Latest way to make Android app?

help

RobertN March 5, 2025, 5:35am 1

I see some old posts about Android, but the way things change I figure I should ask again. Are there any guides on how to build an Android app, specifically an app using winit and wgpu?

With Xcode/iOS I build a `staticlib` crate, and use Xcode to create minimal app that calls the Rust entry point (an `extern "C" fn`) from the Objective-C `main()`. Don't know if I should try to do a similar thing with Android or if there are `cargo` commands or other tools that will help.

jofas March 5, 2025, 6:48am 2

Have you checked out what the rust-mobile people are up to? They have ndk bindings and maintain cargo xbuild. I haven't tried them myself yet, but I'm rather excited to do so when I find the time.

2 Likes

RobertN March 5, 2025, 7:06am 3

jofas:

> Have you checked out what the rust-mobile people are up to?

Nope, but I will now. That looks promising.

RobertN March 6, 2025, 7:21pm 4

The following makes me hesitant to use `xbuild`. It suggests it hasn't worked out too well.

github.com/rust-mobile/cargo-apk

Revive cargo-apk!

opened 03:30PM - 12 Oct 24 UTC

thequver

Can anyone continue development of this project? The advertised xbuild is unintu…itive to use, [lacks documentation](https://github.com/rust-mobile/xbuild/issues/173) and seems to be dead from the start (the last release was in 2022). Cargo apk is way better for simple building android apps without extra bloat functionality like store publish.

When I tried `cargo apk` it gave me the kind of error message that makes want to quit using it immediately:

```
% cargo apk run -p my-rust-app
Using package `my-rust-app` in `/MonoRepo/my-rust-app/Cargo.toml`
Error: Invalid args.
```

I wonder if it's possible to use straight and simple `cargo` like I did with iOS. For that I built a static library:

```
cargo build -p my-rust-app --target aarch64-apple-ios
```

That I used that lib in an Xcode project. I don't mind using Xcode or Android Studio to create a simple shell app. Unless the rust command line tools are 100% done and well documented, I think those IDEs will make it easier to do the Apple/Android specific stuff for running on devices or creating store upload packages.

But when I try to do the same with Android target, I get a linker error:

```
cargo build -p my-rust-app --target aarch64-linux-android
... [building stuff] ....
error: linker `/Users/rob/Library/Android/sdk/ndk/27.2.12479018/toolchains/llvm/prebuilt/darwin-x86_64/bin/aarch64-linux-andro
```

That folder does have `clang` files in it that almost match.

```
ls /Users/[...]/darwin-x86_64/bin/aarch64-linux*
/Users/[...]/darwin-x86_64/bin/aarch64-linux-android21-clang
/Users/[...]/darwin-x86_64/bin/aarch64-linux-android21-clang++
...
/Users/[...]/darwin-x86_64/bin/aarch64-linux-android35-clang
/Users/[...]/darwin-x86_64/bin/aarch64-linux-android35-clang++
```

Maybe I need to set something more in my `Cargo.toml`? Maybe it's not possible to build for Android like this? If it is possible, I'd prefer this simpler path, so I can do the "run on device" step, or "build signed APK" step, from Android studio, at least for now.

If I *must* use `cargo-apk` or `xbuild` then I'll persist in trying to learn those tools.

[RobertN](#) March 6, 2025, 7:46pm 5

I may have found what I'm looking for here with [cargo-ndk](#).

```
cargo install cargo-ndk
cargo ndk -t arm64-v8a -o ./jniLibs build -p my-app
```

That created a `.so` native library, as desired. But it's going to take me some time to see how to call that from the shell app I create in Android Studio.

1 Like

[system](#) Closed June 4, 2025, 7:46pm 6

This topic was automatically closed 90 days after the last reply. We invite you to open a new topic if you have further questions or comments.

Related topics

Topic	Replies	Views	Activity
[Android Rust App help](#)	6	1057	November 9, 2021
[Build App with NDK help](#)	3	646	January 12, 2023
[Announcing cargo-apk announcements](#)	9	22131	February 24, 2020
[Rust mobile app developement community](#)	5	250	November 15, 2025
[Rust on Android Today? help](#)	14	13920	July 3, 2022