

# 使用Python语言实现 Appium自动化测试

APPIUM WITH PYTHON

DAY01

# 内容

上午	09:00 ~ 09:30	Appium概述
	09:30 ~ 10:20	
	10:30 ~ 11:20	Appium环境搭建1
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	Appium环境搭建1
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



# Appium概述

Appium概述

Appium概述

Appium概述

常见测试工具对比

Appium的优缺点

Appium特色

Appium的设计

Appium网站资源

Appium工作原理

Appium服务器原理

# Appium概述



# Appium概述

- Appium是一个开源的、免费的、跨平台、跨语言的移动端的自动化测试框架。
  - 支持多种移动操作系统平台：iOS、Android、FirefoxOS
  - 支持多种PC操作系统：Linux、Windows、Mac
  - 支持多种测试脚本语言：Java、Object-C、Python、Ruby、JavaScript、PHP、C#或Perl



Automation for Apps

# 常见移动测试工具对比

- Appium在跨平台、跨语言方面较为出众

	Robotium	uiautomator	Espresso	Appium	Calabash
Android	Yes	Yes	Yes	Yes	Yes
iOS	No	No	No	Yes	Yes
Mobile web	Yes (Android)	Limited to x.y clicks	No	Yes (Android & iOS)	Yes (Android)
Scripting Language	Java	Java	Java	Almost any	Ruby
Test creation tools	Testdroid Recorder	UI Automator viewer	Hierarchy Viewer	Appium.app	CLI
Supported API levels	All	16 =>	8, 10, 15-19	All	All
Community	Contributors	Google	Google	Active	Pretty quiet



# Appium的优缺点

- 优点：
  - 支持Android、IOS、 Firefox OS平台；
  - 支持Mac OS X、 Windows、 Linux系统；开源、免费（核心驱动不开源）
  - 支持Selenium WebDriver支持的所有语言，包括Java、Object-C、 Python、 Ruby、 JavaScript、 PHP、 C#或Perl语言
  - Appium在不同平台使用了标准的自动化API，跨平台时，无需重新编译和修改自己的应用
  - Appium支持任何一种测试框架
  - Appium实现了真正意义上的自动化测试



# Appium的优缺点（续1）

- 缺点
  - 环境搭建较为复杂
  - 获取的对象属性不全面
  - Android SDK API版本不低于16（Android系统不低于4.2版本）





# Appium特色

- 不需要为了自动化而且重新编译或修改测试app；
- 不限定语言：
  - 不应该让移动端自动化测试限定在某种语言和某个具体的框架；也就是说任何人都可以使用自己最熟悉最顺手的语言以及框架来做移动端自动化测试；
- 复用WebDriver API：
  - 一个自动化框架不要为了移动端的自动化测试而重新发明轮子，重新写一套惊天动地的api；也就是说webdriver协议里的api已经够好了，拿来改进一下就可以了；
- 一个自动化框架需要开源，在精神和实践上实现开源。



# Appium的设计

- 采用底层驱动商提供的自动化框架
  - IOS:苹果的UIAutomation
  - Android 4.2+ : 谷歌UiAutomator
  - Android 2.3到4.2 : 谷歌Instrumentation
- 采用底层驱动商提供统一API , 就是WebDriverAPI
  - WebDriver(也称Selenium WebDriver) 是一个C/S架构的协议, 叫做JSON Wire Protocol
  - 通过这个协议, 用任何语言写成的客户端都可以发送HTTP请求给服务器
- WebDriver是一个非常强大的网页协议, 官方正在起草W3C的标准



# Appium网站资源

- Appium官方网站  
– <http://appium.io/>



# Appium网站资源（续1）

- Appium开源社区
  - <https://github.com/appium/appium>

GitHub, Inc. [US] <https://github.com/appium/appium>

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

appium / appium Watch 613 Star 3,843 Fork 2,225

<> Code Issues 733 Pull requests 4 Pulse Graphs

Automation for iOS and Android Apps. <http://appium.io/>

5,966 commits 42 branches 109 releases 151 contributors

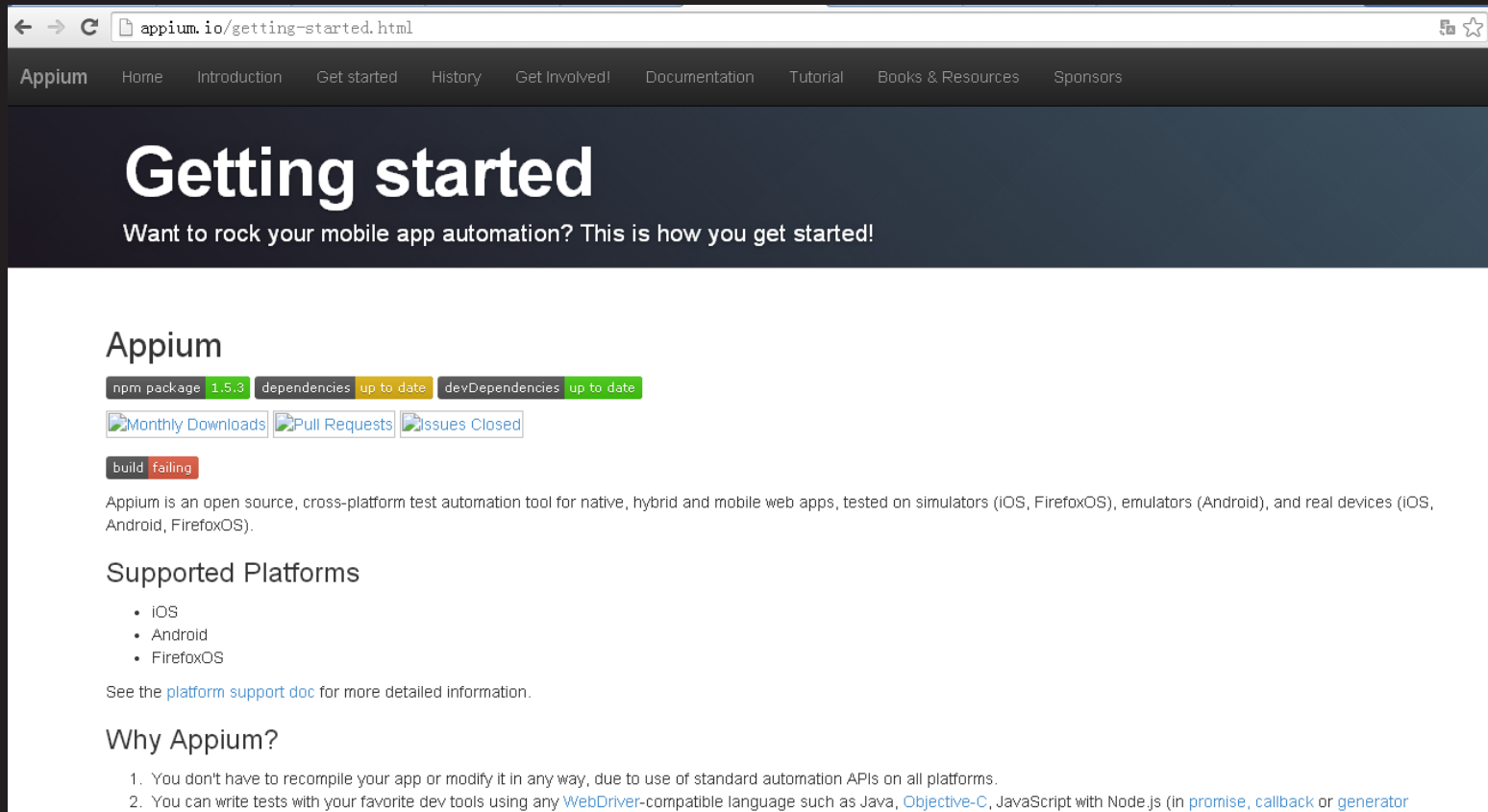
Branch: master New pull request Find file Clone or download

Commit	Message	Time
imurchie	Merge pull request #6671 from jlipps/jlipps-colors	Latest commit 068c4e5 6 days ago
.github	updated issue template asking users to give optional source code as w...	5 months ago
bin	ios-webkit-debug-proxy-launcher fix: replace underscore with lodash	3 months ago
docs	Add support to set directory on device to save screenshots for android	12 days ago
lib	add option for calling library to have appium throw instead of proces...	6 days ago
test	add ability for importing package to send in logHandler which can spy...	6 days ago
.gitignore	Fix python swipe docs	5 months ago
.jscsrc	set up babel/gulp code and update deps	a year ago
.jshintignore	adds Android api 18 to new ci	a year ago



# Appium网站资源（续2）

- Appium学习方式
  - <http://appium.io/getting-started.html>



← → ↻ appium.io/getting-started.html

Appium Home Introduction Get started History Get Involved! Documentation Tutorial Books & Resources Sponsors

## Getting started

Want to rock your mobile app automation? This is how you get started!

### Appium

npm package **1.5.3** dependencies **up to date** devDependencies **up to date**

Monthly Downloads Pull Requests Issues Closed

build **failing**

Appium is an open source, cross-platform test automation tool for native, hybrid and mobile web apps, tested on simulators (iOS, FirefoxOS), emulators (Android), and real devices (iOS, Android, FirefoxOS).

### Supported Platforms

- iOS
- Android
- FirefoxOS

See the [platform support doc](#) for more detailed information.

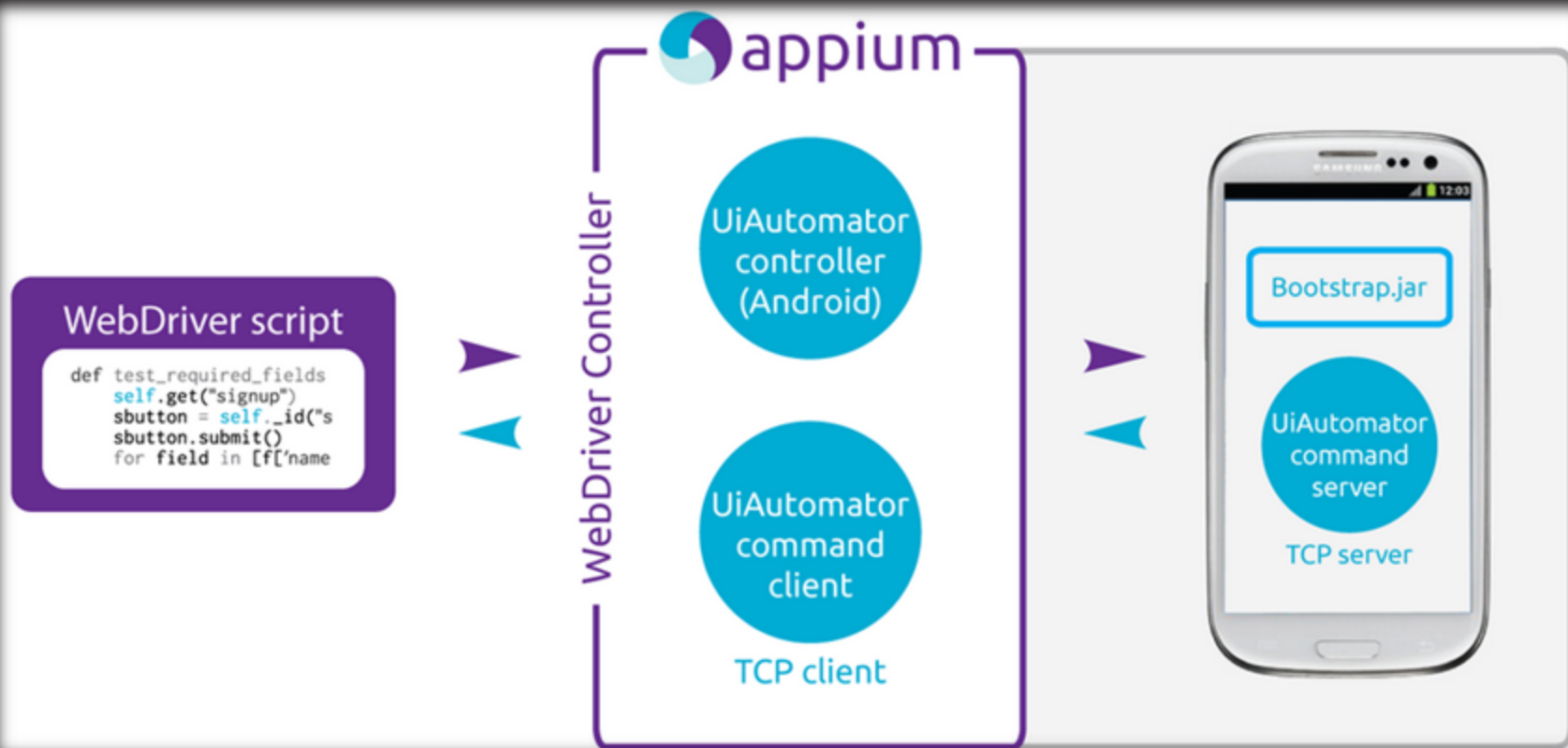
### Why Appium?

- You don't have to recompile your app or modify it in any way, due to use of standard automation APIs on all platforms.
- You can write tests with your favorite dev tools using any [WebDriver-compatible](#) language such as Java, [Objective-C](#), JavaScript with Node.js (in [promise](#), [callback](#) or [generator](#)



# Appium工作原理

- 客户端由脚本 <-> Appium服务端 <-> 手机设备



# Appium服务器原理

- Appium的核心是一个遵守REST设计风格的web 服务器
  - Appium server使用node.js写的http服务器，遵守REST风格
  - Appium服务器接受客户端的连接，接收客户端的命令，在手机设备上执行命令，之后通过HTTP的响应收集命令执行的结果

REST（英文：**Representational State Transfer**，简称**REST**）描述了一个架构样式的网络系统。

REST 原则是，客户端和服务端之间的交互在请求之间是无状态的。

从客户端到服务器的每个请求都必须包含理解请求所必需的信息。如果服务器在请求之间的任何时间点重启，客户端不会得到通知。此外，无状态请求可以由任何可用服务器回答，这十分适合云计算之类的环境。



# Appium服务器原理（续1）

- Session技术
  - 自动化的过程在session上下文中执行
  - 客户端初始化一个session会话，虽然不同的语言初始化的方式不同，但是他们都要发送POST/session 请求到服务器端，这些请求里面都会带有一个对象：desired capabilities,这个时候服务器端会启动自动化session然后返回一个session ID，以后的命令都会用这个session ID去匹配

Session:在计算机中，尤其是在网络应用中，称为“会话控制”。Session 对象存储特定用户会话所需的属性及配置信息。





# Appium服务器原理（续2）

- Desired Capabilities在启动session的时候是必须提供的。
- Desired Capabilities本质上是key-value的对象，它告诉appium server各种各样的信息：
  - 本次测试是启动浏览器还是启动移动设备？
  - 是启动android还是启动ios？
  - 启动android时，app的package是什么？
  - 启动android时，app的activity是什么？



# Appium环境搭建1

Appium环境搭建  
1

Appium环境搭建1

Appium安装总体步骤

安装JDK

配置JDK环境变量

检查Java环境

下载SDK

配置SDK环境变量

下载Android平台和工具

配置平台和工具环境变量

检查Android环境

# Appium环境搭建1



# Appium安装总体步骤

- 安装JDK并设置环境变量
- 安装SDK并设置环境变量
- 安装Appium Desktop
- 安装Appium Python Client
- 安装Genymotion模拟器
- 安装Pycharm配置工程



# 安装JDK

- 步骤1、下载并安装JDK
  - 版本建议1.7以上
- 步骤2、配置系统环境变量
- 步骤3、检查java环境



# 安装JDK ( 续1 )

- 步骤1、下载并安装JDK
- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

## Java SE 8u181

Java SE 8u181 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.  
[Learn more](#) ▶

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Licensing Information User Manual](#)
  - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme Files](#)
  - [JDK ReadMe](#)
  - [JRE ReadMe](#)

### JDK

[DOWNLOAD](#)

### Server JRE

[DOWNLOAD](#)

### JRE

[DOWNLOAD](#)



# 安装JDK (续2)

- 下载后，根据操作系统位数，安装相应的jdk，按照安装向导，一步一步进行安装，不需要单独配置任何信息。



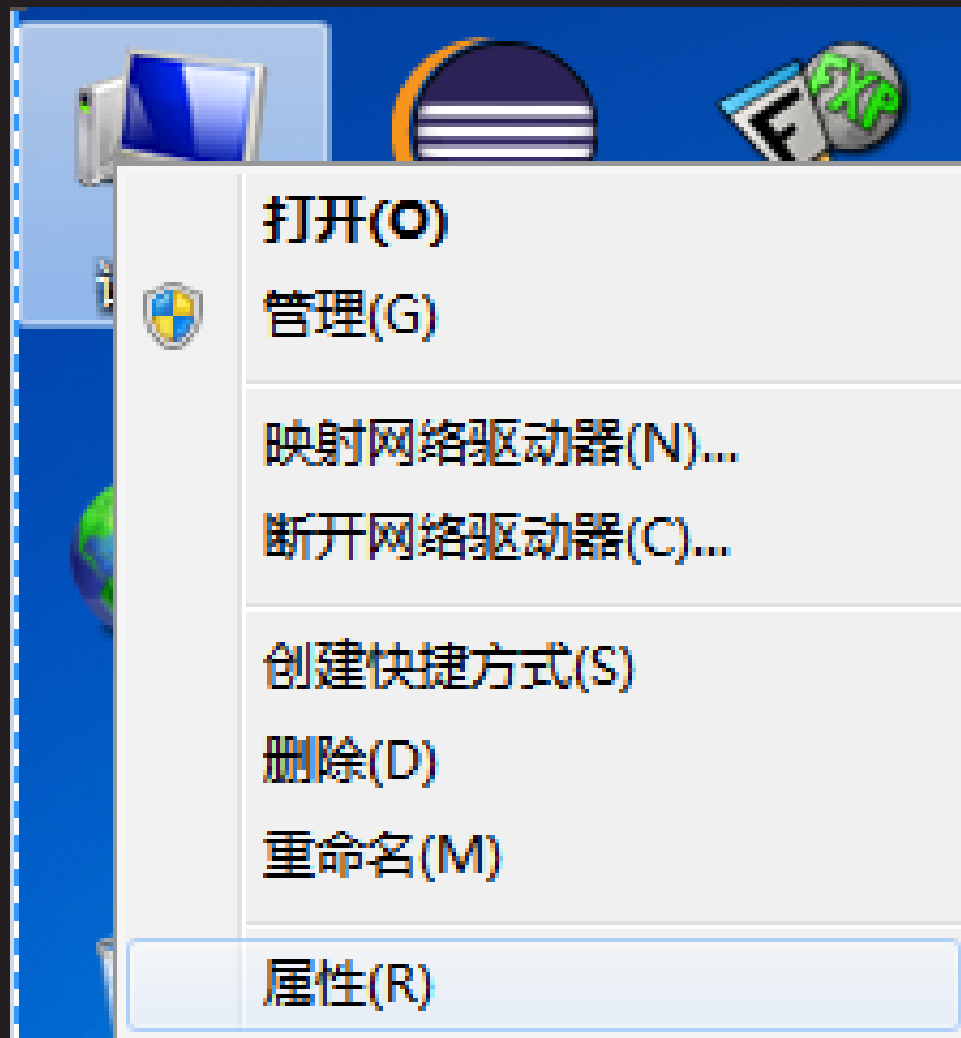
jdk-8u111-windows-i586



jdk-8u111-windows-x64

# 配置JDK环境变量

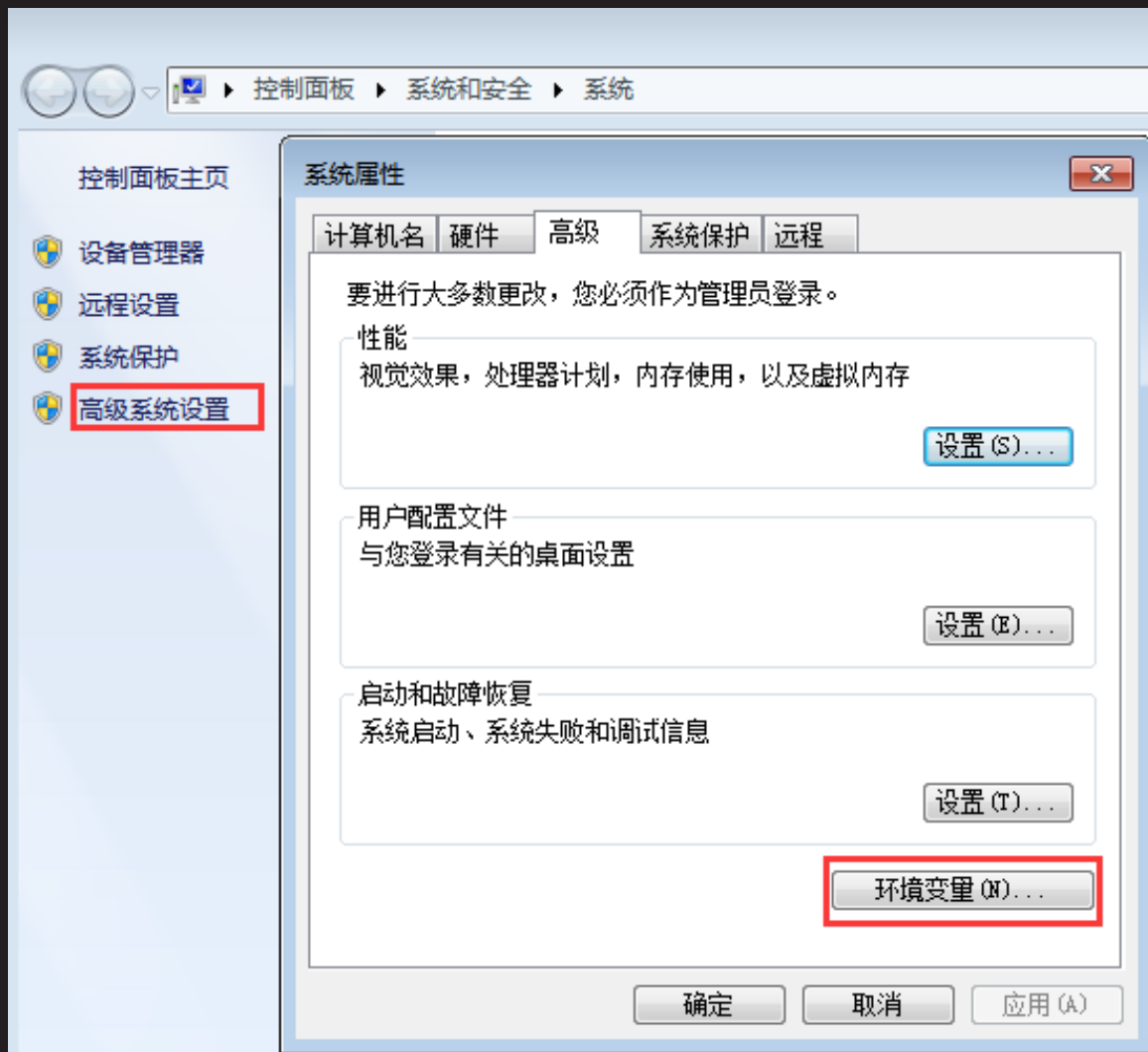
- 步骤2、配置环境变量：计算机上点击右键，选择属性





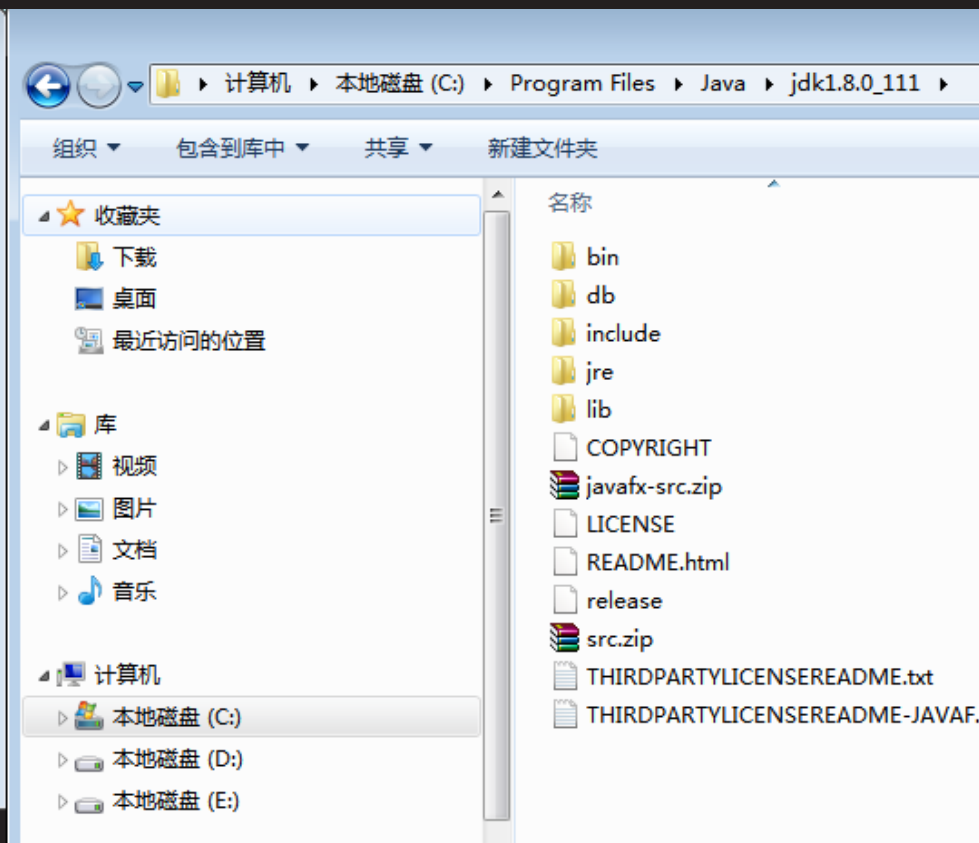
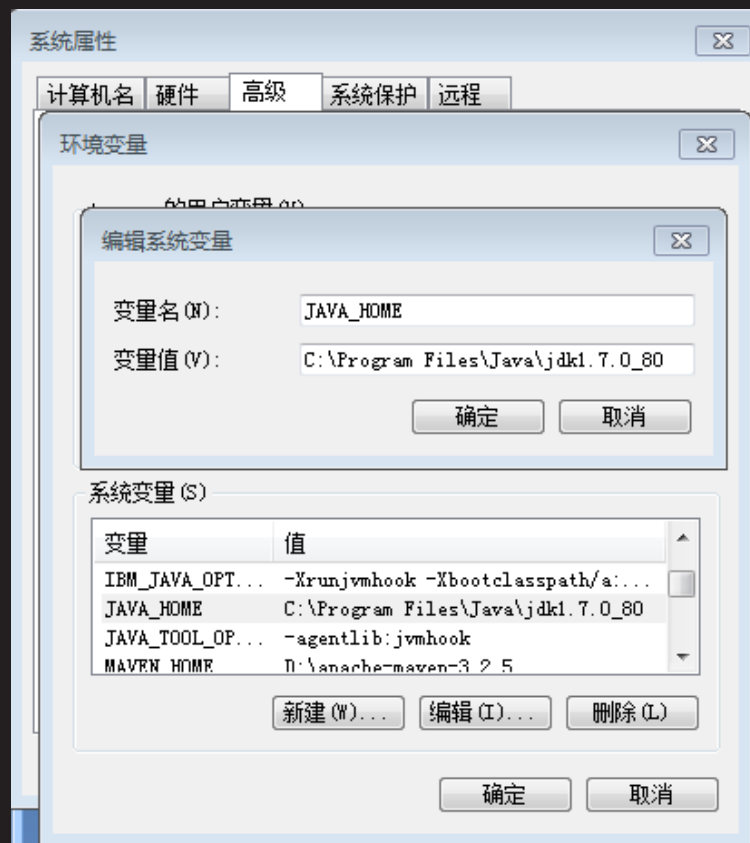
# 配置JDK环境变量（续1）

- 配置环境变量：点击“高级系统设置”，点击环境变量



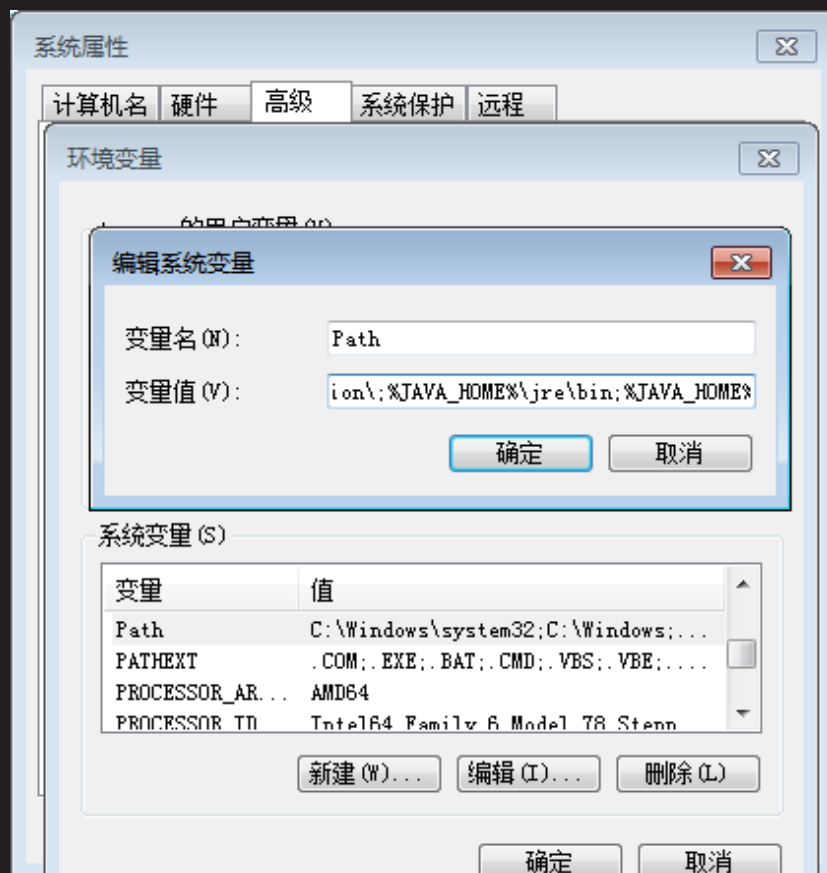
# 配置JDK环境变量（续2）

- 在系统变量中点击“新建”，输入变量名JAVA\_HOME，变量值中输入jdk安装路径，如下图：



# 配置JDK环境变量（续3）

- 在系统变量中找到Path，点击“编辑”，在末尾加入
  - %JAVA\_HOME%\jre\bin;
  - %JAVA\_HOME%\bin;



# 检查Java环境

- 步骤3、检查Java环境：
  - 打开命令行窗口
  - 输入java -version查看java版本



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7600]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\tarena>java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.111-b14, mixed mode)

C:\Users\tarena>
```



# 下载SDK

- 软件开发工具包（缩写：SDK、外语全称：Software Development Kit）一般都是一些软件工程师为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件时的开发工具的集合。Android SDK 指的是Android 专属的软件开发工具包。
- 安装部署SDK的步骤如下：
  - 步骤1、下载并解压缩
  - 步骤2、配置系统环境变量ANDROID\_HOME
  - 步骤3、下载Android工具
  - 步骤4、配置系统环境变量Path
  - 步骤5、检查Android环境



# 下载SDK ( 续1 )

- 步骤1、SDK下载地址：<http://tools.android-studio.org/index.php/sdk/>
- 直接下载zip版本，解压至任意目录即可



# 下载SDK (续2)

- 下载并安装SDK

## Android Studio

ANDROID

TOOLS

JDK

SDK

ADT

GRADLE

镜像

+ 版本

+ 开发社区

+ 系统平台

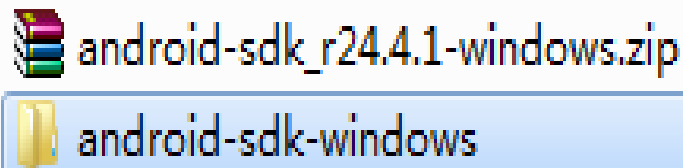
### ANDROID SDK R24.4.1-里程碑版本

Platform	Package	Size	SHA-1 Checksum
Windows	<a href="#">installer_r24.4.1-windows.exe</a> (Recommended)	151659917 bytes	f9b59d72413649d31e63320
	<a href="#">android-sdk_r24.4.1-windows.zip</a>	199701062 bytes	66b6a6433053c152b22bf8d
Mac OS X	<a href="#">android-sdk_r24.4.1-macosx.zip</a>	102781947 bytes	85a9cccb0b1f9e6f1f61633
Linux	<a href="#">android-sdk_r24.4.1-linux.tgz</a>	326412652 bytes	725bb360f0f7d04eaccff5a



# 下载SDK ( 续3 )

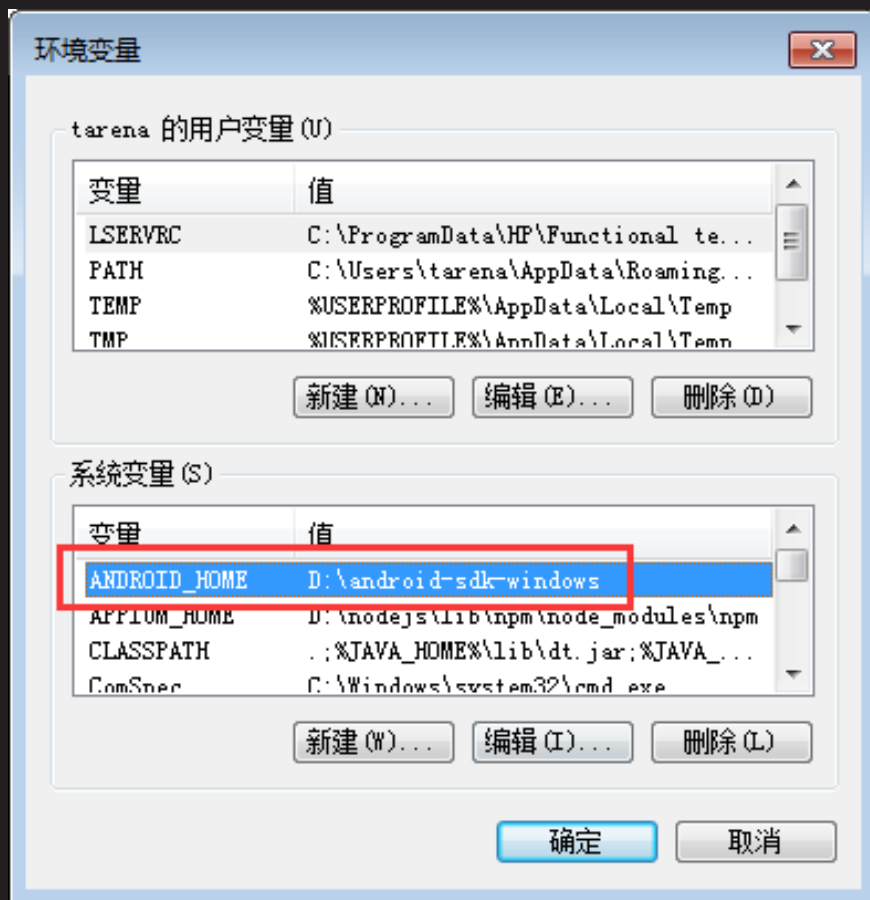
- 下载Android SDK后，复制到指定路径（例如：D:\），解压缩到当前文件夹，就存在了sdk的文件夹（D:\android-sdk-windows）





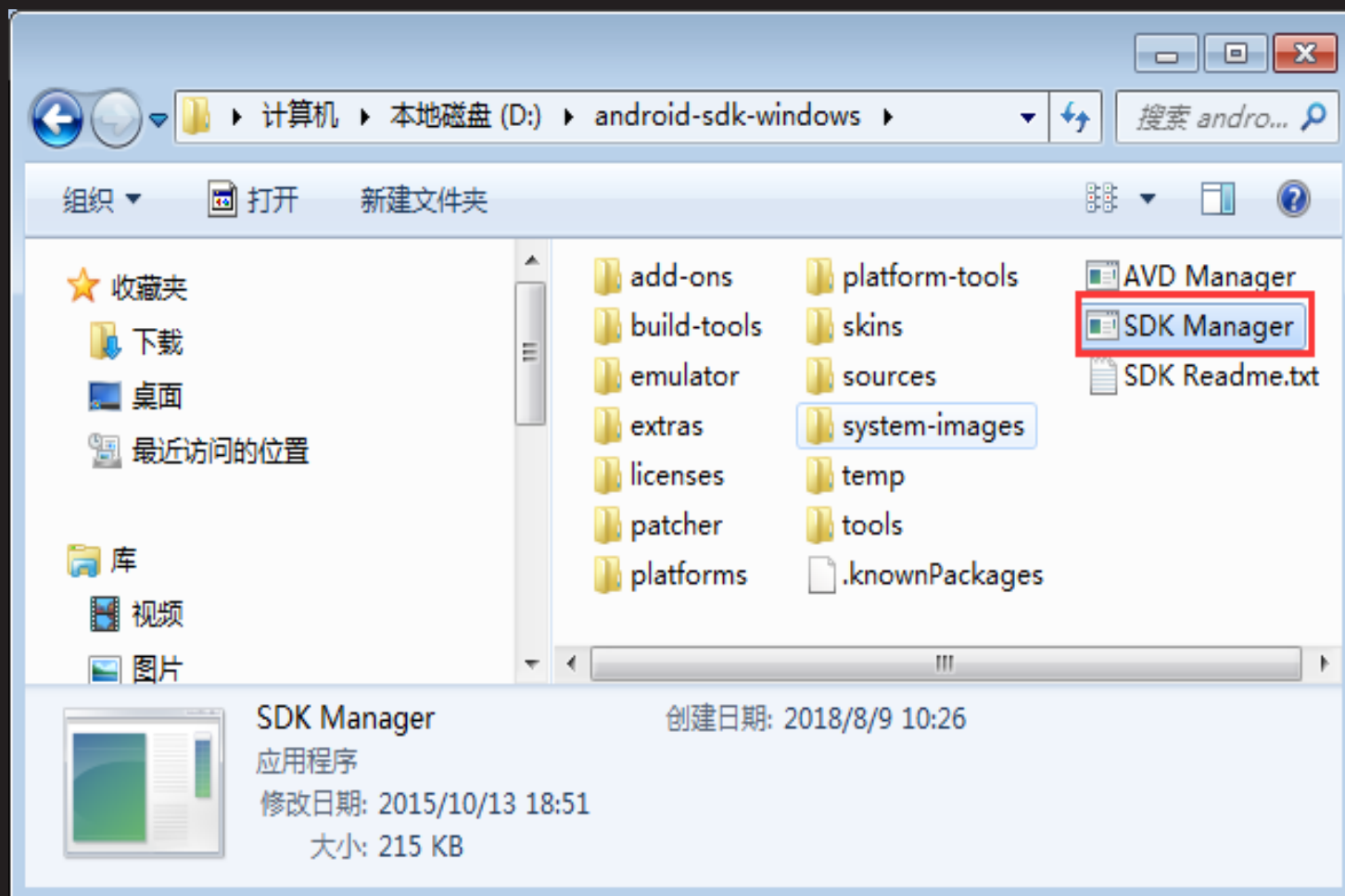
# 配置SDK环境变量

- 步骤2、配置环境变量：在系统变量中点击“新建”，输入变量名ANDROID\_HOME，变量值中输入SDK下的sdk文件夹路径，点击“确定”，如下图：



# 下载Android平台和工具

- 步骤3、进入Android SDK文件夹下，双击SDK Manager，选装工具。



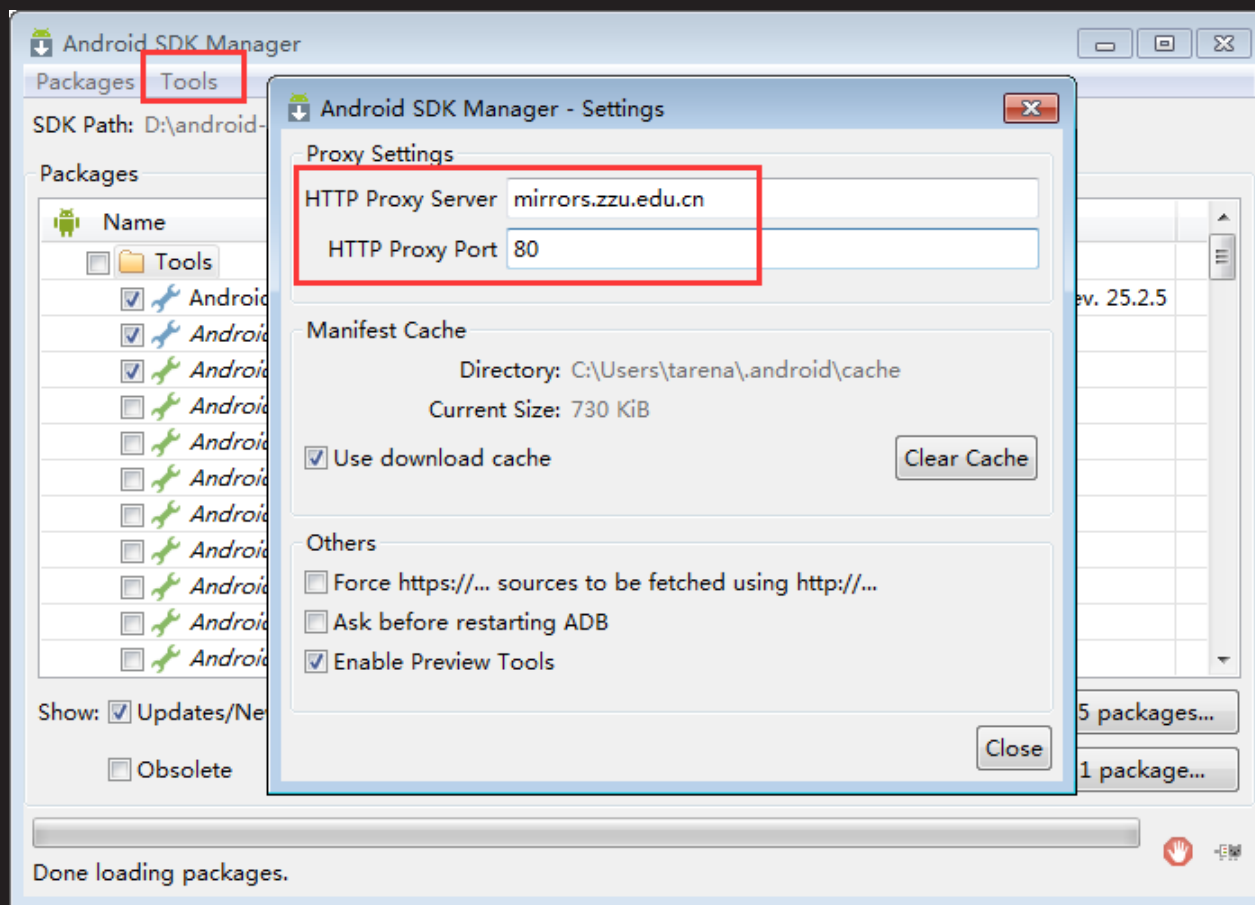
# 下载Android平台和工具（续1）

- SDK比较重要的目录包括【build-tools】【platforms】【platform-tools】【tools】
  - 【build-tools】里面是不同版本(例如21.1.1)的build工具，这些工具包括了aapt打包工具、dx.bat、aidl.exe等等
  - 【platform】是存放不同API-level版本SDK目录的地方
  - 【platform-tools】是一些android平台相关的工具，adb、fastboot等
  - 【tools】是指的安卓开发相关的工具，例如android.bat、ddms.bat(Dalvik debug Monitor Service)、draw9patch.bat等等



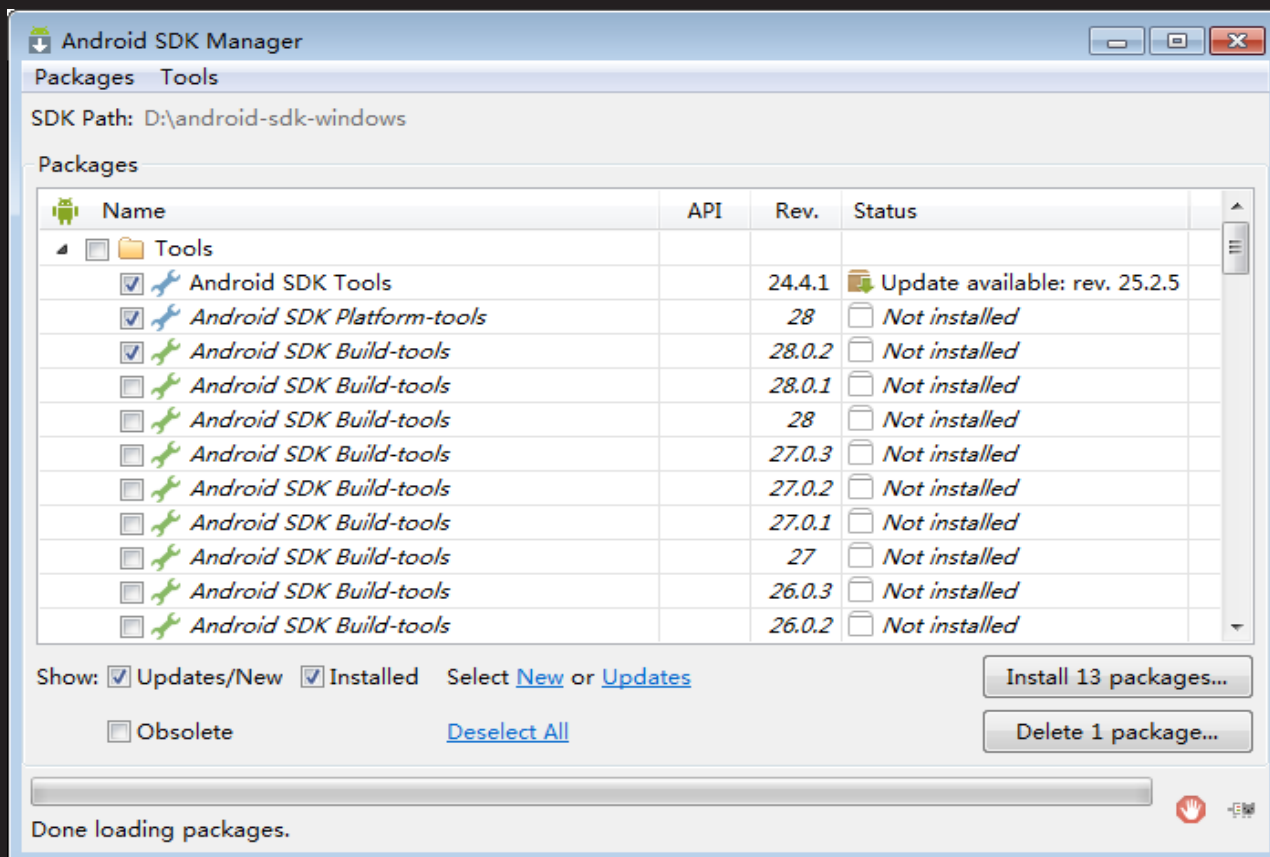
# 下载Android平台和工具（续2）

- 必装：Tools=>Options，设置代理提升速度，我们使用郑州大学开源镜像站mirrors.zzu.edu.cn，端口80

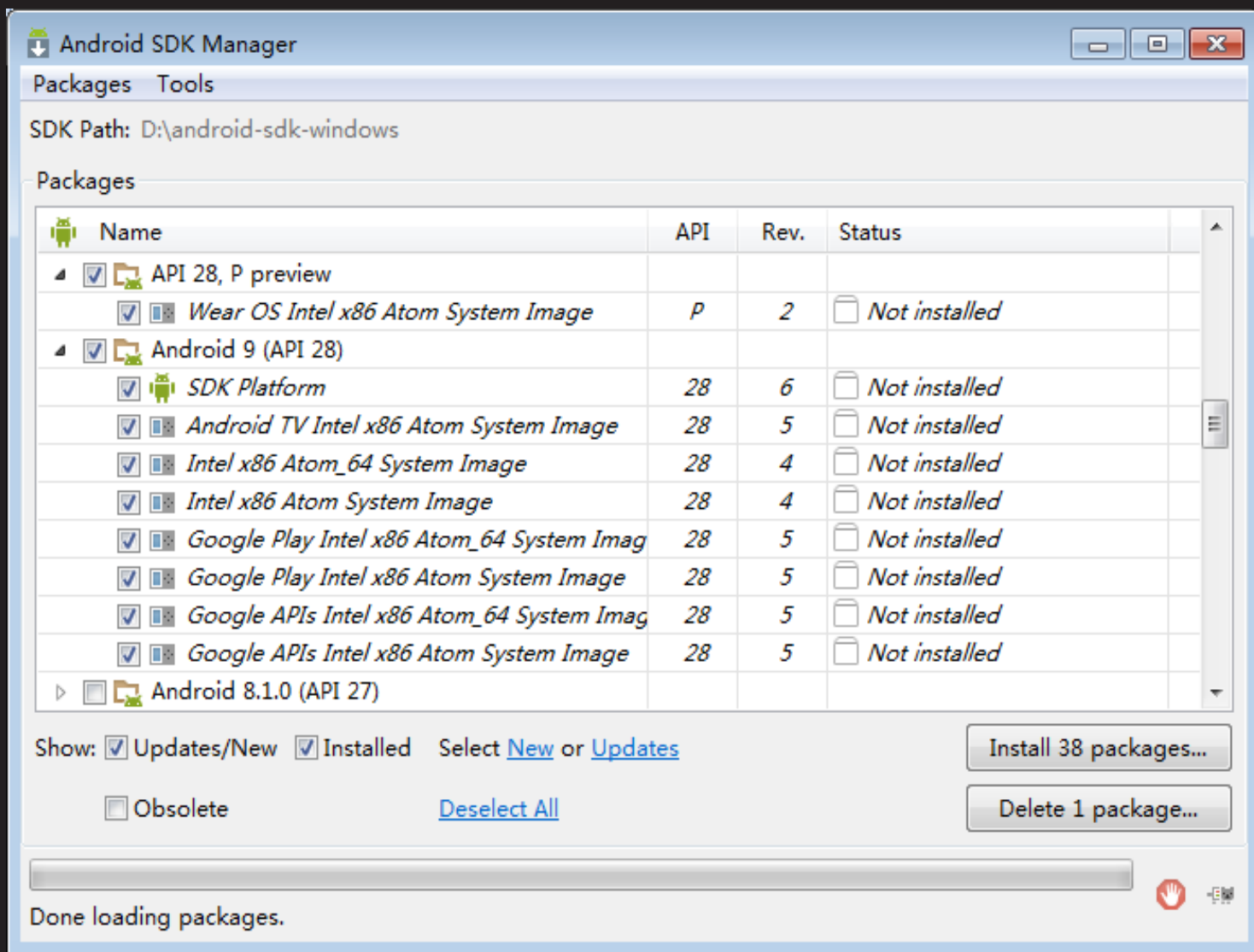


# 下载Android平台和工具（续3）

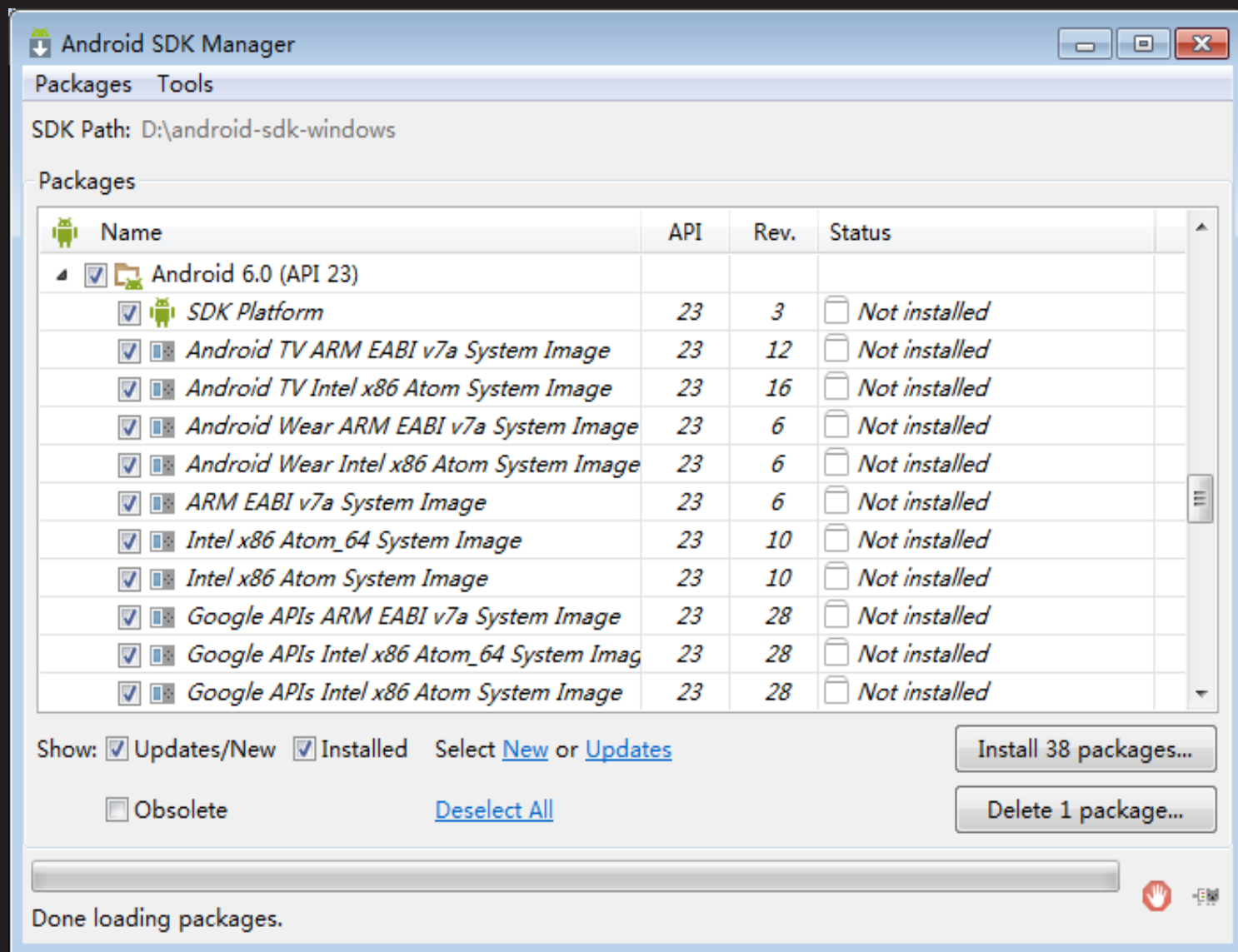
- 必装：Tools文件夹下Android SDK Platform-tools和Android SDK Build-tools，Android SDK Tools可以选择是否更新，建议网速如果支持下载直接更新。



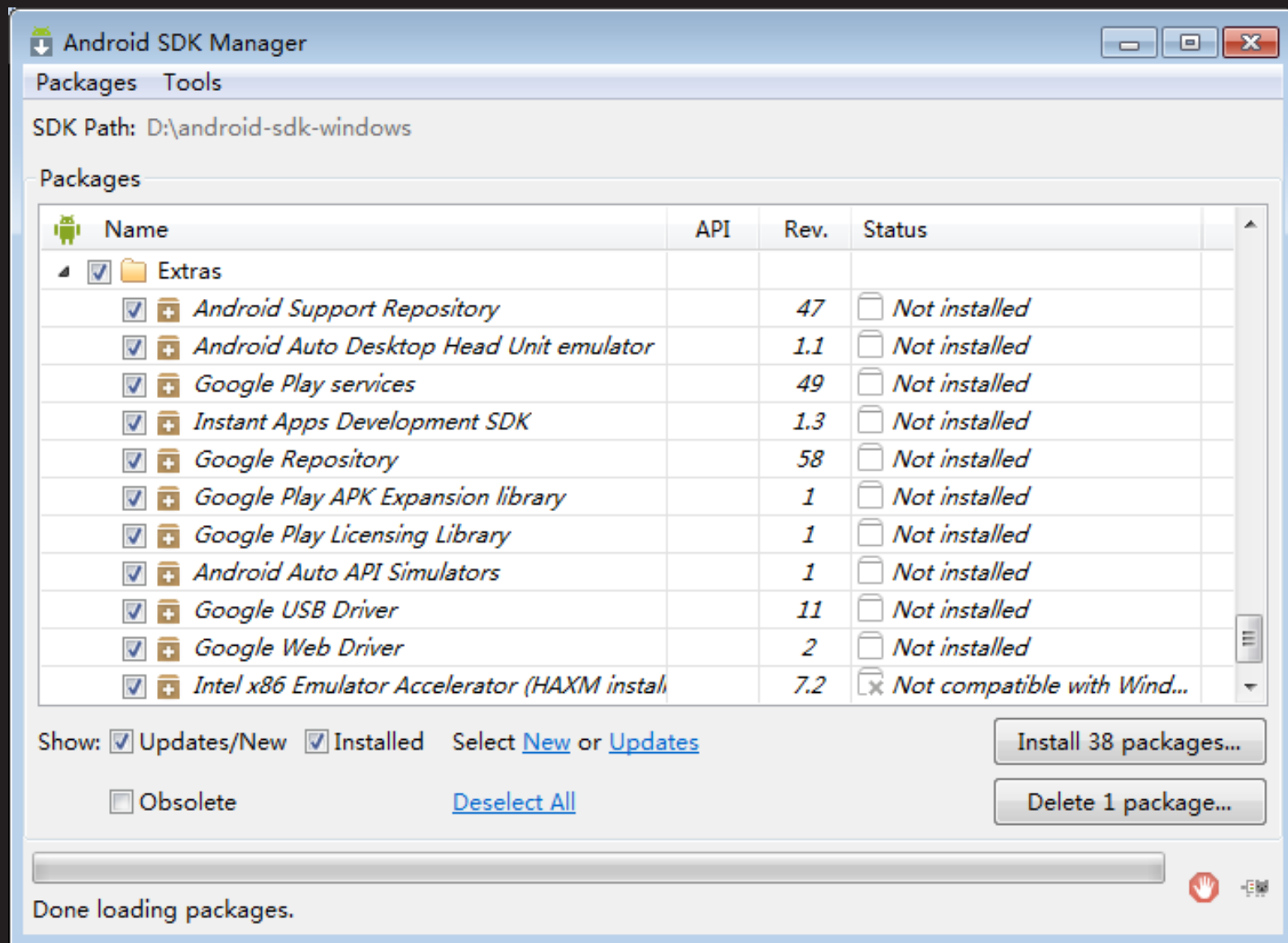
# 下载Android平台和工具（续4）



# 下载Android平台和工具（续5）

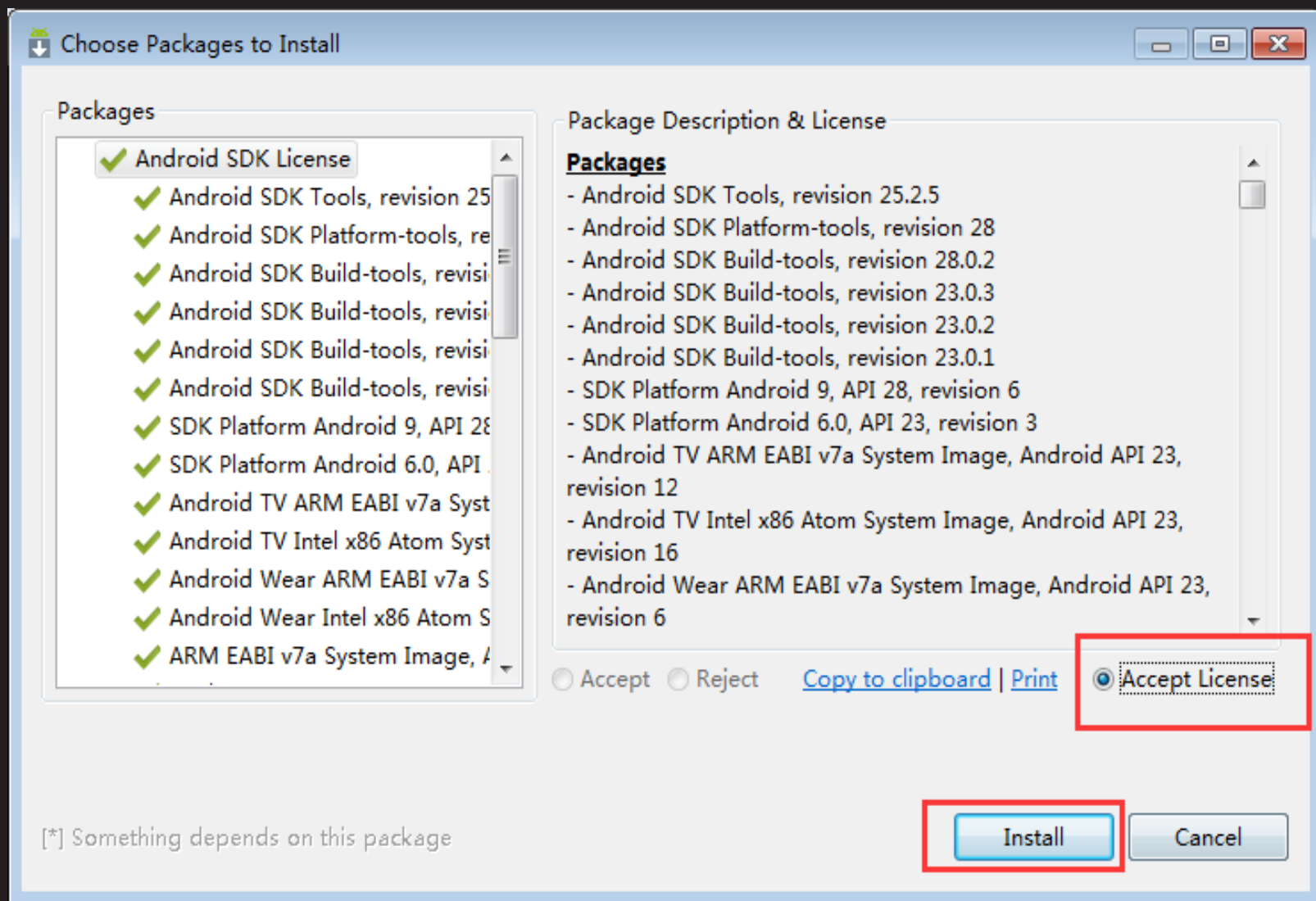


# 下载Android平台和工具（续6）

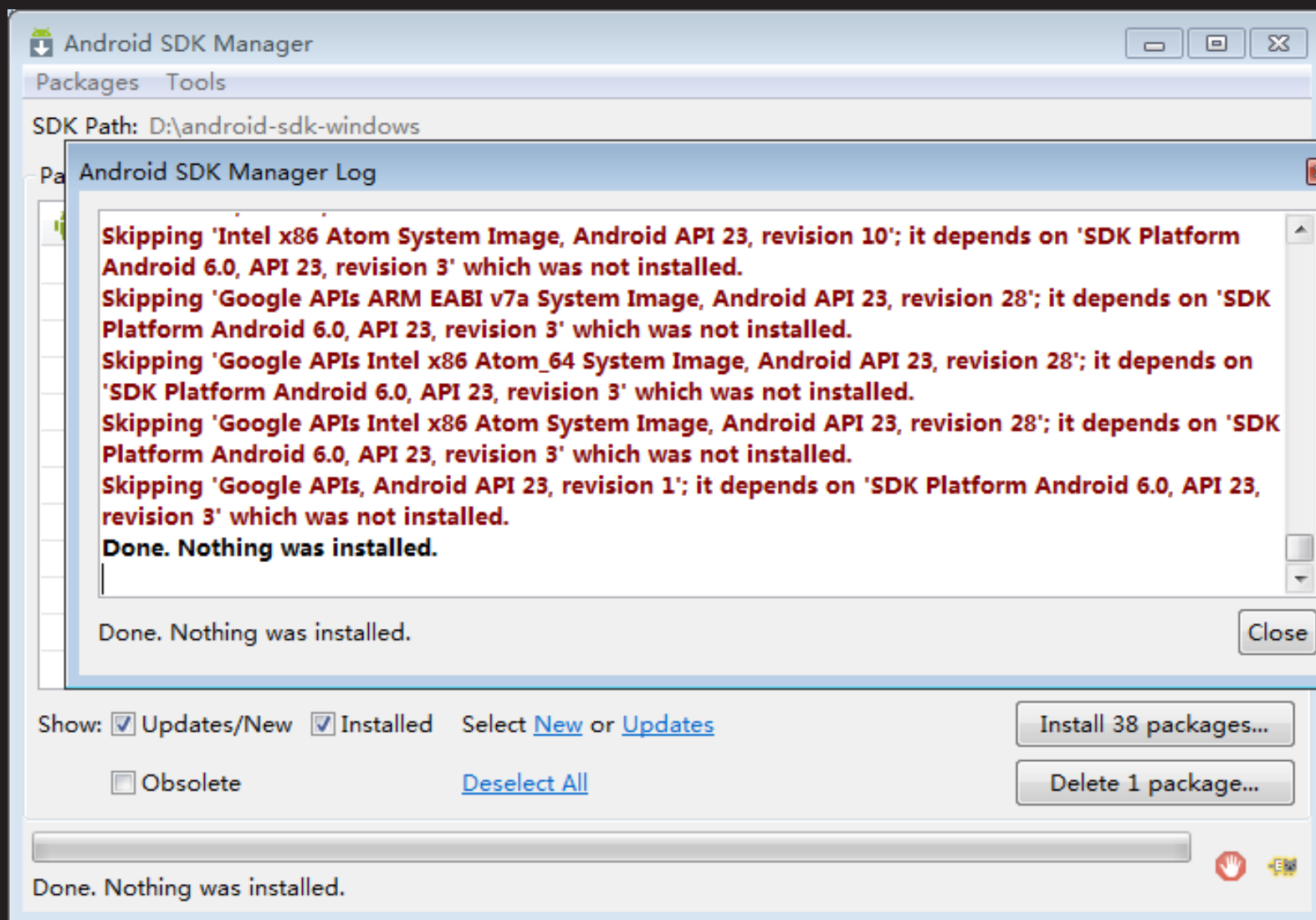




# 下载Android平台和工具（续7）



# 下载Android平台和工具（续8）



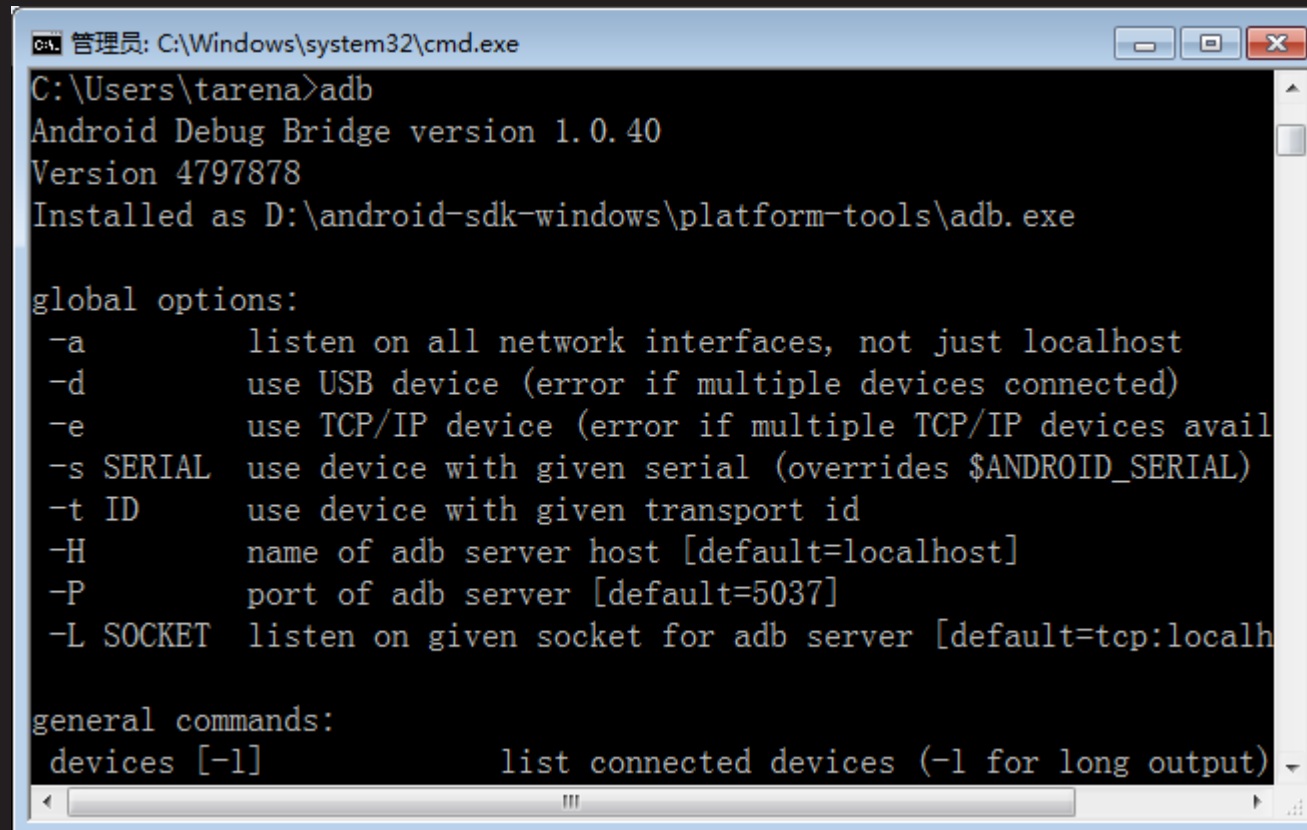
# 配置平台和工具环境变量

- 步骤4、配置环境变量：在系统变量中找到Path，点击“编辑”，在末尾加入
  - %ANDROID\_HOME%\platform-tools;
  - %ANDROID\_HOME%\tools;
  - %ANDROID\_HOME%\build-tools;



# 检查Android环境

- 步骤5、检查Android环境：
  - 打开命令行窗口
  - 输入adb



```
管理员: C:\Windows\system32\cmd.exe
C:\Users\tarena>adb
Android Debug Bridge version 1.0.40
Version 4797878
Installed as D:\android-sdk-windows\platform-tools\adb.exe

global options:
-a          listen on all network interfaces, not just localhost
-d          use USB device (error if multiple devices connected)
-e          use TCP/IP device (error if multiple TCP/IP devices avail
-s SERIAL   use device with given serial (overrides $ANDROID_SERIAL)
-t ID       use device with given transport id
-H          name of adb server host [default=localhost]
-P          port of adb server [default=5037]
-L SOCKET   listen on given socket for adb server [default=tcp:localh

general commands:
devices [-l]          list connected devices (-l for long output)
```



# 总结和答疑

