


FixedBugP3


+ Add Hotlist

 STATUS UPDATE No update yet.

Edit

 DESCRIPTION

[Deleted User] created issue [#1](#)

May 19, 2022 07:11PM 

AGP 4.1.0 introduced a new default tool for building apks called zipflinger/zip2zip. It breaks my native Android app that has integrated Xamarin/Mono .dlls, because these .dlls are stored as compressed entries (or "DEFLATED") instead of ("STORED" like in AGP Version 4.0.1 and before). Xamarin/Mono reads the .dlls via direct memory mapping from the apk without decompression (which I think is also the default way for libraries build with ndk?)

I circumvented this issue by placing the following line in my `gradle.properties`:

```
android.useNewApkCreator=false
```

Building like this with AGP 7.2.0 shows an error that this option will be removed in AGP 8.0, thats why I created this issue.

So what I need is an ability to configure zipflinger which files should not be stored as compressed entries in the final apk.

Related issues on Github:

- Xamarin/MonoDroid (with reproduction steps): <https://github.com/xamarin/xamarin-android/issues/6838>
- KotlinAppWithXamarinDependency: <https://github.com/royd/KotlinAppWithXamarinDependency/issues/8>

Blogpost by Github User [Tinker-S](#) **fixing the issue by recompiling zip2zip:**

<https://copyfuture.com/blogs-details/20210119115509664T>

Studio Build: Android Studio Chipmunk | 2021.2.1 Version of Gradle Plugin: 7.2.0 Version of Gradle: 7.3.3 Version of Java: 11.0.12 OS: MacOS Big Sur 11.6.5

Reporter

 [Deleted User]

Type

Bug

Priority

P3

Severity

S3

Status

Fixed

Access

Default access 

Assignee

 sp...@google.c

Verifier

--

Collaborators

CC

 [Deleted User]

AOSP ID

--

Blocking Release

--

Release Status

--

Found In

--

Targeted To

--

Verified In

--


In Prod




Show 1 additional field

✓ Mentioned issues (1) ✓ Links (3)

Hide all

 Mentioned issues (1)

P2 Provide dsl to specify noCompress patterns for native libraries "This seems related to [Issue 200272774](#)" sp...@ [#4](#)

 Links (3)


"Xamarin/MonoDroid (with reproduction steps): <https://github.com/xamarin/xamarin-android/issues/6838>" nobody@ [#1](#), nobody@ [#5](#), sp...@ [#7](#), sp...@ [#11](#)

"KotlinAppWithXamarinDependency: <https://github.com/royd/KotlinAppWithXamarinDependency/issues/8>" nobody@ [#1](#)


"<https://copyfuture.com/blogs-details/20210119115509664T>..." nobody@ [#1](#)

COMMENTS



All comments ▾ ↓ Oldest first

 mm...@google.com <mm...@google.com> May 20, 2022 03:22AM

Assigned to an...@google.com.



 je...@google.com <je...@google.com> May 25, 2022 05:56AM

Reassigned to sp...@google.com.

 sp...@google.com <sp...@google.com> [#2](#) May 25, 2022 06:43AM 

Can you try adding the following to your build.gradle and see if it works:

```
android {
  aaptOptions {
    noCompress 'dll'
  }
}
```

 [Deleted User] <[Deleted User]> [#3](#) May 25, 2022 10:12PM 

I've tried it in the past and again today. Unfortunately the dlls still end up as "DEFLATED" entries in the apk. I've also tried:

```
androidResources {  
    noCompress 'dll'  
}
```

with no success.

sp...@google.com <sp...@google.com> [#4](#)

Jun 3, 2022 03:00AM ⋮

Can you try the following:

```
android {  
    packagingOptions {  
        jniLibs {  
            useLegacyPackaging = false  
        }  
    }  
}
```

If that doesn't work, can you describe where the dll files are coming from?

This seems related to Issue 200272774

[Deleted User] <[Deleted User]> [#5](#)

Jun 3, 2022 04:10PM ⋮

Thanks for your suggestion, but this packagingOptions configuration doesn't help.

All jni libraries in my project (p. ex. "lib/arm64-v8a/libmonodroid.so") are not compressed even without adding any of the suggested additions. The Mono/Xamarin dlls are always compressed when using the "newApkCreator" (p. ex. "assemblies/Mono.Android.dll")

The dll files are meant to be loaded by Mono/Xamarin-Android (jni Runtime for running hybrid apps written in C#). The files are loaded via direct memory mapping (like jniLibs), that's why they need to stay uncompressed.

I've discussed this issue with the Xamarin-Android developers here: <https://github.com/xamarin/xamarin-android/issues/6838>

The related issue you linked seems to describe the same problem.

When `android.useNewApkCreator` option is dropped in AGP 8.0.0 there will be no way for me to upgrade my project.

Message last modified on Jun 3, 2022 04:14PM

sp...@google.com <sp...@google.com> [#6](#)

Jun 4, 2022 02:32AM ⋮

What is the full path of the dll files inside the APK?

sp...@google.com <sp...@google.com> [#7](#)

Jun 4, 2022 02:42AM ⋮

I'm not able to repro with the instructions [↪ here](#) because I can't install the Mono framework on my machine.

But I might be able to figure out the issue if you tell me where the dlls are during the build - i.e., do you see them in any of the library intermediate build folders?

ap...@gmail.com <ap...@gmail.com> [#8](#)

Jun 21, 2022 04:17PM ⋮

Comment has been deleted.

Message last modified on Jun 21, 2022 04:17PM

[Deleted User] <[Deleted User]> [#9](#)

Jun 21, 2022 04:17PM ⋮

Inside the apk the dlls are located at: `/assemblies/[dllName].dll`

When building the apk the .dlls are located at: `[libraryModuleName]/build/intermediates/java_res/[buildtype]/out/assemblies/[dllName].dll`

sp...@google.com <sp...@google.com> [#10](#)

Jun 22, 2022 02:50PM ⋮

Thanks. I'm able to reproduce locally now, and I'm working on a fix.

sp...@google.com <sp...@google.com> [#11](#)

Jun 22, 2022 02:52PM ⋮

ps - I don't think it's mentioned in this bug thread, but you mentioned [↪ here](#) that it's an issue only when `minifyEnabled` is true. That's correct, right?

[Deleted User] <[Deleted User]> [#12](#)

Jun 22, 2022 03:31PM ⋮

Thank you for your time!
Yes, it only happens when `minifyEnabled` is set to true. Sorry, I forgot to mention that in this bug thread.



sp...@google.com <sp...@google.com> [#13](#)

Jun 23, 2022 08:09AM ⋮

Marked as fixed.

This is fixed with `Change-Id: Ic50b383746519475f9a25878461b0c1164c95a03`.

The fix will be in AGP 7.4.0-alpha07.



[Deleted User] <[Deleted User]> [#14](#)

Jun 24, 2022 12:54AM ⋮

Is there any chance this makes it into 7.3 or a 7.2 patch?