```
This guide is based on Android Studio 3.1.3 and NDK version r17b.
```

Compiling the Orx Library and the Android Demo Project

To follow this guide, you must first ensure you have completed the Getting Android Studio, NDK and Orx step.

> To decide on which Android port to use, see: Which Android Port should you use?

1. Compiling the Orx library for Android

```
cd <DEV_FOLDER>/orx/code/build/android
ndk-build
./install.sh

For windows, use install.bat instead.
```

ndk-build will compile for all four cpu architectures (arm64-v8a, arm64-v8a, x86 and x86_64) into android/obj/local.

Each folder will contain the three versions of the Orx static library: liborx.a, liborxd.a and liborxp.a (release, debug and profile).

There is an install.bat (for windows) and install.sh (for mac and linux) to copy the libraries into place in the orx project structure at: orx/code/lib/static

Holding the android versions of the Orx libraries at this location will make things more convenient when building your own projects.

A good guide to ndk-build can be found at: https://developer.android.com/ndk/guides/ndk-build

2. Compiling the Android Demo Native Code

Let's compile the native code portion of the Demo which links in the static Orx library that we prepared in step 1:

```
cd <DEV_FOLDER>/orx/code/demo/android/app/src/main
ndk-build
```

This step creates the native library for the demo.

Creating a native library is only half the story, and will provide only a shared library (.so). This library

and assets still need to be packaged for use on a device.

That's what Android Studio (and Gradle: it's build system) are used for.

3. Building the Android Demo in Android Studio

Now that we have the shared library of the demo, we need to build it in Android Studio. What really happens at the step?

Android Studio (using Gradle) will take the shared library and wrap some java code around it so that the application can be executed. Everything will be packaged up.

1. Open Android Studio
2. Load in the demo by selecting the orx/code/demo/android folder.
3. Gradle inside Android Studio will begin building automatically.
4. If there are no setup issues, the build will complete.
5. You can test the build on a real Android device or the emulator by selecting the Run / Run 'app' menu.
6. When running this step, an APK file be created and deployed to your device.

You can also create an APK at any time and copy this to your Android device as well.

General Troubleshooting

Tip: If you receive the following error: *[OrxTest] Unable to resolve target* '*android-19*'

You don't have the minimum required android SDK (API 19) that the orx Demo is set to. Use the SDK Manager (Tools/Android/SDK Manager) to check and download the latest SDK. In the build.gradle, you can also set the SDK you want to target. This is an excellent resource to help you choose.

Tip: If you only receive a debug build, and no release build, there is an excellent workaround detailed here:
http://stackoverflow.com/questions/27646262/how-to-create-a-release-android-library-package-aar-in-android-studio-not-deb

Tip: Android NDK: Could not find application project directory ! Android NDK: Please define the NDK_PROJECT_PATH variable to point to it.

If you get this when running ndk-build, then you are not in the correct folder. Check the instructions to ensure you are at the correct path. There should be a jni folder present. ndk-build operates on the contents of a jni folder.

If you have completed this tutorial and would like to try it with your own project, continue to Using the Android demo as a template for your own projects.

From:
https://orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://orx-project.org/wiki/en/tutorials/android/setup_android?rev=1597923586

Last update: **2025/09/30 17:26 (14 months ago)**