Setting up a Mac OS build environment

In a default installation, Mac OS runs on a case-preserving but case-insensitive filesystem. This type of filesystem is not supported by git and will cause some git commands (such as git status) to behave abnormally. Because of this, we recommend that you always work with the AOSP source files on a case-sensitive filesystem. This can be done fairly easily using a disk image, discussed below.

Once the proper filesystem is available, building the master branch in a modern Mac OS environment is very straightforward. Earlier branches, including ICS, require some additional tools and SDKs.

MacOS的文件系统是一个保留大小写，但不区分大小写的文件系统，这样的文件系统不支持git的部分指令，因此我们需要在MacOS上重新创建一个区分大小写的文件系统。

case-preserving 保留大小写

case-insensitive 不区分大小写(大小写不敏感)

case-sensitive 区分大小写(大小写敏感)

Creating a case-sensitive disk image

You can create a case-sensitive filesystem within your existing Mac OS environment using a disk image. To create the image, launch Disk Utility and select "New Image". A size of 25GB is the minimum to complete the build; larger numbers are more future-proof. Using sparse images saves space while allowing to grow later as the need arises. Be sure to select "case sensitive, journaled" as the volume format.

在MacOS下，我们可以通过创建磁盘镜像(disk image, dmg)来创建这个区分大小写的文件系统。

需要确保在创建磁盘镜像时，选择 ‘Case-sensitive Journaled’格式。

安卓源码编译最少需要25G的磁盘空间，因此我们创建一个40G的磁盘镜像

hdiutil命令，用于创建磁盘镜像，我们常用命令有

hdiutil create，attach，detach，resize

You can also create it from a shell with the following command:

# hdiutil create -type SPARSE -fs 'Case-sensitive Journaled HFS+' -size 40g ~/android.dmg

创建的dmg文件可能会变成android.dmg.sparseimage，这是正常的情况。

This will create a .dmg (or possibly a .dmg.sparseimage) file which, once mounted, acts as a drive with the required formatting for Android development.

If you need a larger volume later, you can also resize the sparse image with the following command:

# hdiutil resize -size <new-size-you-want>g ~/android.dmg.sparseimage

For a disk image named android.dmg stored in your home directory, you can add helper functions to your ~/.bash\_profile:

下面的例子是编写了两个函数，用来一键挂载和卸载磁盘镜像

• To mount the image when you execute mountAndroid:

# mount the android file image  
function mountAndroid { hdiutil attach ~/android.dmg -mountpoint /Volumes/android; }

**Note:** If your system created a .dmg.sparseimage file, replace ~/android.dmg with ~/android.dmg.sparseimage.

• To unmount it when you execute umountAndroid:

# unmount the android file image  
function umountAndroid() { hdiutil detach /Volumes/android; }

Once you've mounted the android volume, you'll do all your work there. You can eject it (unmount it) just like you would with an external drive.

挂载磁盘镜像后，我们就可以直接在这个目录下进行所有编译操作了

Optimizing a build environment (optional)

Setting up ccache

You can optionally tell the build to use the **ccache compilation tool,** which is a compiler cache for C and C++ that can help make builds faster. It is especially useful for build servers and other high-volume production environments. Ccache acts as a compiler cache that can be used to speed up rebuilds. This works very well if you use make clean often, or if you frequently switch between different build products.

使用ccache可以加快编译速度，尤其是在编译大型工程或者需要频繁make，clean时。

**Note:** If you're instead conducting incremental builds (such as an individual developer rather than a build server), ccache may slow your builds down by making you pay for cache misses.

To use ccache, issue these commands in the root of the source tree:

$ export USE\_CCACHE=1

$ export CCACHE\_DIR=/<path\_of\_your\_choice>/.ccache

$ prebuilts/misc/linux-x86/ccache/ccache -M 50G

The suggested cache size is 50-100G.

Put the following in your .bashrc (or equivalent):

$ export USE\_CCACHE=1

By default the cache will be stored in ~/.ccache. If your home directory is on NFS or some other non-local filesystem, you will want to specify the directory in your .bashrc file too.

On Mac OS, you should replace linux-x86 with darwin-x86:

$ prebuilts/misc/darwin-x86/ccache/ccache -M 50G

When building Ice Cream Sandwich (4.0.x) or older, ccache is in a different location:

$ prebuilt/linux-x86/ccache/ccache -M 50G

This setting is stored in the CCACHE\_DIR and is persistent.

On Linux, you can watch ccache being used by doing the following:

$ watch -n1 -d prebuilts/misc/linux-x86/ccache/ccache -s