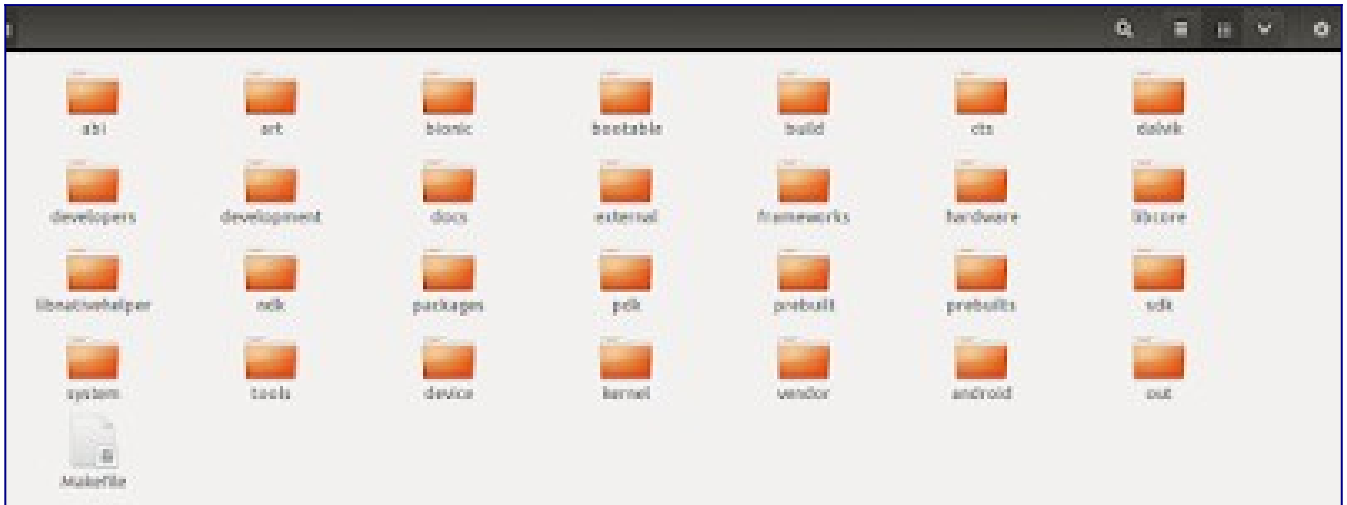

Content of AOSP Folder



Folder name in alphabetical order .

- 1. abi** – This folder contains a subfolder called **cpp** which actually contains many **C++** files linked to many places.
- 2. art** – it is the folder that deals with the compilation of the latest android ART run time.
- 3. bionic** – Bionic is mainly a port of the BSD C library to our Linux kernel with the following additions/changes:
 - **No support for locales.**
 - **No support for wide chars (i.e. multi-byte characters).**
 - **its own smallish implementation of pthreads based on Linux futexes.**
 - **Support for x86, ARM and ARM thumb CPU instruction sets and kernel interfaces.**
- 4. bootable** – Boot and startup related code. Some of it is legacy, the fastboot protocol info could be interesting since it is implemented by boot loaders in a number of devices such as the Nexus ones.
- 5. build** –The main entry point of the build system resides here – *envsetup.sh*.
First step before you do anything to *build Android* is to use the command
source build/envsetup.sh
This script will check few things to make sure all the needed application available in the local machine. It also setup the devices that can be built, which is extracted from the directory device.
- 6. cts** –the compatibility tests. The test suite to ensure that a build complies with the Android specification.
- 7. dalvik** –This is the folder responsible for the compilation of the Dalvik runtime for the Android devices.
Have a look at the difference between the two(art and dalvik) folders and you'll have an idea of how things work in that case.
- 8. Development** –This directory contains application that are not part of the deployed app in the OS. There are some useful application such as widget builder, etc.

9. Device –It contains the device specific configurations for many devices. It contains following subfolder ‘common’, ‘generic’, ‘Google’ and ‘Sample’.

common–This directory contains gps information and also a script that allows you to extract proprietary binary files from your phone to be part of the build process.(You can try to have a look at your device’s device tree and then the **cm.mk** file, where you could find relations of these files.

In my case, it shows like this:

```
# Include GSM stuff$(call inherit-product, vendor/cm/config/gsm.mk)-and-# Inherit some common cyanogenmod stuff.$(call inherit-product, device/common/gps/gps_eu_supl.mk)generic – This directory contains the generic device configuration that is called ‘goldfish’.This is the device classification used when building for the emulator.
```

Google– This directory contains the Android Accessories Kit code. It contains a demokit Android app that allows you to control the ADK board.

sample –This directory contains a full example of writing your own Android platform shared library, without changing the Android framework. It also shows how to write JNI code for incorporating native code into the library, and a client application that uses the library. This example is ONLY for people working with the open source platform to create a system image that will be delivered on a device which will include a custom library as shown here. It can not be used to create a third party shared library, which is not currently supported in Android.

10. external – This directory contains source code for *all external open source projects* such as *SQLite, Freetype, webkit and webview*.

11. frameworks –One of the most important directories. it contains the sources for the framework. Here you will find the implementation of key services such as the *System Server* with the **Package** and *Activity managers*.

A lot of the mapping between the java application APIs and the native libraries is also done here.

12. hardware –Hardware related source code such as the Android hardware abstraction layer (**HAL**) specification and implementation. This folder also contains the reference Radio Interface Layer(**RIL**– To communicate with the modem side) implementation.

13. Kernel –It’s not a default folder in the source code, but it’s a part of device configuration set-up. It contains the kernel source of your device.

14. libcore –I’ll explain this one with the important folders inside this, since every folder performs a different function.

dalvik– DalvikVM runtime for Android

dom– Java test classes for DOM

expectations– Contains information about the test cases

include– Some C/C++ include files that used for Array and String handling

json– JSON based Java implementation

luni– Contains test source code for loading .jar and .dex files

support –Contains support class file for testing Dalvik

xml –XML pull and push implementation

15. ndk -Contains build scripts and helper files for building the NDK

16. out– The build output will be placed here after you run make.

The folder structure is *out/target/product/*. In the default build for *the emulator the output* will be

placed in *out/target/product/generic*. This is where you will find the images used by the emulator to start (or to be downloaded and flashed to a device if you are building for a hardware target).

17. packages – Standard Android application that are available as part of the AOSP – Camera, SMS, Dialer, Launcher, etc

18. prebuilt – Contains files that are distributed in binary form for convenience. Examples include the cross compilations toolchains for different development machines.

19. sdk – This directory contains lots of apps that are not part of operating system. There are quite useful apps that developers can leverage on and can be enhanced further as part of the operating system.

20. system – Source code files for the core Android system. That is the minimal Linux system that is started before the Dalvik VM and any java based services are enabled. This includes the source code for the init process and the default init.rc script that provide the dynamic configuration of the platform.

21. tools – Some external important tools that help in compiling. Not sure.

22. vendor – This directory contains vendors specific libraries. Most of the proprietary binary libraries from non-open source projects are stored here when building AOSP.

Apart from above mention folder , there is a hidden *.repo* directory that contains the source for the repo utility. It also contain a manifest specifying what git repositories you want to track for this Android source project. If you have your own additions you could automatically track them by adding a local manifest here.

For modifications of the platform framework there are some instructions available in the device/sample folder of the source code tree. That will show you how to add APIs to Android without having to modify the core framework.