>` in the pom file for the jars fix Bazel coursier fetch?

```
Error in fail: Error while fetching artifact with coursier: OpenJDK 64-Bit Server VM warning: Options -Xverify:none and -noverify were deprecated in JDK 1
Error fetching artifacts:
https://maven.google.com/androidx/compose/foundation/foundation-layout/1.5.0-beta03/foundation-layout-1.5.0-beta03.aar: not found: https://maven.google.co
https://maven.google.com/androidx/compose/animation/animation-core/1.5.0-beta03/animation-core-1.5.0-beta03.aar: not found: https://maven.google.com/andro
https://maven.google.com/androidx/compose/ui/ui-util/1.5.0-beta03/ui-util-1.5.0-beta03.aar: not found: https://maven.google.com/androidx/compose/ui/ui-uti
https://maven.google.com/androidx/compose/foundation/foundation/1.5.0-beta03/foundation-1.5.0-beta03.aar: not found: https://maven.google.com/androidx/com
https://maven.google.com/androidx/compose/ui/ui/1.5.0-beta03/ui-1.5.0-beta03.aar: not found: https://maven.google.com/androidx/compose/ui/ui/1.5.0-beta03/
https://maven.google.com/androidx/compose/runtime/runtime-saveable/1.5.0-beta03/runtime-saveable-1.5.0-beta03.aar: not found: https://maven.google.com/and
https://maven.google.com/androidx/compose/animation/animation/1.5.0-beta03/animation-1.5.0-beta03.aar: not found: https://maven.google.com/androidx/compos
https://maven.google.com/androidx/compose/ui/ui-text/1.5.0-beta03/ui-text-1.5.0-beta03.aar: not found: https://maven.google.com/androidx/compose/ui/ui-tex
https://maven.google.com/androidx/compose/runtime/runtime/1.5.0-beta03/runtime-1.5.0-beta03.aar: not found: https://maven.google.com/androidx/compose/runt
```

---

al...@google.com <al...@google.com> #40

> Would specifying `<packaging>jar</packaging>` in the pom file for the jars fix Bazel coursier fetch?

The default packaging when no `<packaging>` element has been specified is `jar`. This sounds like an issue with Bazel.

> `https://maven.google.com/androidx/compose/foundation/foundation-layout/1.5.0-beta03/foundation-layout-1.5.0-beta03.aar: not found: https://maven.google.`

There's nothing in the `pom` that would indicate an `aar` is present. Again, this seems like a Bazel issue.

---

te...@gmail.com <te...@gmail.com> #41

Hi, after updating compose to 1.5.0-beta03 I have such error:

```
Execution failed for task ':app:mergeDevDebugNativeLibs'.
> Could not resolve all files for configuration ':app:devDebugRuntimeClasspath'.
   > Failed to transform runtime-1.5.0-beta03.aar (androidx.compose.runtime:runtime:1.5.0-beta03) to match attributes {artifactType=android-jni, org.gradl
      > Could not find runtime-1.5.0-beta03.aar (androidx.compose.runtime:runtime:1.5.0-beta03).
        Searched in the following locations:
```

I don't have androidx.compose.runtime:runtime added as explicit dependency, it's transitive one

Message last modified on Jul 24, 2023 10:09PM

---

ar...@grabtaxi.com <ar...@grabtaxi.com> #42

This issue is also affecting androidx.annotation artifact.

`androidx.annotation:1.6.0` POM's dependencies should point to `andoridx.annotation-jvm:1.6.0`, but it does not. Even though `annnotation`s build.gradle does specify PlatformIdent

This not only affects bazel but also any type of resolution as `pom.xml` is not valid here.

`androidx.annotation:1.6.0` POM:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4
  <!-- This module was also published with a richer model, Gradle metadata,  -->
  <!-- which should be used instead. Do not delete the following line which  -->
  <!-- is to indicate to Gradle or any Gradle module metadata file consumer  -->
  <!-- that they should prefer consuming it instead. -->
  <!-- do_not_remove: published-with-gradle-metadata -->
  <modelVersion>4.0.0</modelVersion>
  <groupId>androidx.annotation</groupId>
  <artifactId>annotation</artifactId>
  <version>1.6.0</version>
  <name>Android Support Library Annotations</name>
  <description>The Support Library is a static library that you can add to your Android application in order to use APIs that are either not available for
  <url>https://developer.android.com/jetpack/androidx/releases/annotation#1.6.0</url>
  <inceptionYear>2013</inceptionYear>
  <licenses>
    <license>
      <name>The Apache Software License, Version 2.0</name>
      <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
      <distribution>repo</distribution>
    </license>
  </licenses>
  <developers>
    <developer>
      <name>The Android Open Source Project</name>
    </developer>
  </developers>
  <scm>
    <connection>scm:git:https://android.googlesource.com/platform/frameworks/support</connection>
    <url>https://cs.android.com/androidx/platform/frameworks/support</url>
  </scm>
  <dependencies>
    <dependency>
      <groupId>org.jetbrains.kotlin</groupId>
      <artifactId>kotlin-stdlib</artifactId>
      <version>1.8.0</version>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
</project>
```

The root cause seems to be due to configuration caching?

https://github.com/gradle/gradle/issues/18369

> I'm observing that when a project depends on "org.jetbrains.kotlinx:kotlinx-coroutines-core:1.5.0", the published .pom file is different based on whether the configuration cache was reused

> In particular the difference I'm observing is that the published .pom file lists a dependency of "kotlinx-coroutines-core-jvm" if the build did not load from the configuration cache, but lists a c

Not sure if androidx:annotation was published with CC enabled.

It seems a similar issue was handled for coroutines specifically here https://github.com/androidx/androidx/blob/888da7eec8266ecec3c9e229581ba40b076588b7/buildSrc/private/src/mair

Message last modified on Jul 26, 2023 08:35PM

---

al...@google.com <al...@google.com> #43

`androidx.annotation:1.6.0` was released on Feb 22, this issue was fixed Jun 13. A version of annotation that contains the fix will be going out with today's batch of releases.

---

ar...@grabtaxi.com <ar...@grabtaxi.com> #44

Thanks! I meant to say since 1.6.0, since I could see it present till 1.7.0-alpha02

---

et...@spotify.com <et...@spotify.com> #45

Hey, we're still having trouble upgrading to compose `1.5.0-RC01` (which was released yesterday). First of all I just want to confirm, as Bazel users with vanilla POMs, should we still be using use the `-android` ones? Either way, both options still don't work for us, at least not OOTB.

For example, the POM for `androidx.compose.foundation:foundation` has a mixture of non `-android` deps and a single `-android` dep:

non `-android` dependency:

```
<dependency>
  <groupId>androidx.compose.animation</groupId>
  <artifactId>animation</artifactId>
  <version>1.5.0-rc01</version>
  <scope>runtime</scope>
</dependency>
```

`-android` dependency:

```
<dependency>
  <groupId>androidx.compose.foundation</groupId>
  <artifactId>foundation-android</artifactId>
  <version>1.5.0-rc01</version>
  <scope>runtime</scope>
</dependency>
```

`animation` also has an `animation-android` AAR artifact, so why is there a difference here?

If we look at the POM for the `-android` version of `foundation`, so `androidx.compose.foundation:foundation-android`, all dependencies are on `-android` except one on `foundatio`

```
<dependency>
  <groupId>androidx.compose.foundation</groupId>
  <artifactId>foundation-layout</artifactId>
  <version>[1.5.0-rc01]</version>
  <scope>runtime</scope>
```

Also those square brackets around the version are odd, is that valid POM-wise?

Other artifacts are also inconsistent like this, which I believe is why the resolving is still broken for us using the current version of `rules_jvm_external` + `coursier`. I also looked at recent S

Message last modified on Jul 28, 2023 06:43AM

al...@google.com <al...@google.com> #46

> First of all I just want to confirm, as Bazel users with vanilla POMs, should we still be using the non `-android` coordinates and then internally it should know to choose the `-android` ones

Anchor artifact POMs now depend on default platform artifacts, so you should use the non `-android` coordinates.

> so why is there a difference here?

The `available-at` artifact from the default platform in the Gradle Metadata has been inserted as a POM dependency. None of the other dependencies have been modified, which means if t

This continues transitively for all multiplatform artifacts.

> which I believe is why the resolving is still broken for us using the current version of

Can you explain in what way resolving is broken?

et...@spotify.com <et...@spotify.com> #47

Basically it's the same type of failure the original author posted; while resolving the transitive deps it will try to grab an AAR from the non `-android` artifacts, which fails since those are non-

I can try to come up with a small Bazel + rules_jvm_external reproducer project tomorrow if it helps.

al...@google.com <al...@google.com> #48

> Basically it's the same type of failure the original author posted; while resolving the transitive deps it will try to grab an AAR from the non -android artifacts, which fails since those are non-

There's no `<packaging>aar</packaging>` element in the POM, so it sounds like Bazel must be hard-coding some assumptions. We cannot do anything about that from Jetpack. You will nee

az...@lyft.com <az...@lyft.com> #49

Hi friends from Spotify!

We've experienced exactly the same issue with Bazel + rules_jvm_external at Lyft where for some artifacts rules_jvm_external will end up trying to download `.aar` for an Androidx library that

You don't need to create a reproducible example, we've traced the issue down to Coursier (used by rules_jvm_external) and filed it there with minimal reproducible example with AndroidX libs

HOWEVER:

1. We were able to solve it by finding a slice of AndroidX libraries versions where all libs are changed to jar/aar releases and have no incorrect referencing of packaging, basically we had t
2. After debugging the issue we've concluded that if lib A says it wants an `.aar` of a particular transitive dep B, but we force version of B to when it became a different file like `.jar` then i format of lib B so the failure is okay-ish.

et...@spotify.com <et...@spotify.com> #50

Thanks Lyft friends! The coursier bug is very useful and it does prove that there's an issue with how 'Bazel' resolves the repacked JC artifacts. However I'm still struggling with a reasonable w

> We were able to solve it by finding a slice of AndroidX libraries versions where all libs are changed to jar/aar releases and have no incorrect referencing of packaging, basically we had to up

I was wondering if you could share how your setup currently looks to make this work; like which library versions?

> After debugging the issue we've concluded that if lib A says it wants an .aar of a particular transitive dep B, but we force version of B to when it became a different file like .jar then it's COR of lib B so the failure is okay-ish.

By forcing here, does that mean that in `rules_jvm_external` you were using the `pinned` version resolution strategy and listing these specific versions? We do that too, and I'm wondering h

Message last modified on Jul 29, 2023 04:37AM

va...@gmail.com <va...@gmail.com> #51

I was able to fix similar issue with this code, maybe it can be helpful for you too.

```
allprojects {
    configurations.all {
        resolutionStrategy.eachDependency {
            if (requested.group == "androidx.datastore" && requested.name == "datastore-preferences") {
                artifactSelection {
                    selectArtifact(
                        /* type */ "jar",
                        /* extension */ "jar",
                        /* classifier */ null,
                    )
                }
            }
        }
    }
}
```