

# Java 虚拟机学习笔记（一） macOS 10.12 编译 JDK9

作者: [Qixian](#)

原文链接: <https://ld246.com/article/1505751135178>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

为了深入了解Java的运行机制，了解久负盛名的JVM，笔者决定开始学习JVM相关知识。

参考书为 [《深入理解Java虚拟机\(第2版\)》](https://book.douban.com/subject/24722612/) (以下简称为《深入》)

## 零、为什么是JDK9?

《深入》中，作者使用的是JDK6来编译JDK7，环境使用的是OSX 10.8，对Xcode的要求为4.5.2及以上。笔者从JDK7尝试到JDK8，皆因为Xcode版本的原因导致编译失败。简而言之就是：虽然JDK源码称要求4.5及以上，但我的Xcode9不能被调用，同时安装的Xcode4在10.12上也不能运行。直到换了JDK9，configure,build都没再报什么令人头炸的错，顺利完成。

## 一、获取JDK9

### 1.安装Homebrew

macOS一直没有自己官方的软件包管理器，所以要安装Homebrew来填补这个空缺。

安装方式<https://brew.sh/>

之后运行**brew -v** 能查看到版本信息即安装完成。

### 2.安装Mercurial

Mercurial是一个类似于Git的版本控制工具。OpenJDK的源码便是存放于Mercurial的仓库中。

安装命令为**brew install mercurial**

之后运行**hg --version**验证安装。

### 3.获取JDK9源代码

Repository的地址为<http://hg.openjdk.java.net>，在里面我们可以看到从JDK6到JDK10的JDK源码有一堆其他的东西。

获取过程：

```
hg clone http://hg.openjdk.java.net/jdk9/jdk9
cd jdk9
chmod +x get_source.sh
./get_source.sh
```

因网络下载能力相异，所以这一步有可能需要较长时间的等待。

## 二、构建 Build

在<common/doc/building.html>中详细介绍了build相关的环境要求、参数设置、可能的问题及解决方案和对整个编译系统的介绍。文档也有Markdown格式的，在同一目录下。

简单摘录一下在 MacOS下的环境需求。

- Apple Xcode 6.3 (using clang 6.1.0)
- JDK 8 on macOS (as Boot JDK)

- FreeType (at least version 2.3)

其他再因为机器相异而导致的环境缺失，在运行configure的结果中应该能够得出缺少什么，再去请搜索引擎，应该就能解决。

## 1.配置 Configure

幸运的是，作为10.12和JDK9的受益者，笔者似乎并没有进行什么过多的配置，并没有像《深入》中译JDK7所需要写的长长的环境变量设置脚本。

那么多的参数，笔者只用到了一个`--disable-warnings-as-errors`以去除一些烦人且不知如何修改的Error。

运行`configure --disable-warnings-as-errors`之后，configure会打印出一个详细的配置清单。如果问题的话，此时，在`./build`文件夹中应该出现了一个代表这次配置的文件夹。这个文件夹也是此后编译结果的默认所在之处。

## 2.编译 make

很简单一条指令`make`。

接下来，便是疯狂的编译过程，终端窗口会被一堆杂乱的代码不断刷屏，时不时还会有个WARN蹦出吓唬人。笔者的电脑是i7-6700K，开8核编译，并没有花多长时间便编译完成。

编译结果位于`build / macosx-x86_64-normal-server-release / 下`。

可以通过`jdk9/build/macosx-x86_64-normal-server-release/jdk/bin/下的./java -version`命令查编译结果的版本信息。

笔者的如下：

```
$ ./java -version
openjdk version "9-internal"
OpenJDK Runtime Environment (build 9-internal+0-adhoc.kurosei.jdk9)
OpenJDK 64-Bit Server VM (build 9-internal+0-adhoc.kurosei.jdk9, mixed mode)
```

如果上述信息正确显示（具体细节可能有所不同），则应该可以证明JDK9的编译是成功的了，接下来可以继续学习关于JVM的知识了。