

第二章 Android开发前的准备

本章内容

- **Android开发环境搭建**
- **第一个Android程序： HelloWorld**

Android开发环境搭建

- Android开发所需要的工具：
JDK + Eclipse + Android SDK + ADT

本部分主要内容

- JDK的安装、JAVA环境配置及测试
- Android SDK的下载、安装和配置
 - ◆ Android Software Development Kits
- Eclipse的下载、安装和启动
- Android ADT插件的下载、安装和测试
 - ◆ Android Development Tools
- 在Eclipse中设置Android SDK
- 管理升级Android SDK
- 创建AVD（Android Virtual Device）
 - ◆ Android模拟器（Android Emulator）

JDK的安装及JAVA环境配置

- JDK: Java Development Kit, 是JAVA的核心, 也是搭建Java开发环境的基本要素
- JRE: Java Run Environment, 这是Java的运行环境
- 注意区别JDK和JRE。JDK工具里面就包含了JRE, 同时还包括一些其他的工具和JAVA的基础类库。
- 下载说明:
 - 由于2009年4月SUN公司被Oracle公司收购, 输入旧的SUN公司的官网 <http://java.sun.com>, 会自动跳转到 <http://www.oracle.com/technetwork/java/index.html>

JDK下载

- 具体下载地址为:

- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- 目前最新版为: **jdk-7u21-windows-i586.exe**

Java Platform, Standard Edition		
Java SE 7u9 This releases address security concerns. Oracle strongly recommends that all Java SE 7 users upgrade to this release. JavaFX 2.2.3 is now bundled with the JDK on Windows, Mac and Linux x86/x64. Learn more ▶ "What Java Do I Need?" You must have a copy of the JRE (Java Runtime Environment) on your system to run Java applications and applets. To develop Java applications and applets, you need the JDK (Java Development Kit), which includes the JRE.	<div>JDK DOWNLOAD ⬇</div> JDK 7 Docs <ul style="list-style-type: none">▪ Installation Instructions▪ ReadMe▪ ReleaseNotes▪ Oracle License▪ Java SE Products▪ Third Party Licenses	<div>JRE DOWNLOAD ⬇</div> JRE 7 Docs <ul style="list-style-type: none">▪ Installation Instructions▪ ReadMe▪ ReleaseNotes▪ Oracle License▪ Java SE Products▪ Third Party Licenses

JDK安装

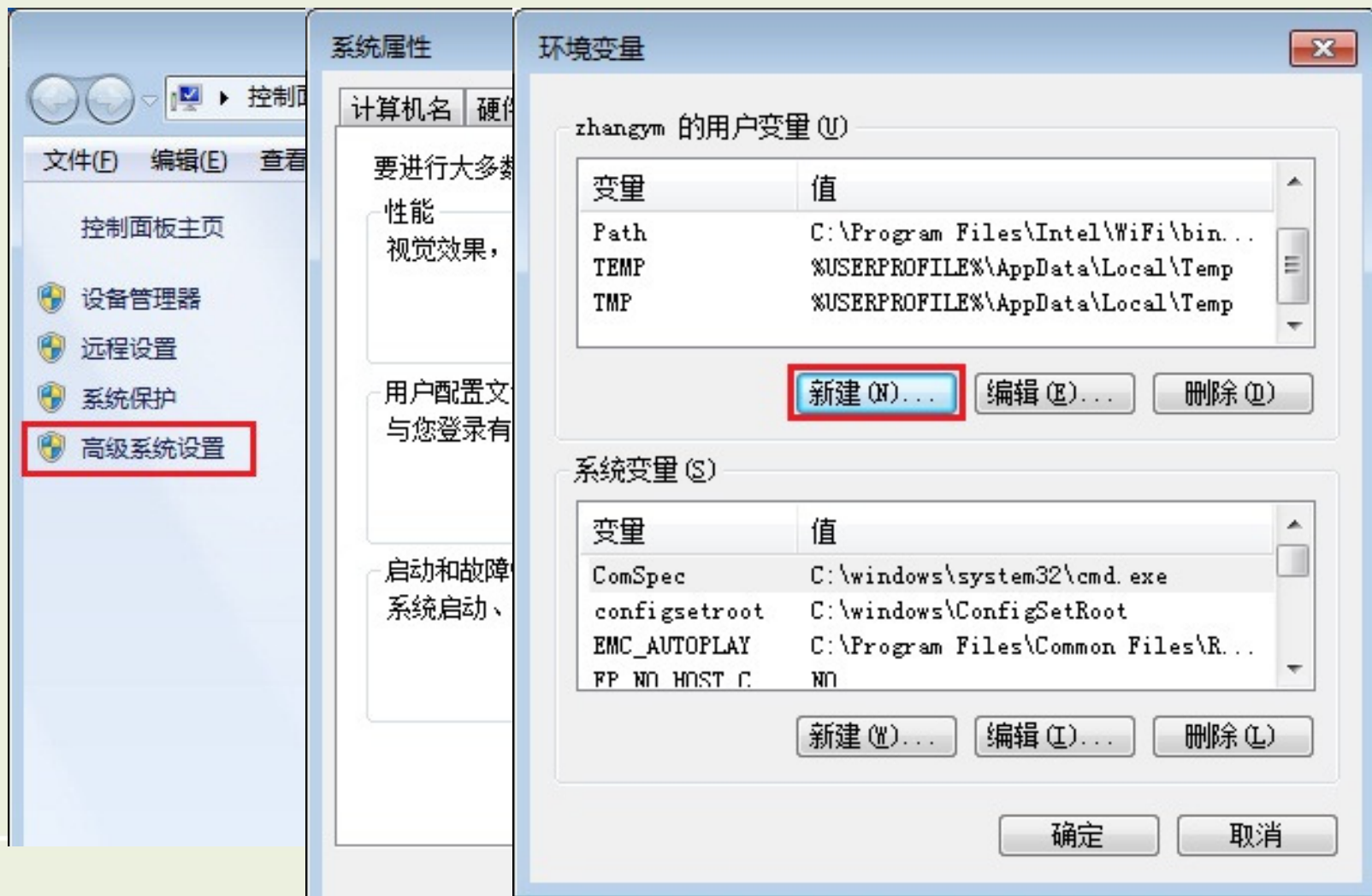
- 双击jdk-7u6-windows-i586.exe安装JDK



- 注意最好不要将其安装在有空格或者中文文件夹下，其他无限制，注意记下其安装路径，例如：C:\JDK7.0
 - 本机实际安装路径：C:\Program Files\Java\jdk1.7.0_06
- JAVA环境变量
 - Path ---- 用于指定操作系统的可执行指令的路径
 - CLASSPATH ----用于指定Java的类文件存放路径

JAVA环境配置

- 在桌面的“计算机”图标上单击右键—>选择“属性”—>页面左侧选择“高级系统设置”—>选择最下面的“环境变量”

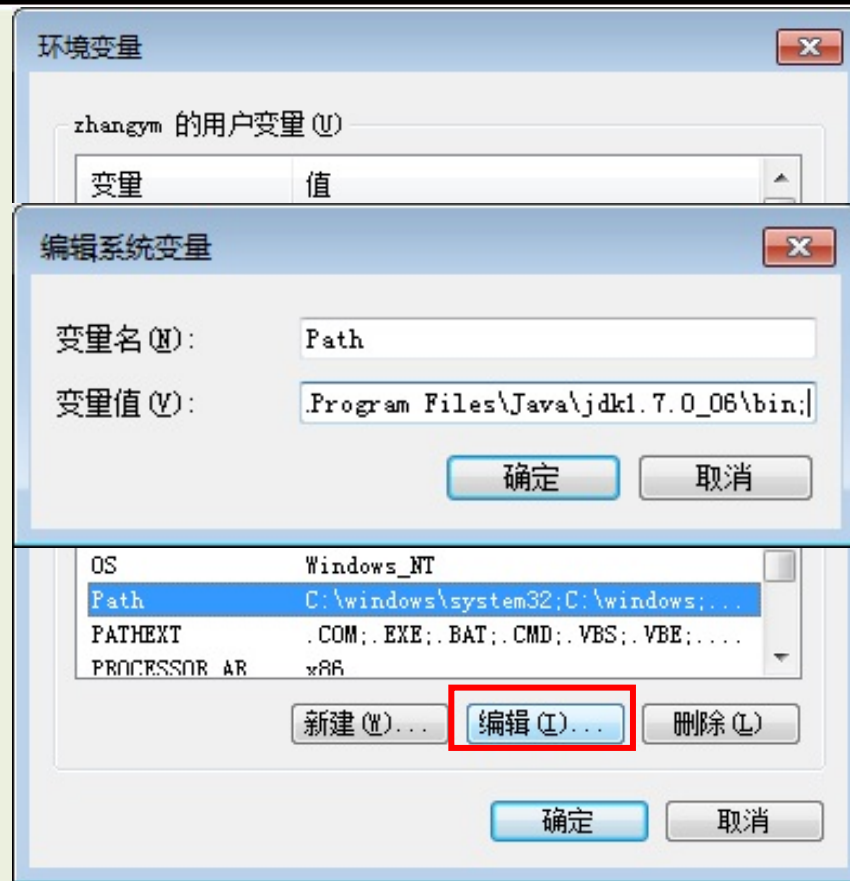




- 在“XXX的用户变量”中新建变量“**JAVA_PATH**”，值为JDK安装的路径，本机为：“**C:\Program Files\Java\jdk1.7.0_06**”。



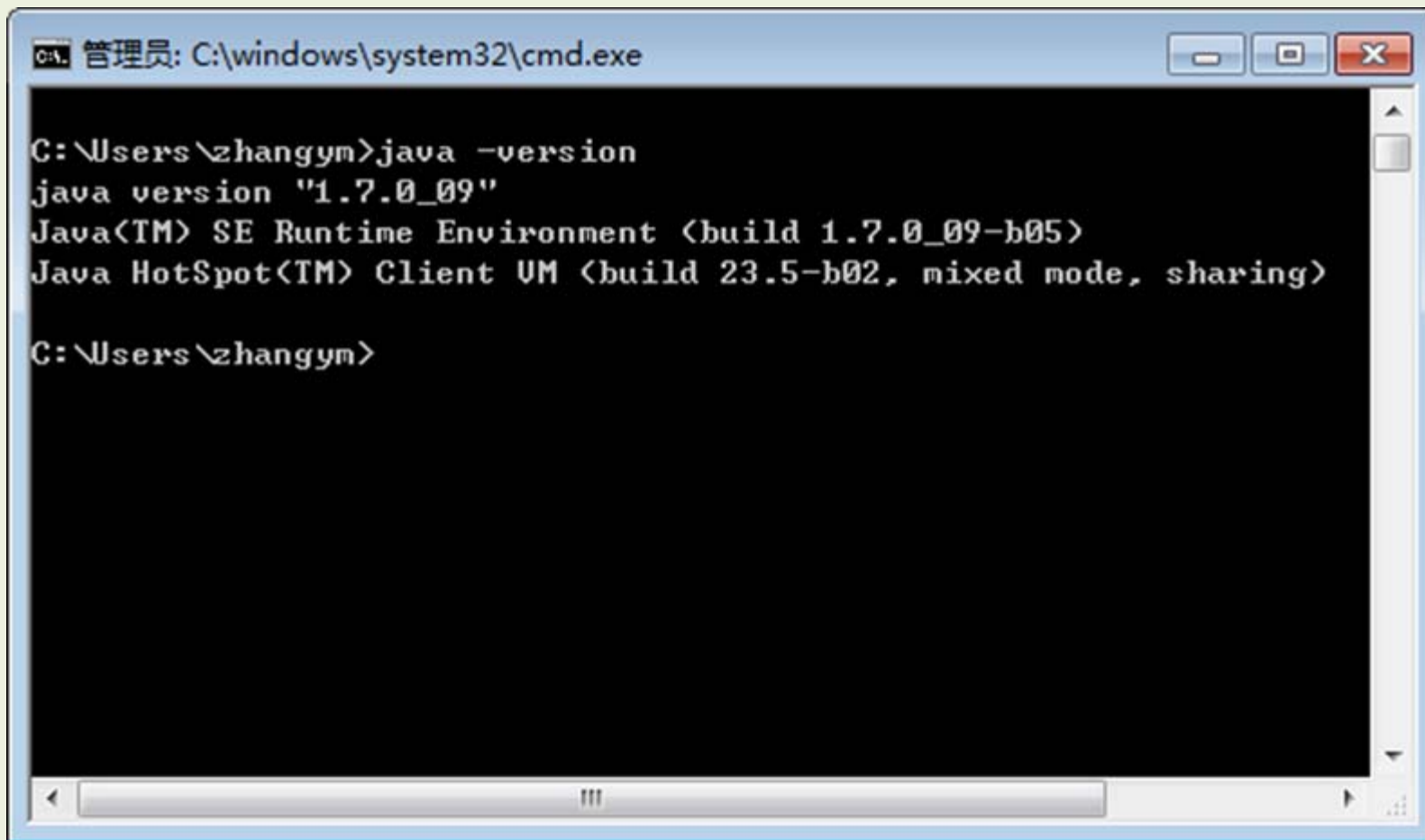
- 新建变量**classpath**，值为“**;%JAVA_PATH%\lib\dt.jar;%JAVA_PATH%\lib\tools.jar;%JAVA_PATH%\jre\lib\rt.jar;**”。



- 修改“系统变量”中的“Path”变量的值，在最后添加默认安装JDK的路径为“C:\Program Files\Java\jdk1.7.0_06”（根据实际安装路径而定）
- 也可在用户变量中添加“Path”变量和上述值

Java环境测试

- 为了确认JDK是否安装成功，打开命令提示符，键入 `java -version`，出现下面的结果，就表示JDK已经安装成功



```
管理员: C:\windows\system32\cmd.exe

C:\Users\zhangym>java -version
java version "1.7.0_09"
Java(TM) SE Runtime Environment (build 1.7.0_09-b05)
Java HotSpot(TM) Client VM (build 23.5-b02, mixed mode, sharing)

C:\Users\zhangym>
```

Android SDK的下载

◆从Google的官网上下载最新版本的Android SDK，分为Windows、Mac OS、Linux三种版本。选择合适的版本下载：

➤ <http://developer.android.com/sdk/index.html>

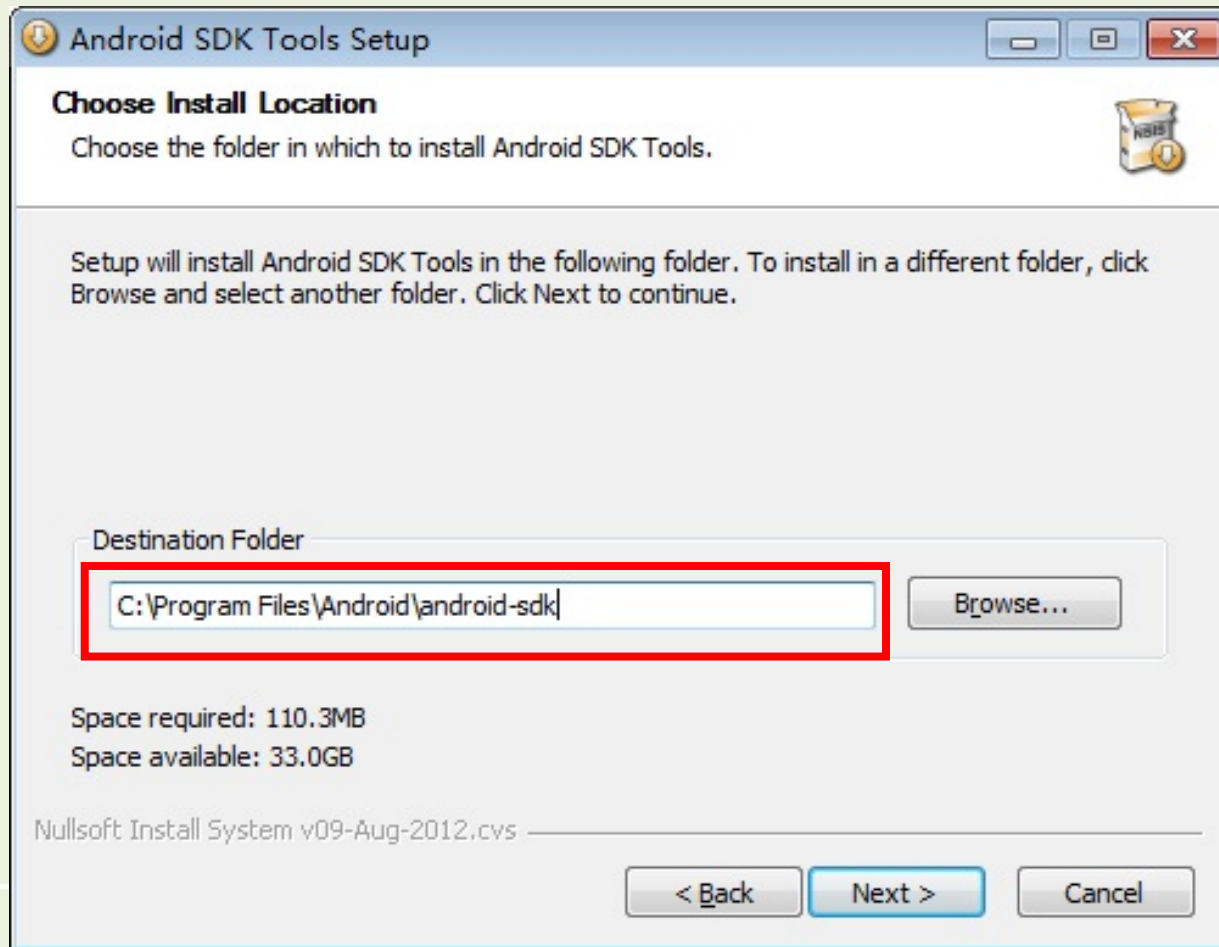
SDK Tools Only

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r21-windows.zip	99093893 bytes	7311452823470365f7975a545f8a2be4
	installer_r21-windows.exe (Recommended)	77523031 bytes	29ca8cb8f0bc8db627fa2adc2139a3cc
Mac OS X	android-sdk_r21-macosx.zip	65792626 bytes	67e46adca90dd18d7291443f6c15d6af
Linux	android-sdk_r21-linux.tgz	91378351 bytes	7f8d73b629f808cdcfc9f9900bbd7580

Android SDK的安装

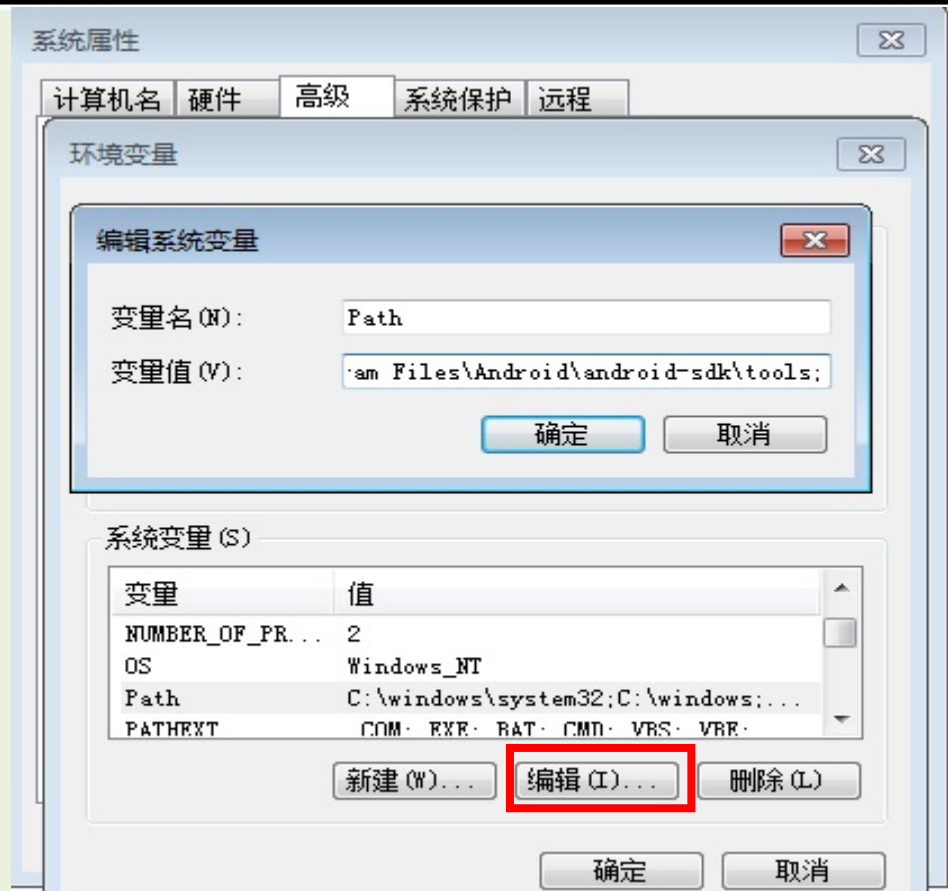
◆如果下载的是exe安装文件，点击安装，安装路径可选择默认的文件夹。（如果下载的是压缩包，解压缩即可。）

➤本机实际安装路径为：[C:\Program Files\Android\android-sdk](#)



Android SDK的配置

- ◆与JDK的配置步骤类似，打开“环境变量”配置窗口：



- 修改“系统变量”中的“Path”变量的值，在最后添加：“C:\Program Files\Android\android-sdk\tools;C:\Program Files\Android\android-sdk\platform-tools;”（根据实际安装路径而定）
- 检验是否安装成功：在命令提示符中运行“android -h”以及“adb -h”

Eclipse的下载

◆ 下载Window版本Eclipse开发工具:

- <http://www.eclipse.org/downloads/>
- 选择“Eclipse IDE for Java Developers”下载即可。

Eclipse Downloads

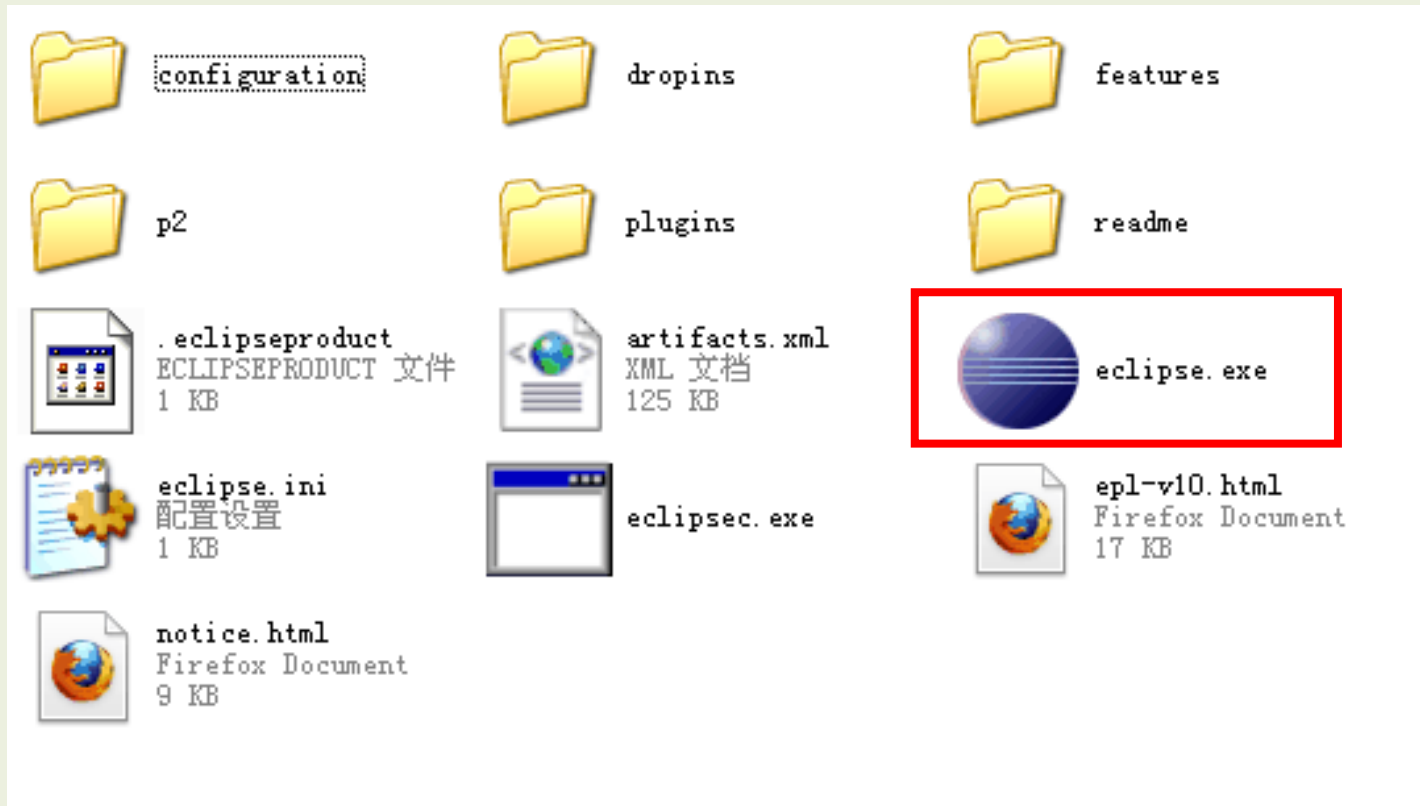
[Packages](#) [Developer Builds](#) [Projects](#)

Eclipse Juno (4.2) SR1 Packages for Windows

	Eclipse IDE for Java EE Developers , 221 MB Downloaded 1,545,684 Times Details	 Windows 32 Bit Windows 64 Bit
	Eclipse IDE for Java Developers , 150 MB Downloaded 715,337 Times Details	 Windows 32 Bit Windows 64 Bit
	Eclipse IDE for C/C++ Developers , 129 MB Downloaded 348,855 Times Details	 Windows 32 Bit Windows 64 Bit
	JBoss Developer Studio Download JBoss Developer Studio 5.0 with packages for Mac, Windows or Linux.	 Download
	Eclipse for Mobile Developers , 144 MB Downloaded 209,769 Times Details	 Windows 32 Bit Windows 64 Bit

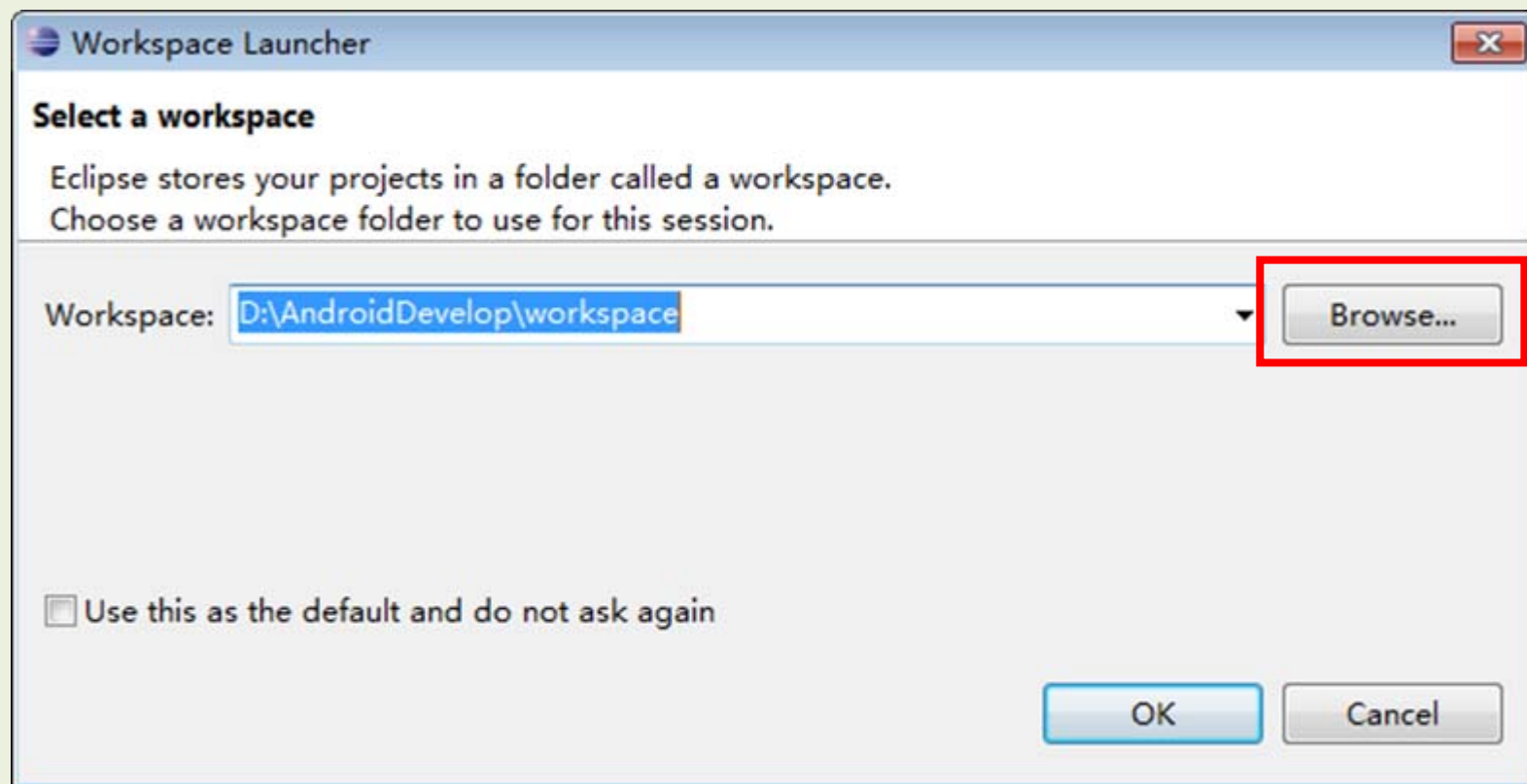
Eclipse的安装

- ◆解压缩，即可运行解压文件中的eclipse.exe来打开Eclipse开发工具



Eclipse首次启动

- 设置工作区域Workspace（又叫工作空间，以后建立Project的默认存放位置）



Android ADT的下载和安装

- 通过Eclipse安装ADT（Android Development Tools）
 - 官方网页上的ADT安装说明，地址：
 - <http://developer.android.com/sdk/installing/installing-adt.html>

Download the ADT Plugin

1. Start Eclipse, then select **Help > Install New Software**.
2. Click **Add**, in the top-right corner.
3. In the Add Repository dialog that appears, enter "ADT Plugin" for the *Name* and the following URL for the *Location*:

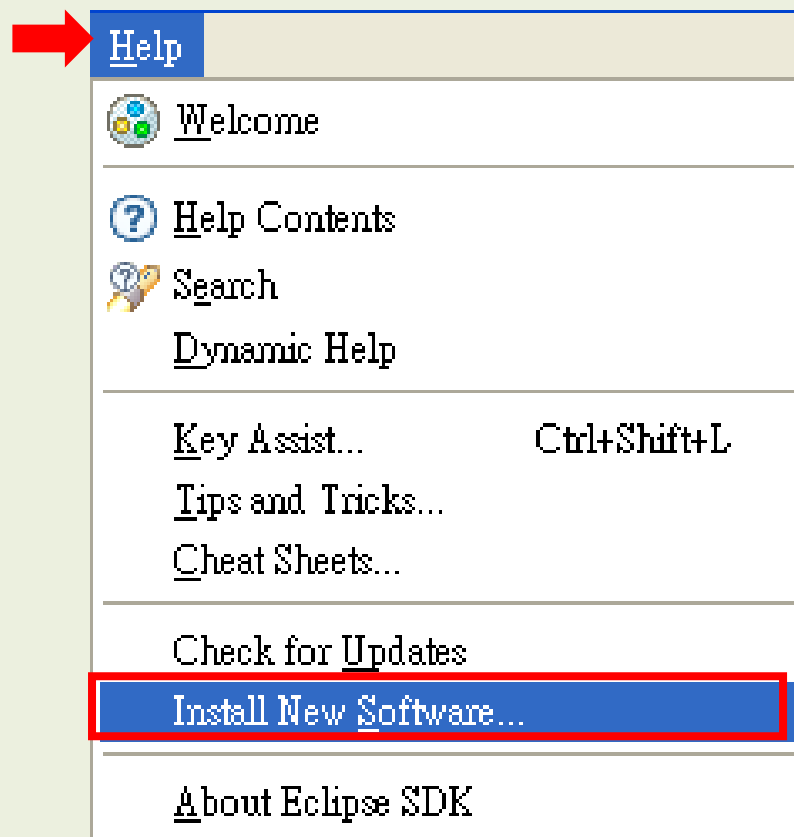
```
https://dl-ssl.google.com/android/eclipse/
```

4. Click **OK**.
If you have trouble acquiring the plugin, try using "http" in the Location URL, instead of "https" (https is preferred for security reasons).
5. In the Available Software dialog, select the checkbox next to Developer Tools and click **Next**.
6. In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
7. Read and accept the license agreements, then click **Finish**.
If you get a security warning saying that the authenticity or validity of the software can't be established, click **OK**.
8. When the installation completes, restart Eclipse.

Android ADT的下载和安装

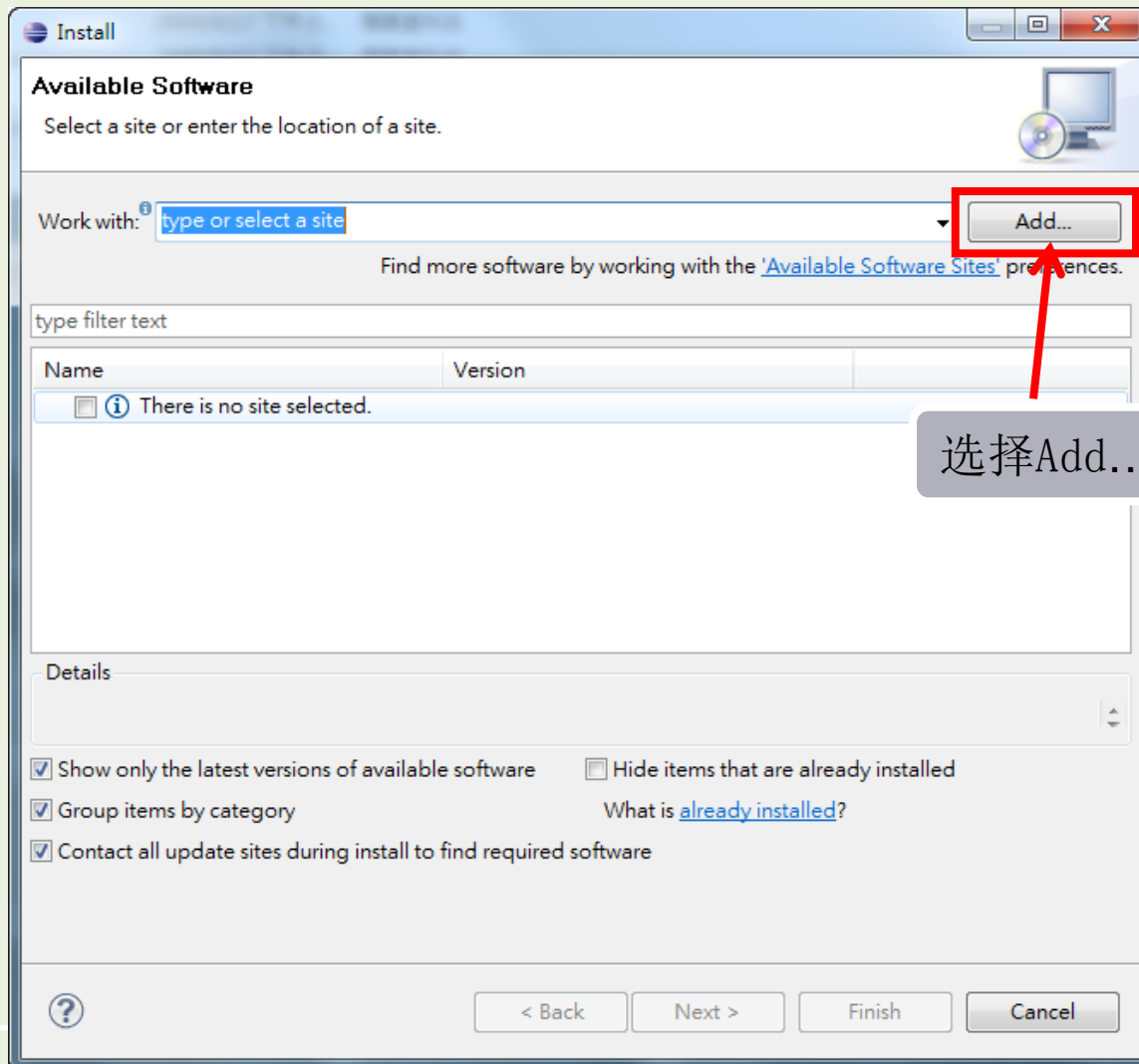
■ 安装Android Development Tools (ADT)

◆ 在 Eclipse 菜单上选择Help→ Install New Software...



Android ADT的下载和安装

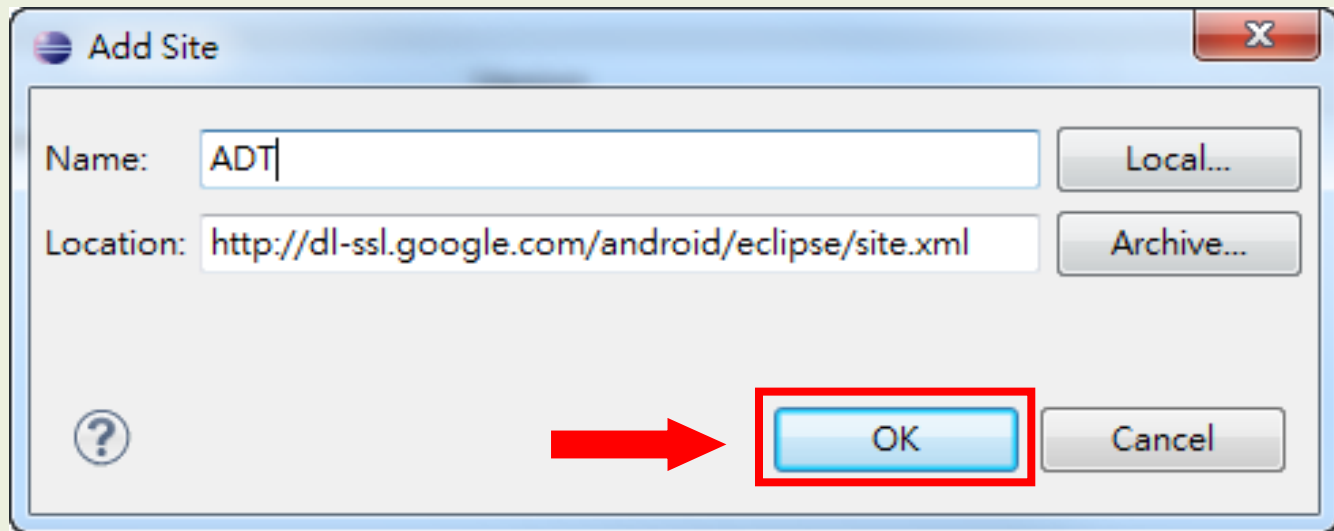
■ 安装Android Development Tools (ADT)



选择Add...按钮

Android ADT的下载和安装

■ 安装Android Development Tools (ADT)

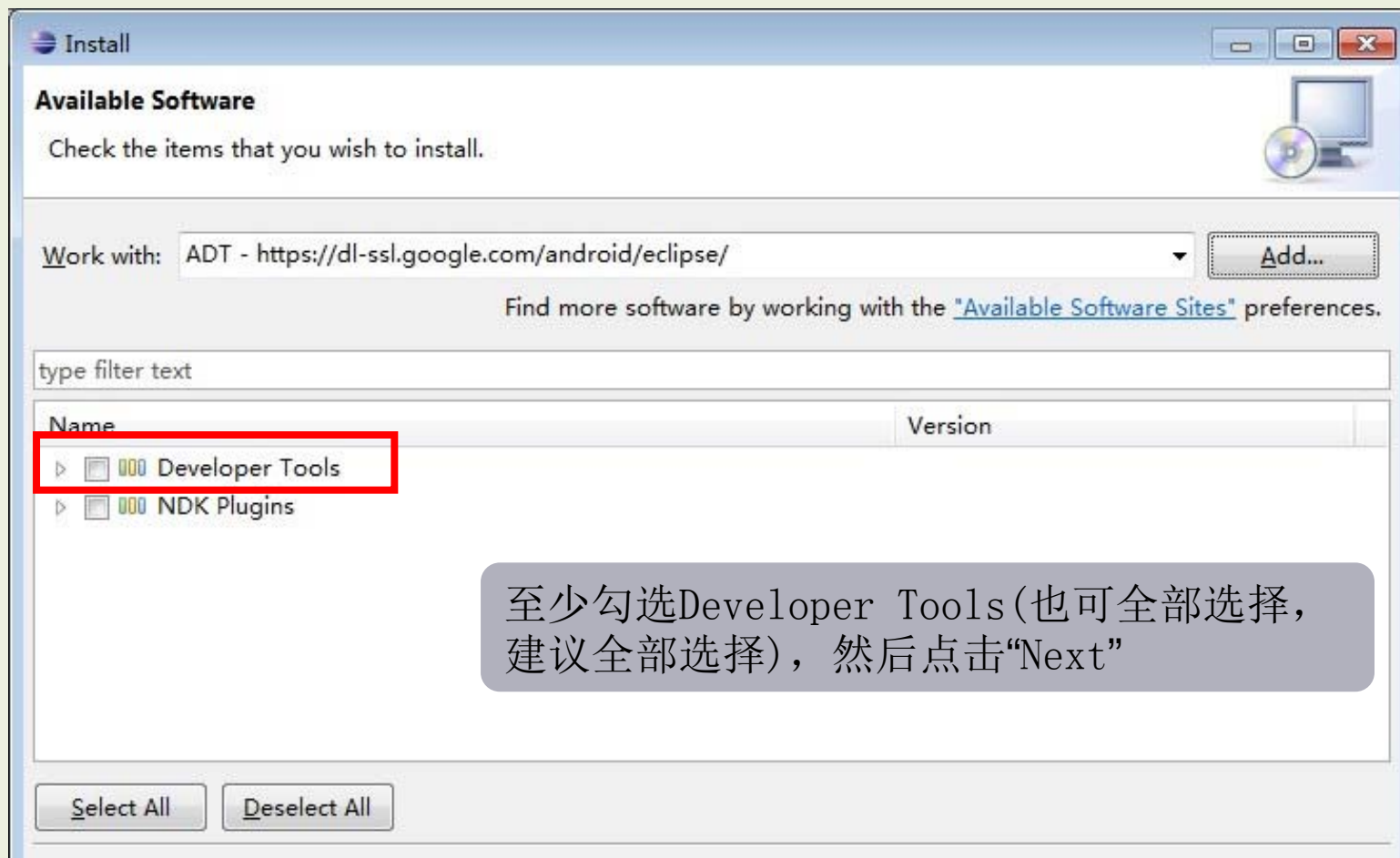


•填写说明:

- Name字段可自行命名; Location字段填下面的网址:
- <https://dl-ssl.google.com/android/eclipse/>
或者: <http://dl-ssl.google.com/android/eclipse/>

Android ADT的下载和安装

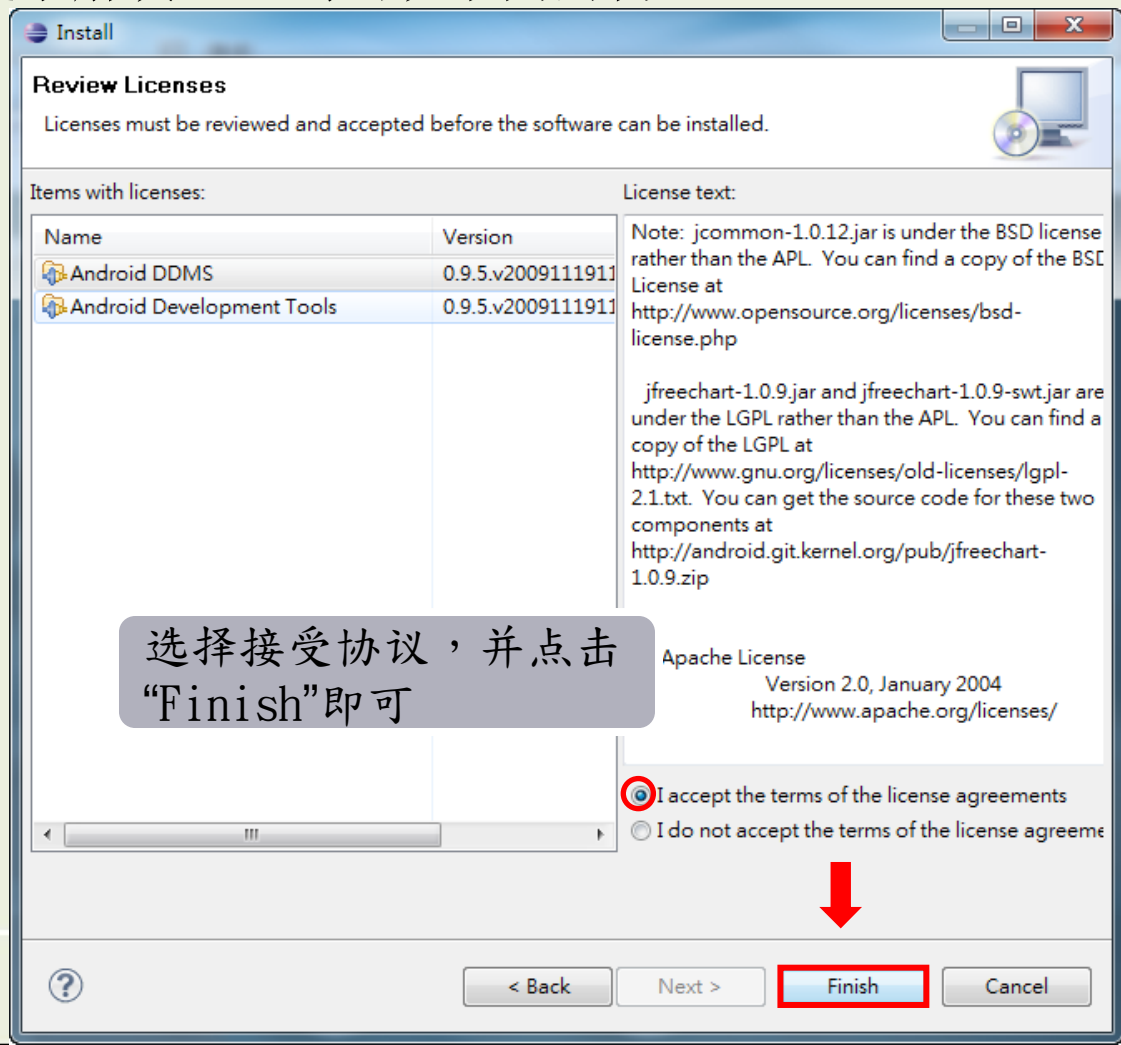
■ 安装Android Development Tools (ADT)



Android ADT的下载和安装

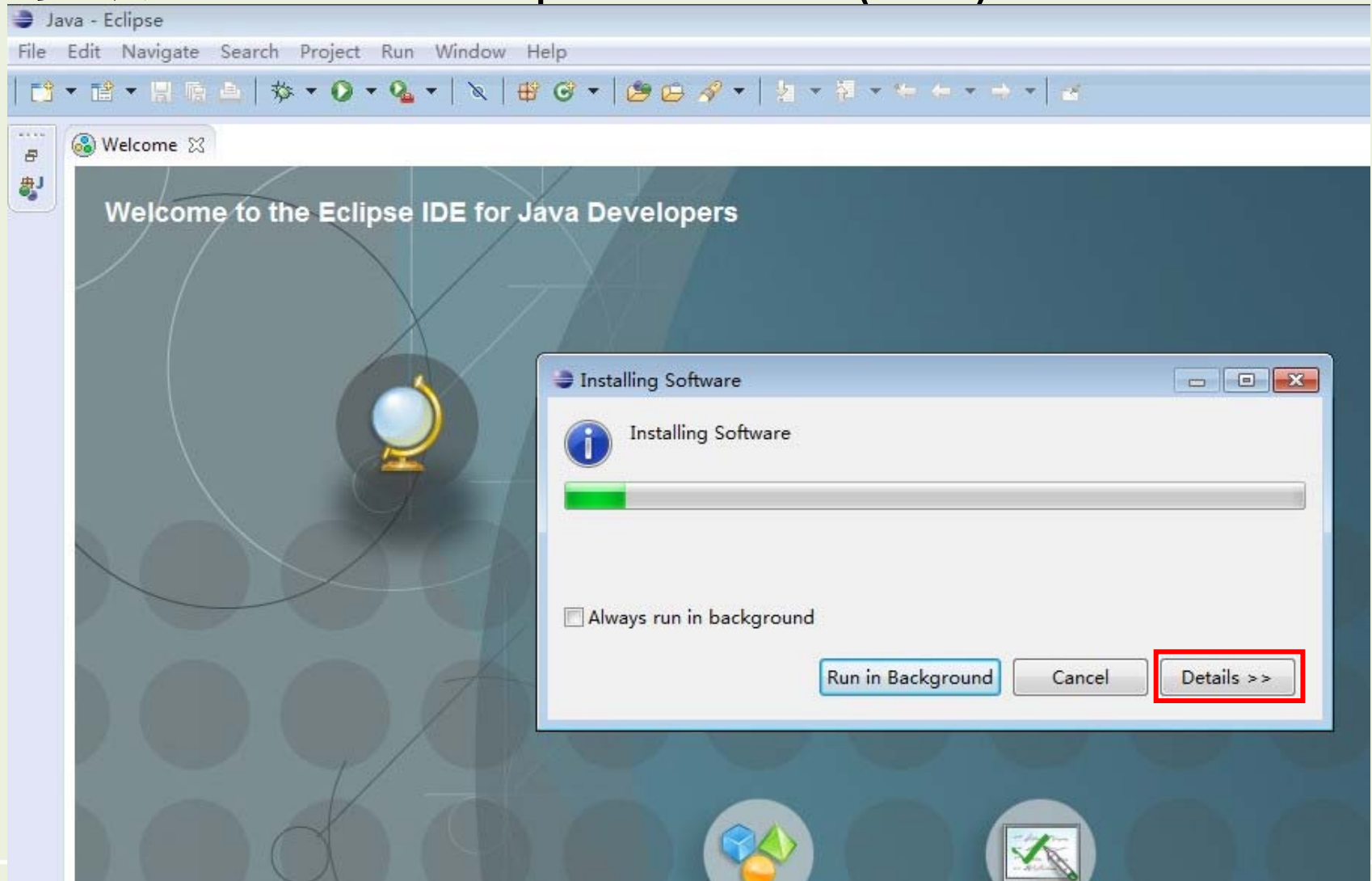
■ 安装Android Development Tools (ADT)

- ◆ 在接下来出现的窗口中一直选择点击“Next”，如果出现错误提示，按提示解决。直到出现下面的窗口



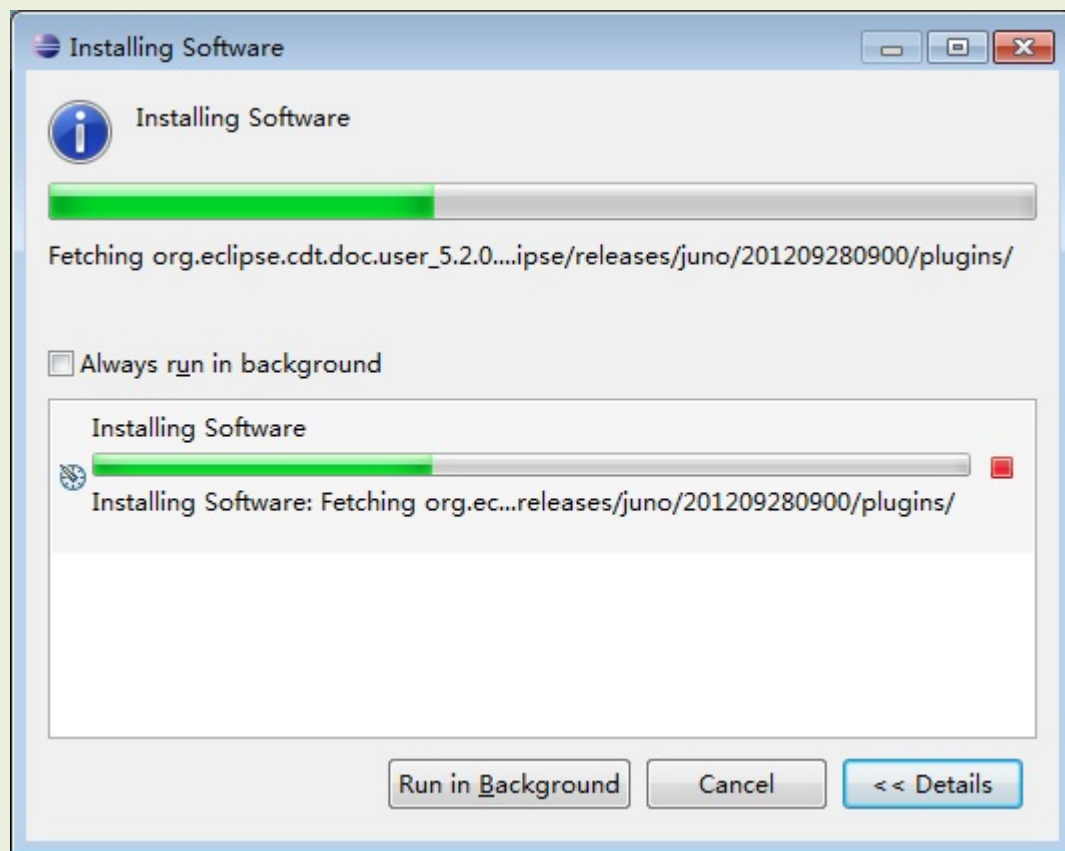
Android ADT的下载和安装

■ 安装Android Development Tools (ADT)



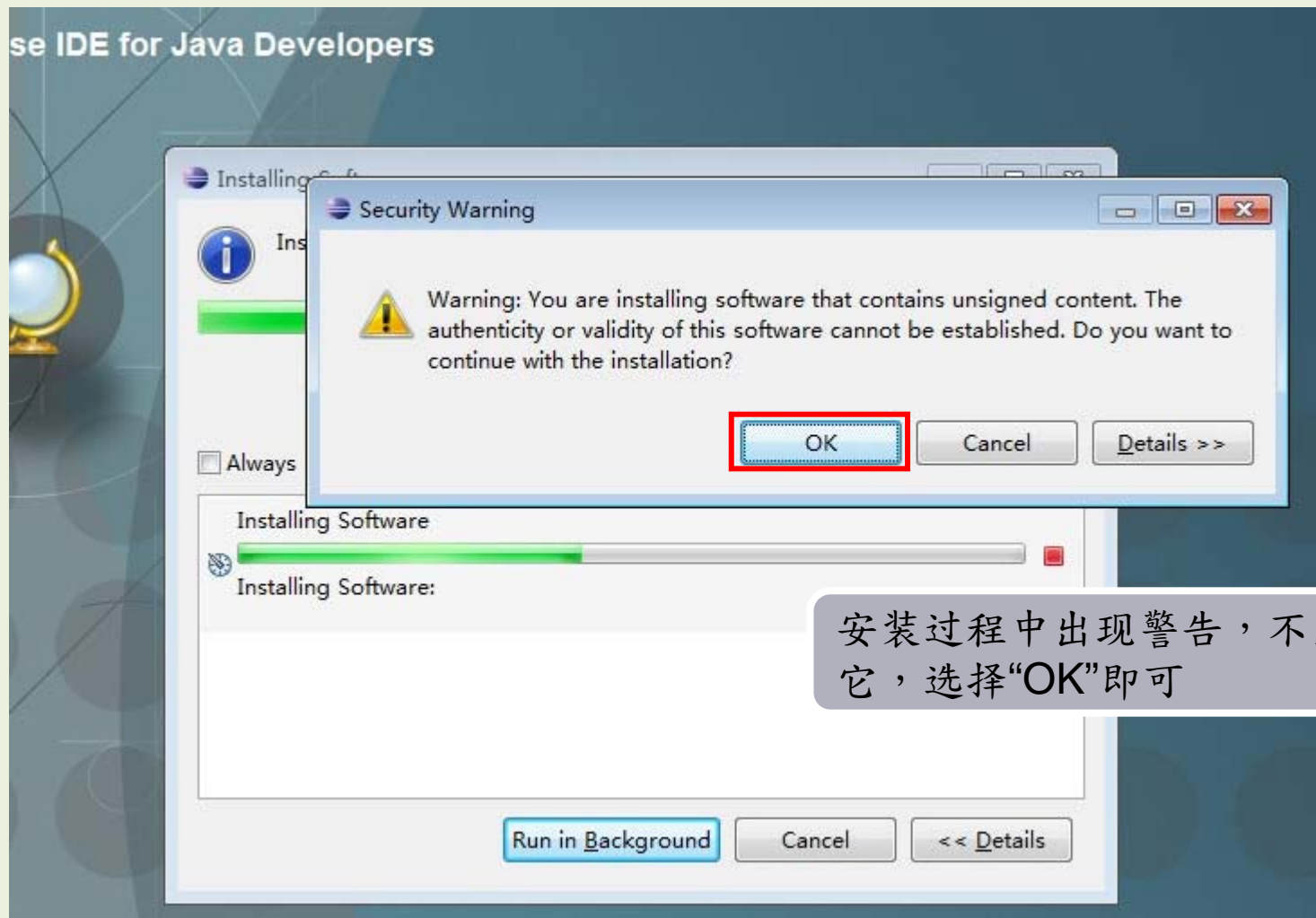
Android ADT的下载和安装

■ 安装Android Development Tools (ADT)



Android ADT的下载和安装

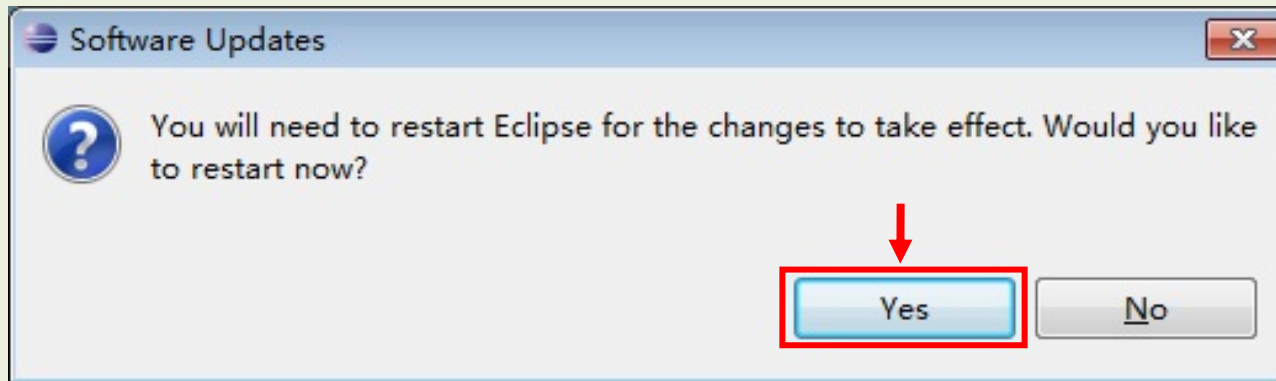
■ 安装Android Development Tools (ADT)



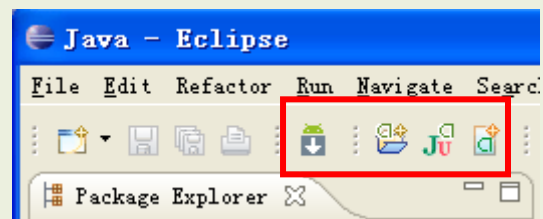
安装过程中出现警告，不用管它，选择“OK”即可

Android ADT的下载和安装

■ 安装Android Development Tools (ADT)



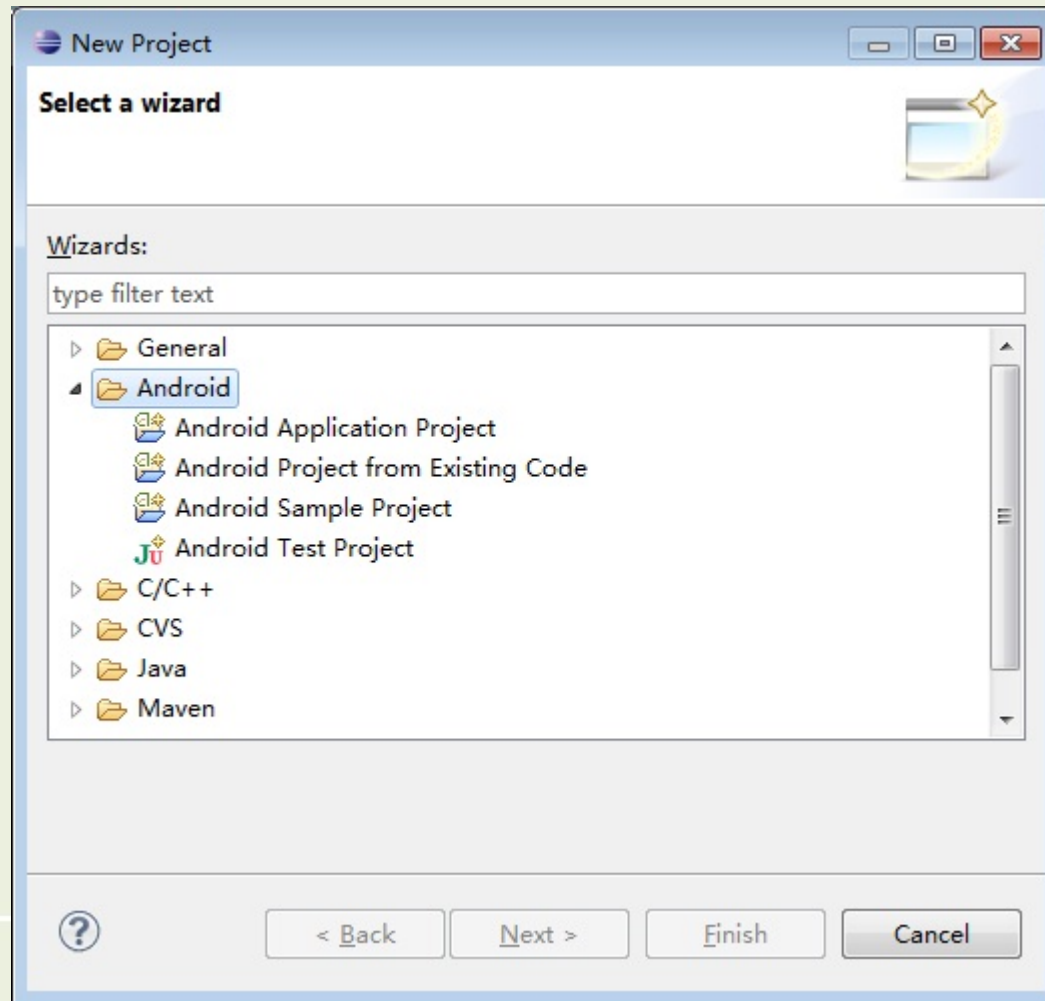
安装完成后需要重新启动Eclipse



重启后在工具栏中可以看到Android相关的图标


Android ADT测试

- 打开Eclipse，选择“File”—>选择“New”—>选择“Project”，
 - ◆ 如果在如图所示的界面中出现Android相关选项，则表示我们的Android配置成功，就可以开始Android开发



最新变化

◆注意现在最新变化：已经提供“三合一”包下载

 Developers ▾ | Design Develop Distribute

Training API Guides Reference Tools

Developer Tools

Download ▴

Setting Up the ADT Bundle

Setting Up an Existing IDE ▾

Exploring the SDK

Download the NDK

Workflow ▾

Tools Help ▾

Revisions ▾

Extras ▾

Samples

ADK ▾


Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in **ADT (Android Developer Tools)** to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator



Download the SDK

ADT Bundle for Windows

在Eclipse中设置Android SDK

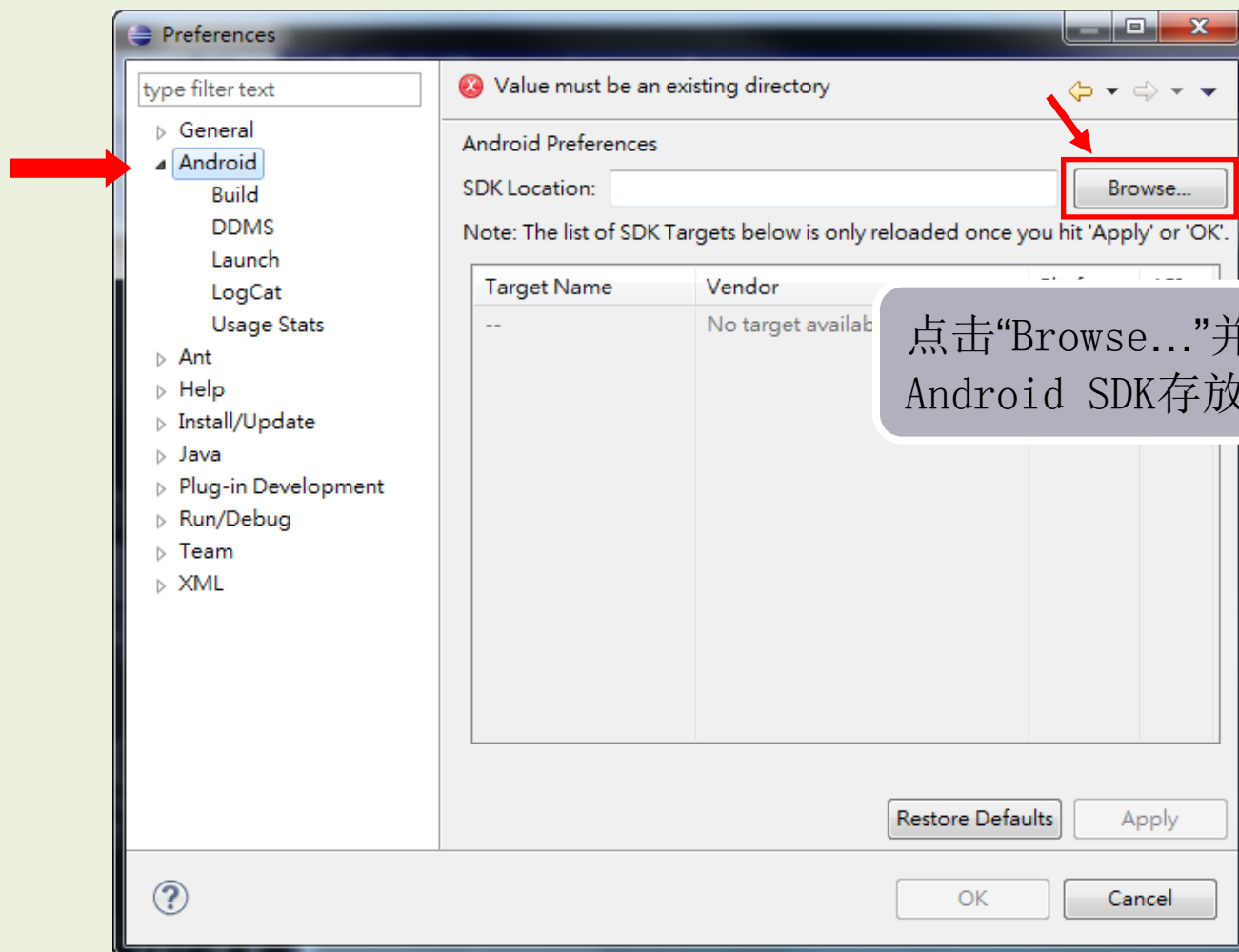
■ 设定Android SDK细项



点击Window → Preferences进行Android SDK细项设定

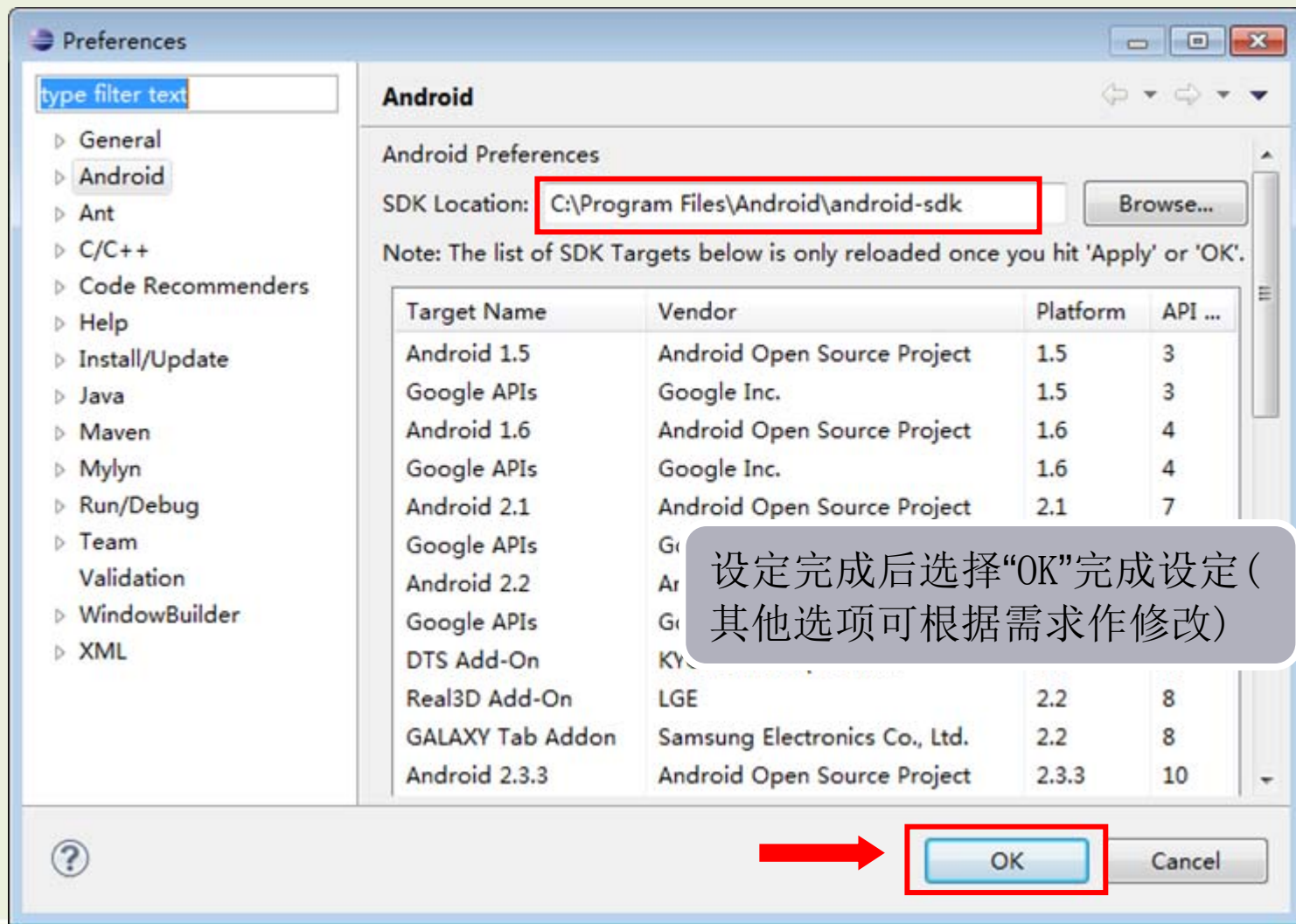
在Eclipse中设置Android SDK

■ 设定Android SDK细项



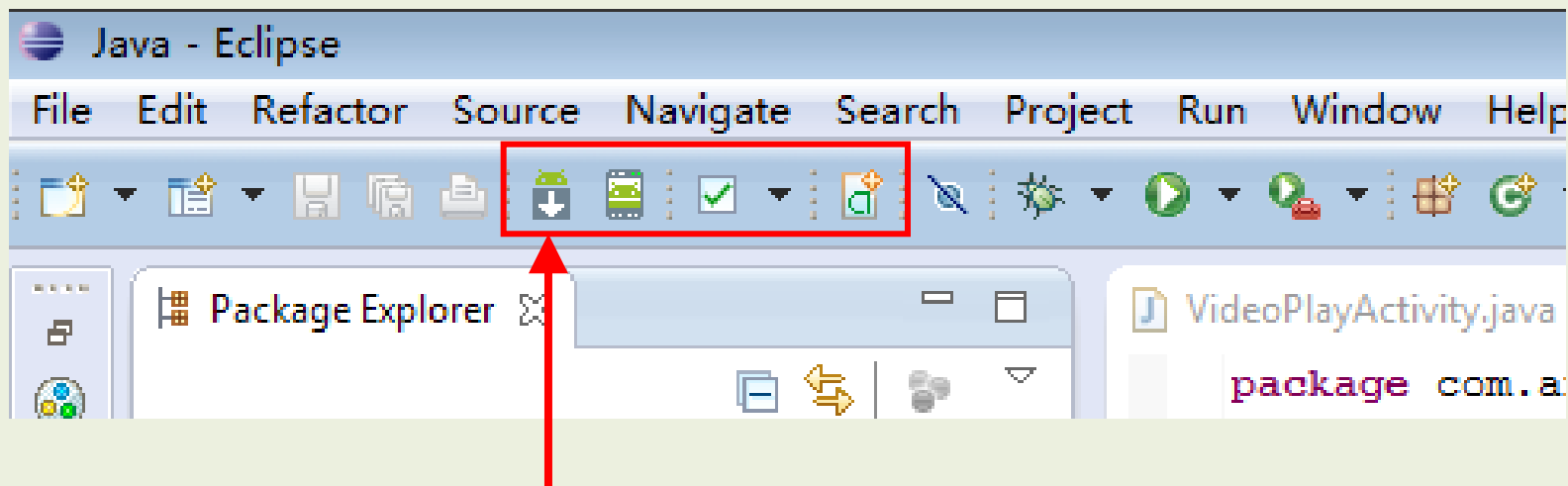
在Eclipse中设置Android SDK

■ 设定Android SDK细项



在Eclipse中设置Android SDK

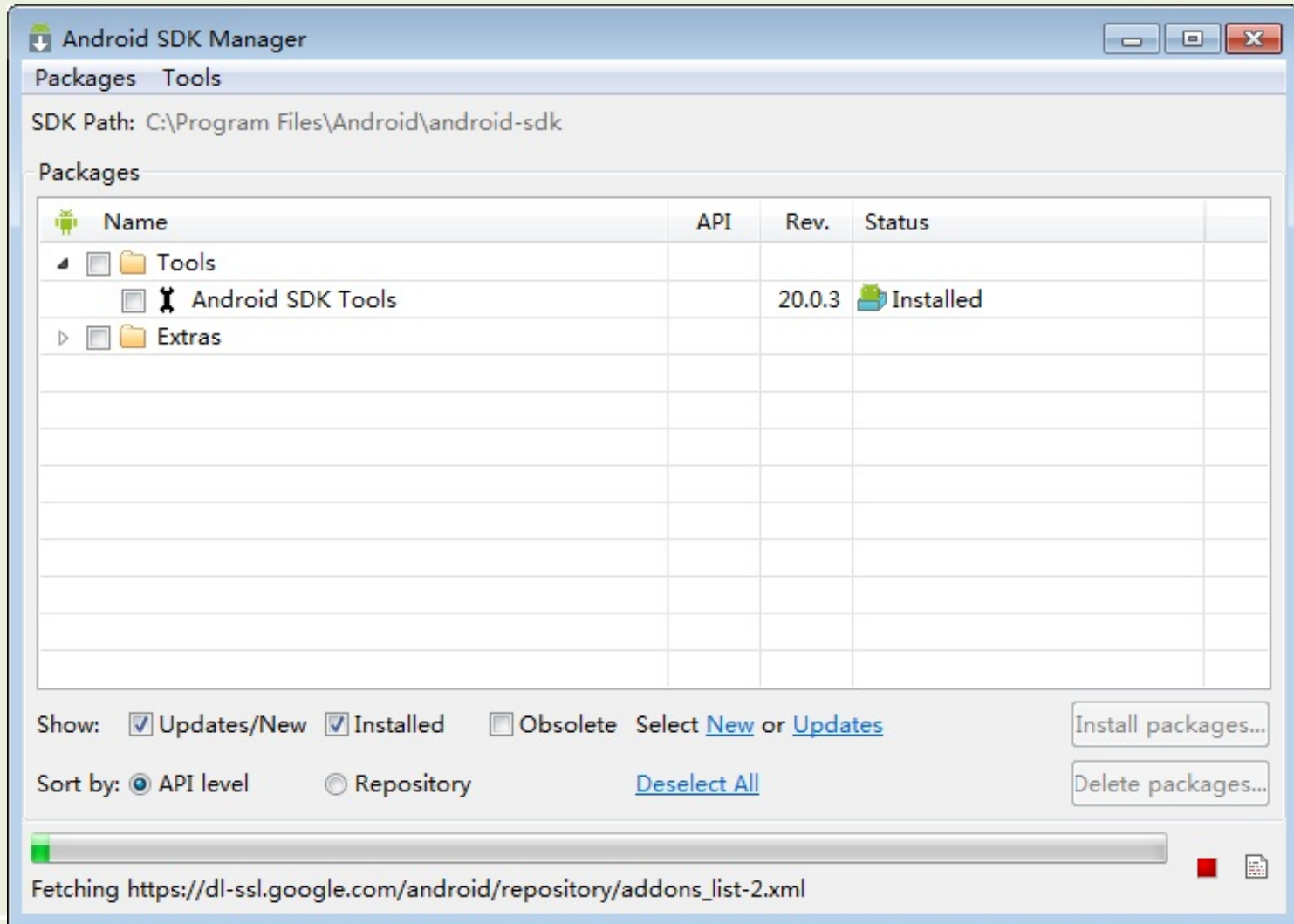
■ 设定Android SDK细项



前面设定完成后，回到Eclipse界面，选择红色箭头所示的工具栏图标，可以打开“Android SDK Manager”，进行Android SDK相关版本的下载升级等。

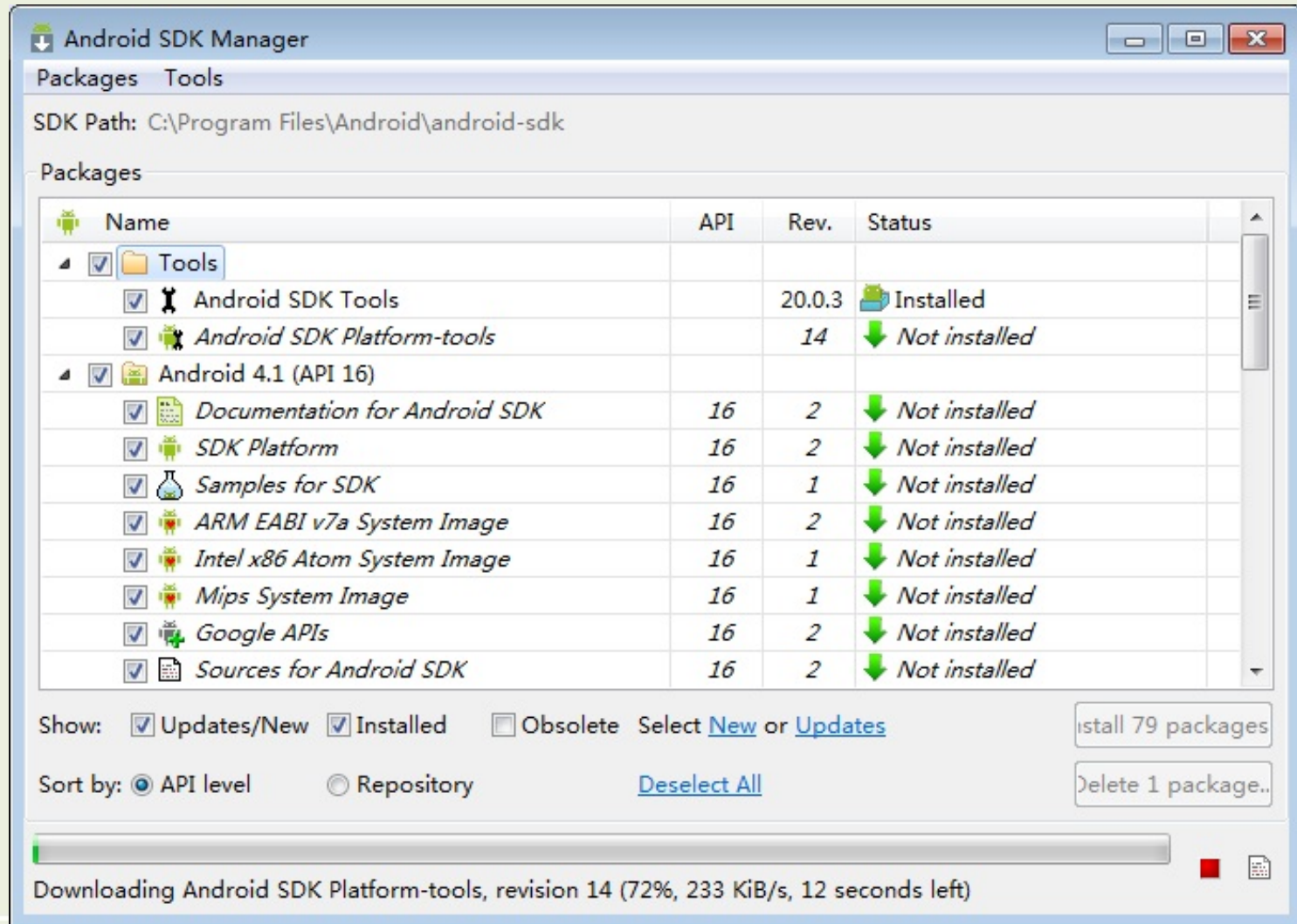
管理升级Android SDK

■ 管理、升级Android SDK



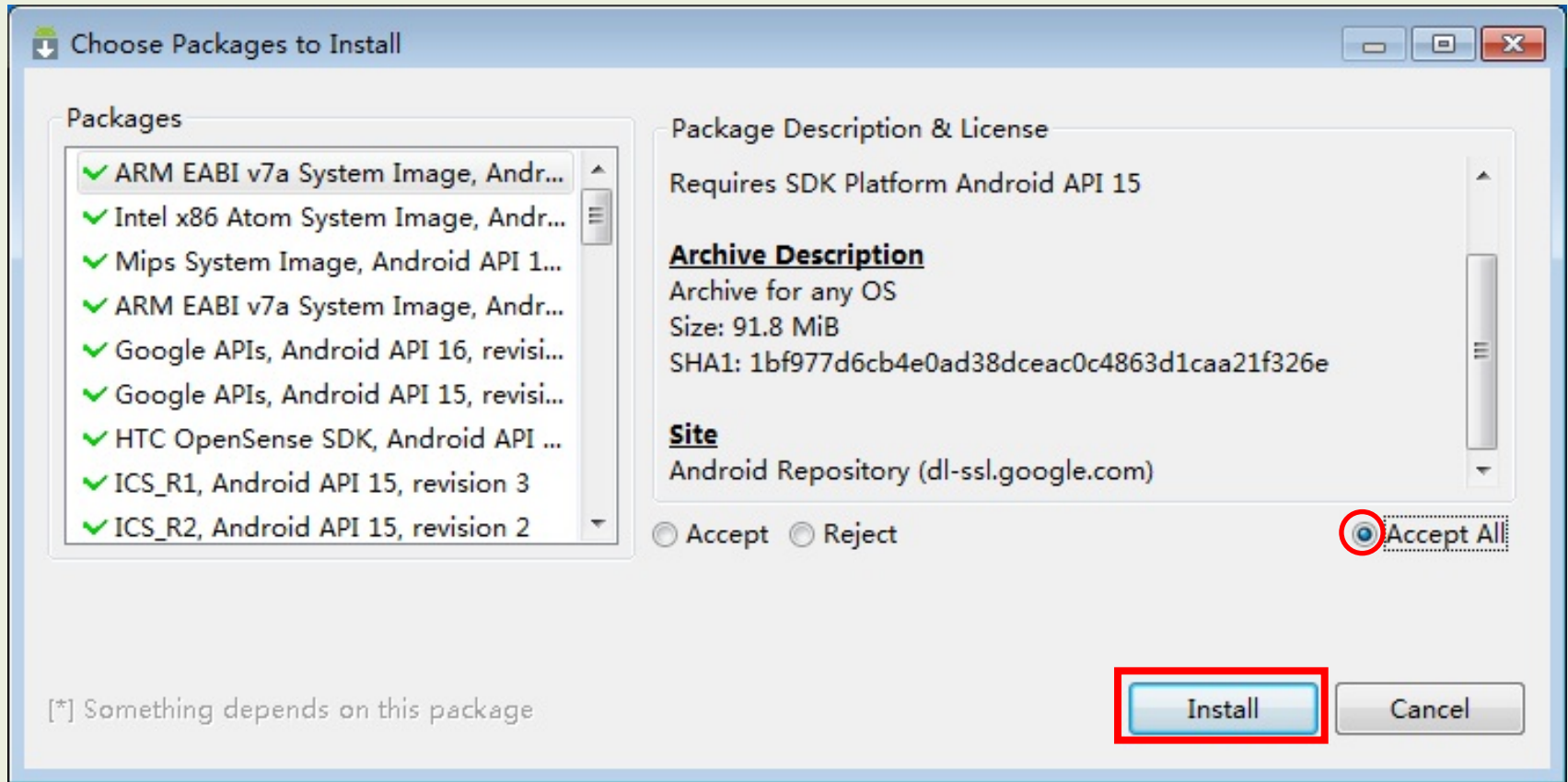
管理升级Android SDK

■ 管理、升级Android SDK



管理升级Android SDK

■ 管理、升级Android SDK

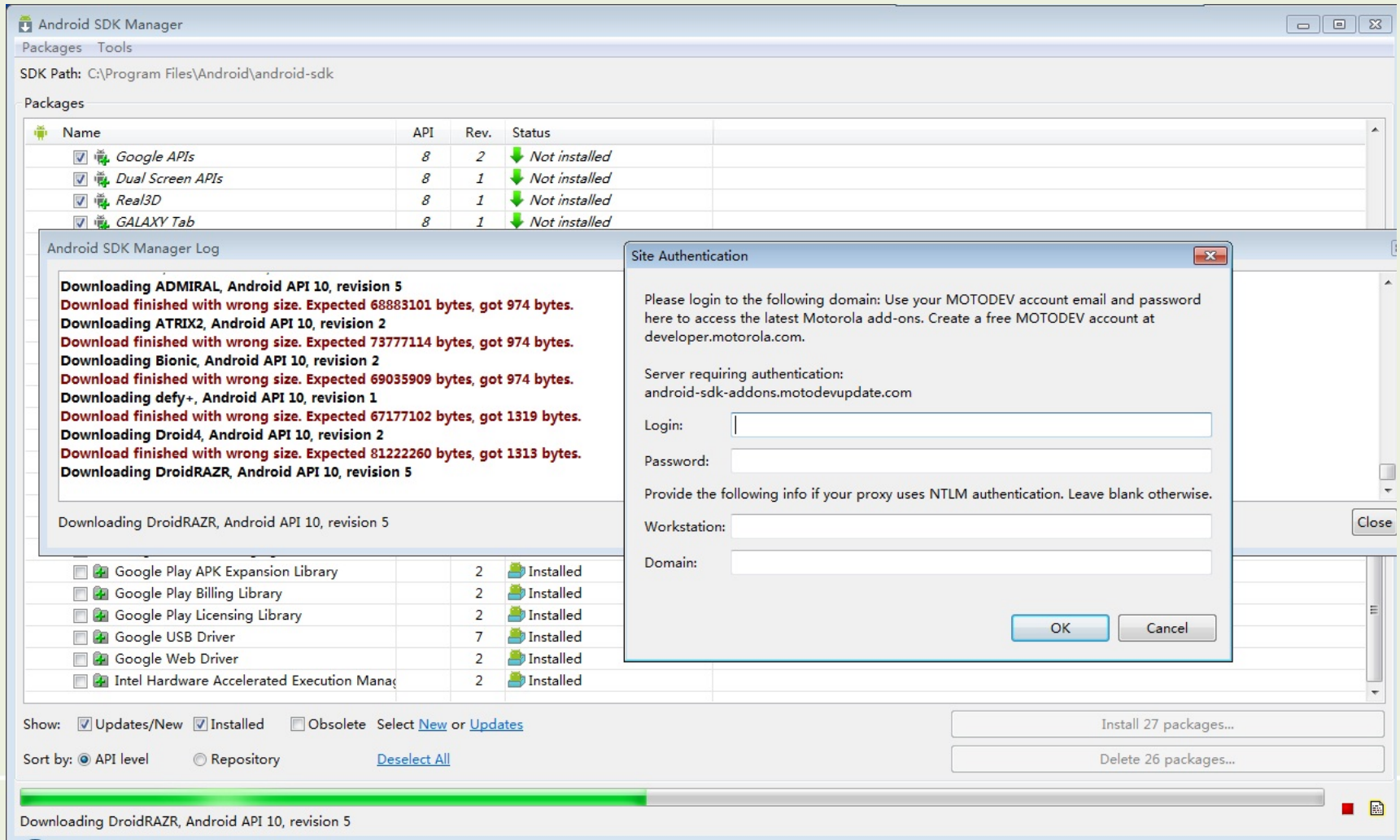


点击“Accept All”，接着点击“Install”，然后开始漫长的下载等待.....

管理升级Android SDK

■ 管理、升级Android SDK

◆ 如果下载过程中出现注册提示，点击Cancel（取消）



管理升级Android SDK

■ 注意事项:

- ◆ 由于国家对国外部分网站有限制，在下载升级ADT、Android SDK及访问android官网的过程中如果出现无法访问的情况。请换成教育网或者使用国外代理，方法：

The screenshot shows a Baidu search result for the query "国外代理". The search results include a link to "www.proxyie.cn/" dated 2012-9-7. A red box highlights the search results section. Another red box highlights a specific link in the results: "2012-10-19 国外IP代理服务器列表 美国VPN代理地址".

国外代理 - Powered by EmpireCMS

国外代理 2012-9-4 美国代理IP, 英国IP代理服务器列表 2012-09-04 2012-9-4
址 英国、巴西IP代理服务器 2012-09-03 2012-8-25 全球透明IP代理...
www.sooip.cn/guowaidaili/ 2012-9-7 - 百度快照

2012-10-19 国外IP代理服务器列表 美国VPN代理地址

2012-10-12 外国代理IP服务器 国外VPN代理列表

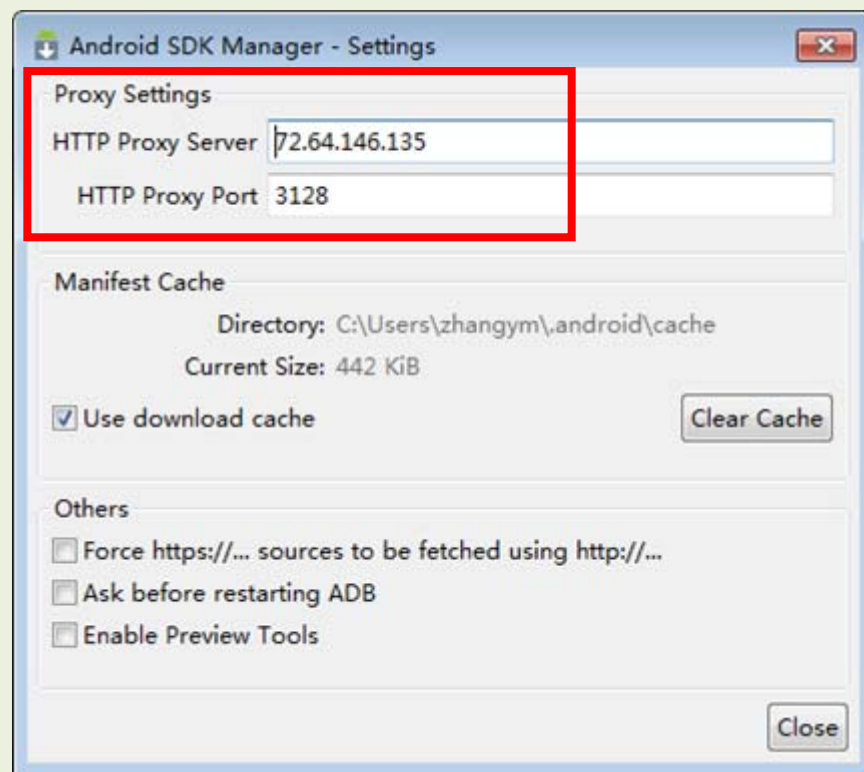
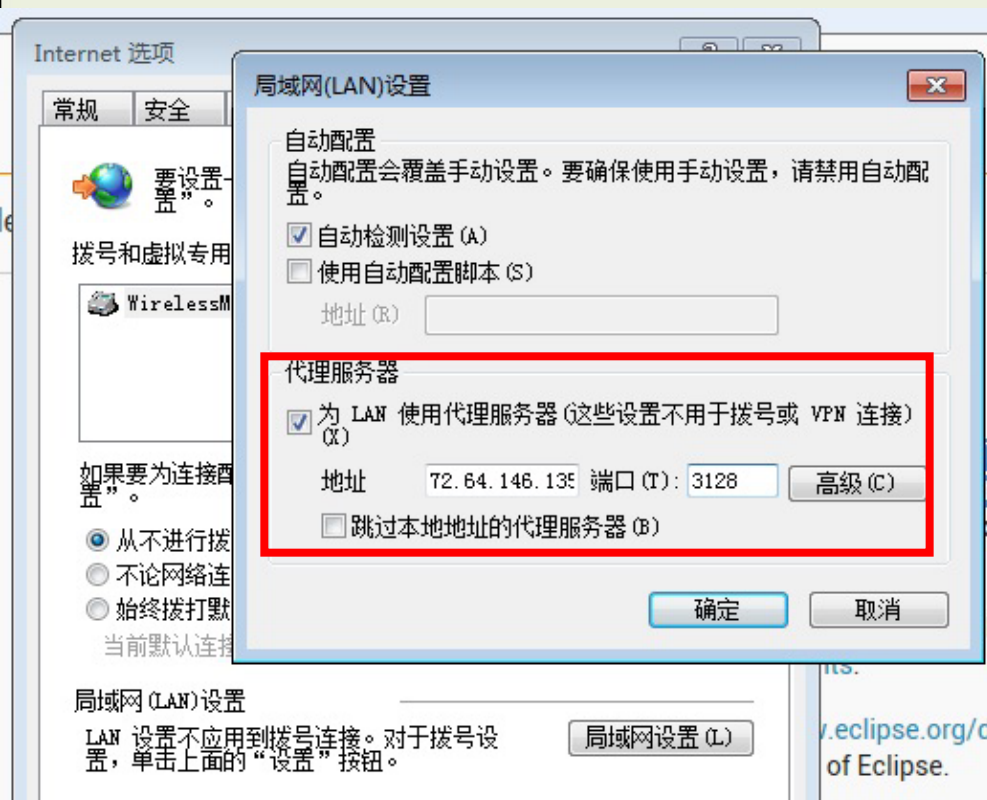
2012-10-9 国外VPN代理IP汇集 美国、巴西IP代理服务器

2012-9-30 国外代理IP地址 美国日本IP代理服务器

2012-9-25 国外代理IP 外国VPN代理服务器列表

管理升级Android SDK

■ 注意事项:



IE浏览器代理设置

Android SDK Manager
代理设置

创建AVD

- AVD的全称是Android Virtual Device。AVD是一个经过配置的Android模拟器（Android Emulator）



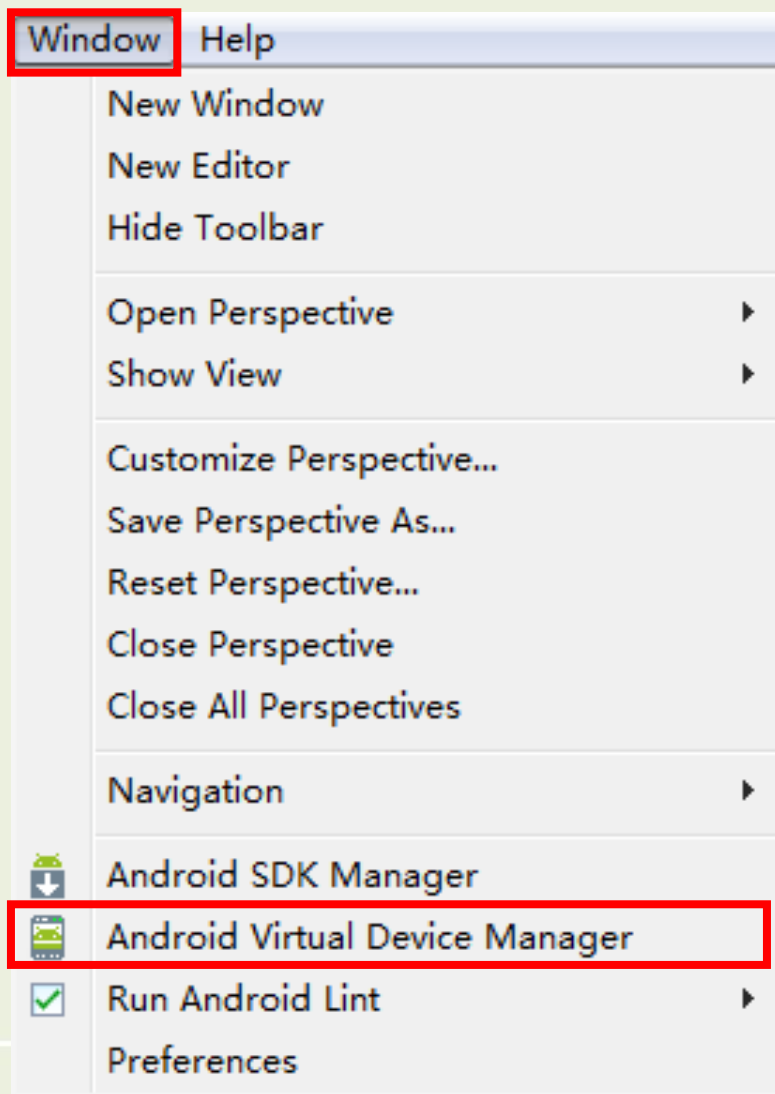
- Android Emulator能做什么？
 - 可模拟电话本、通话等功能
 - 内置的浏览器和Google Maps都可以联网
 - 可以使用键盘输入
 - 可单击模拟器按键输入
 - 可以使用鼠标单击、拖动屏幕进行操作

•Android Emulator和真机的不同之处

- 不支持呼叫和接听实际来电，但可以通过控制台模拟电话呼叫(呼入和呼出)。
- 不支持USB连接。
- 不支持相机/视频捕捉。
- 不支持音频输入(捕捉)，但支持输出(重放)。
- 不支持扩展耳机。
- 不能确定连接状态。
- 不能确定电池电量水平和交流充电状态。
- 不能确定SD卡的插入/弹出。
- 不支持蓝牙。

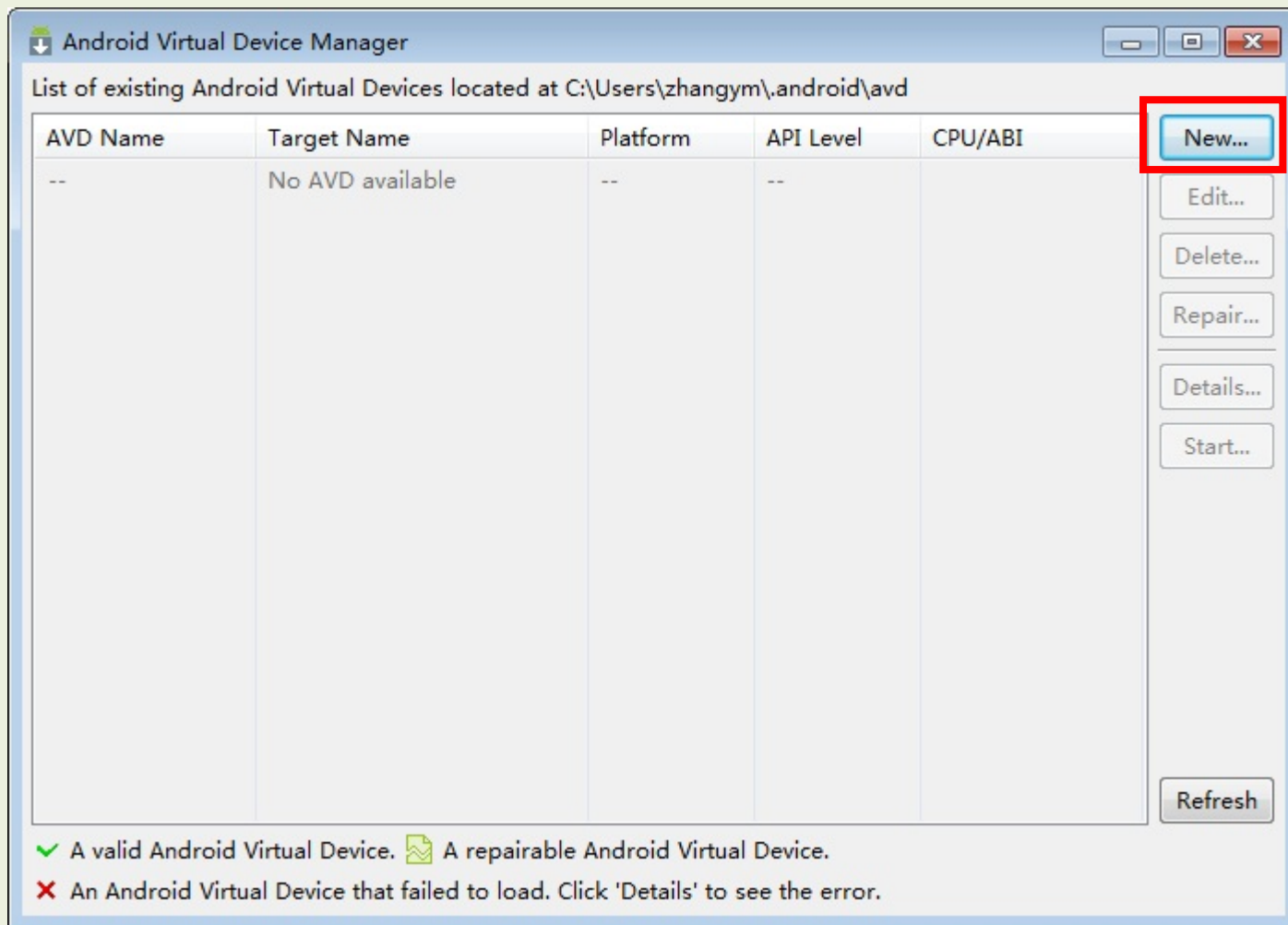
创建AVD

- 打开Eclipse，选择Window—>Android Virtual Device Manager



创建AVD

- 在打开的AVD管理器窗口中，选择“New”按钮



创建AVD

- 填入“Name”，选择“Target”的API，“SD Card”的“Size”大小随意，设置“Skin”屏幕大小，“Hardware”就保持默认值。
- 注意：这些值的设置虽然比较随意，但如果设置不当会对模拟器启动速度以及程序调试带来影响。请参考实际的手机配置来设置。设置好后也可以修改。

Edit Android Virtual Device (AVD)

Name: Test1

Target: Android 2.3.3 - API Level 10

CPU/ABI: ARM (armeabi)

SD Card:

☒ Size: 256 ☐ File: Browse...

☐ MiB

Snapshot: ☐ Enabled

Skin:

☒ Built-in: WQVGA400

☐ Resolution: x

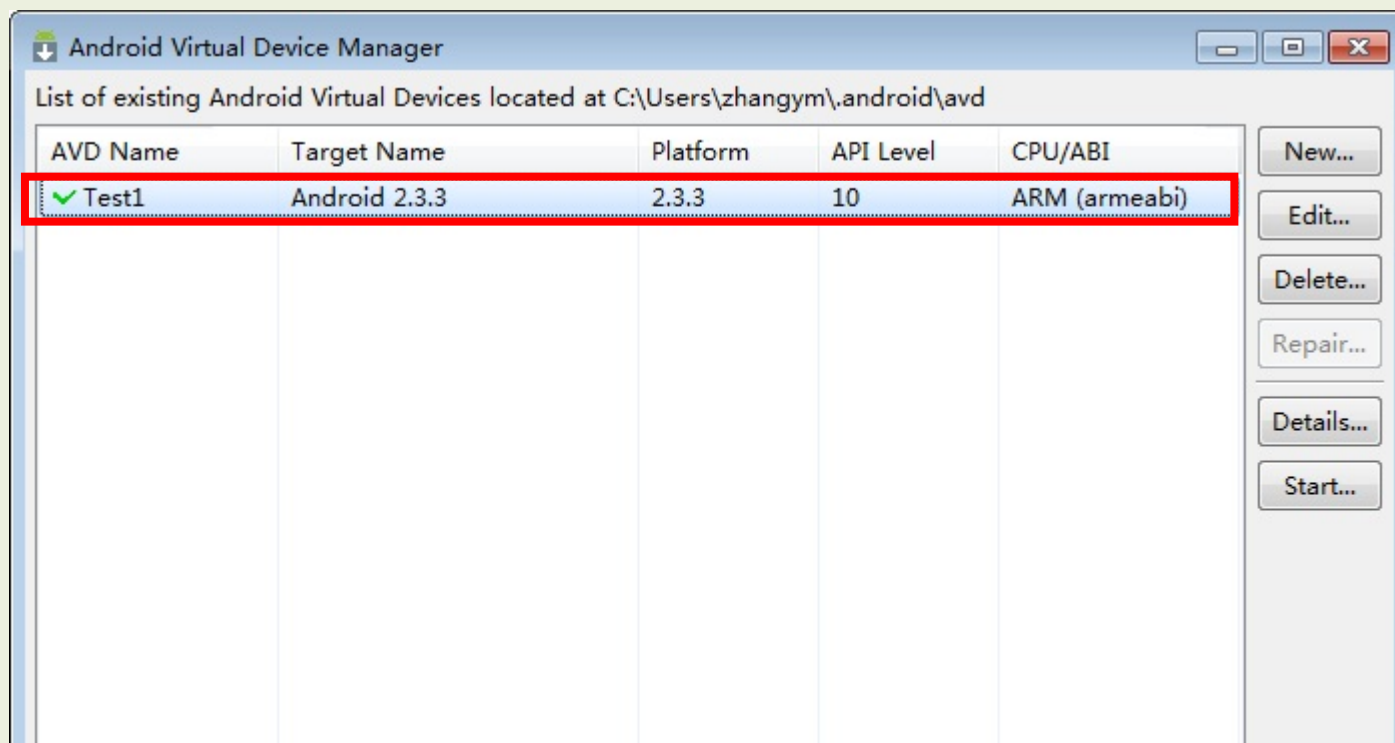
Hardware:

Property	Value	
Abstracted LCD density	120	
Max VM application h...	24	
Device ram size	256	

☐ Override the existing AVD with the same name

创建AVD

- 创建好的AVD模拟器在列表中显示出来了



到此步为止，Android开发前的准备工作全部完成！

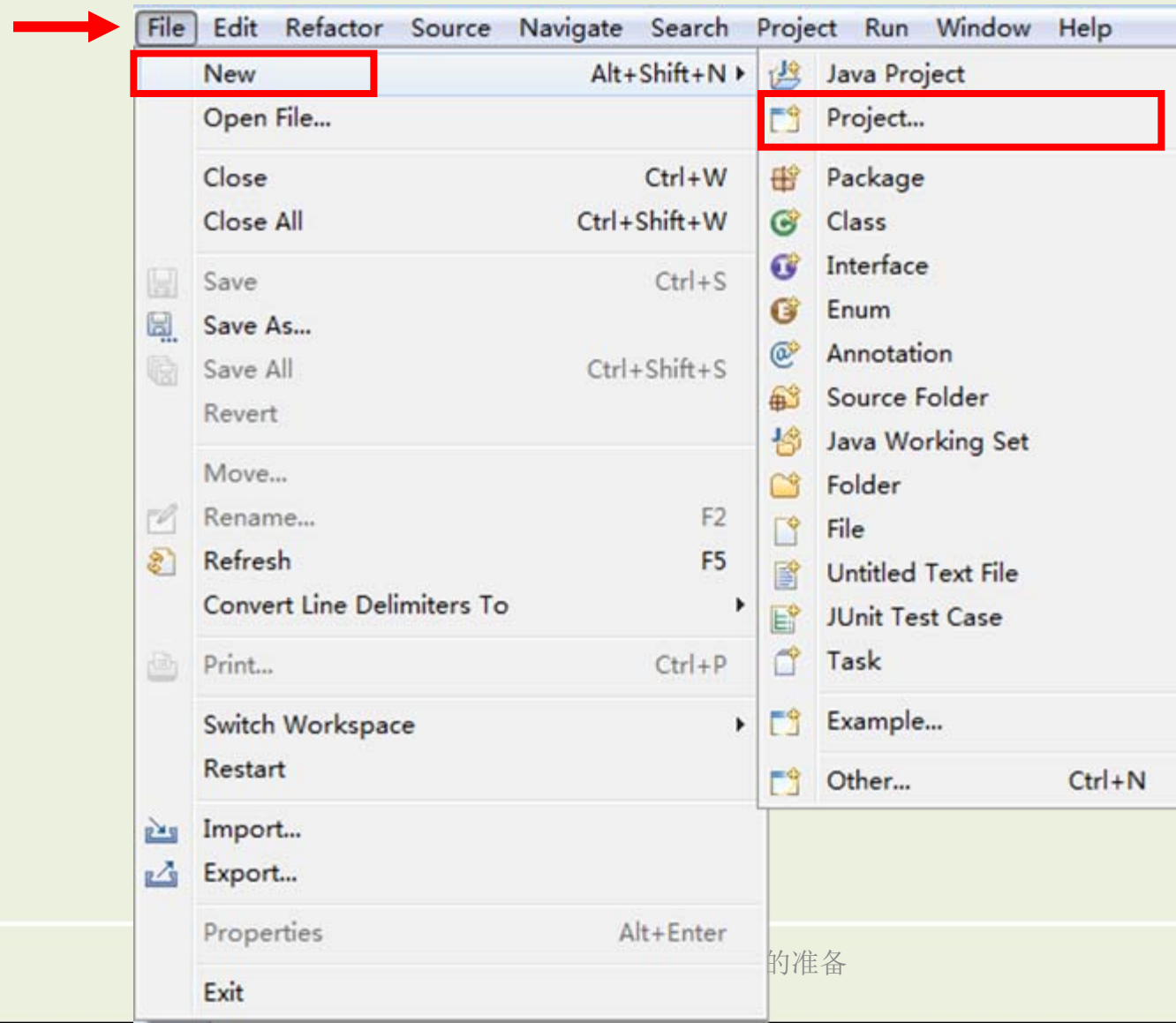
第一个Android程序： HelloWorld

本部分主要内容

- 建立HelloWorld工程
- 运行HelloWorld
- 真实设备上运行HelloWorld
- Android应用程序目录结构
- AndroidManifest.xml文件的作用
- HelloWorld布局文件
- HelloWorld Activity

建立HelloWorld工程

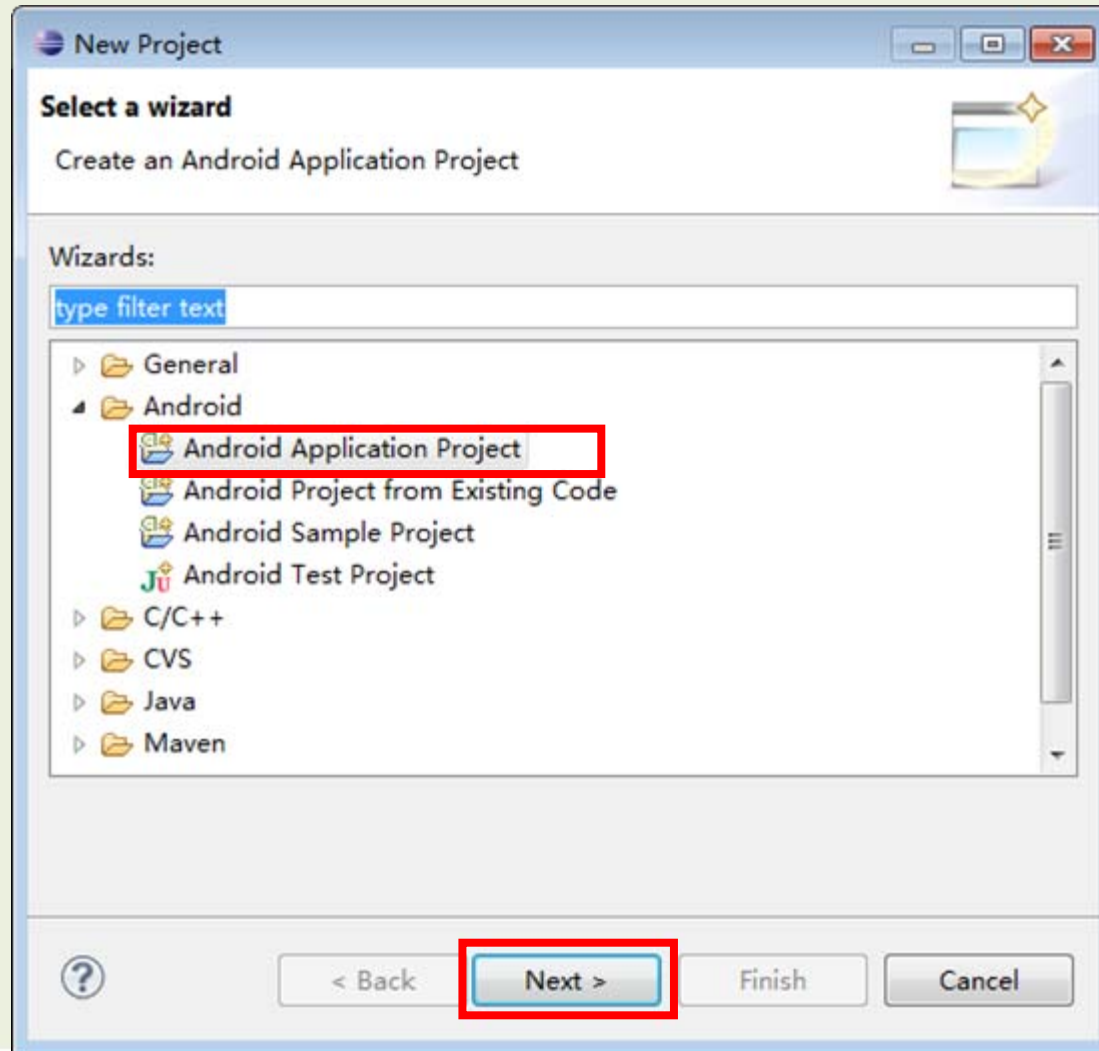
■ 打开Eclipse，新建项目（点击File→New→Project），如图：



的准备

建立HelloWorld工程

- 在项目列表中展开Android目录，选择 Android Application Project



建立HelloWorld工程

- 在第一红色框中填入名称信息，在第二红色框中选择版本

New Android Application

New Android Application

Enter an application name (shown in launcher)

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

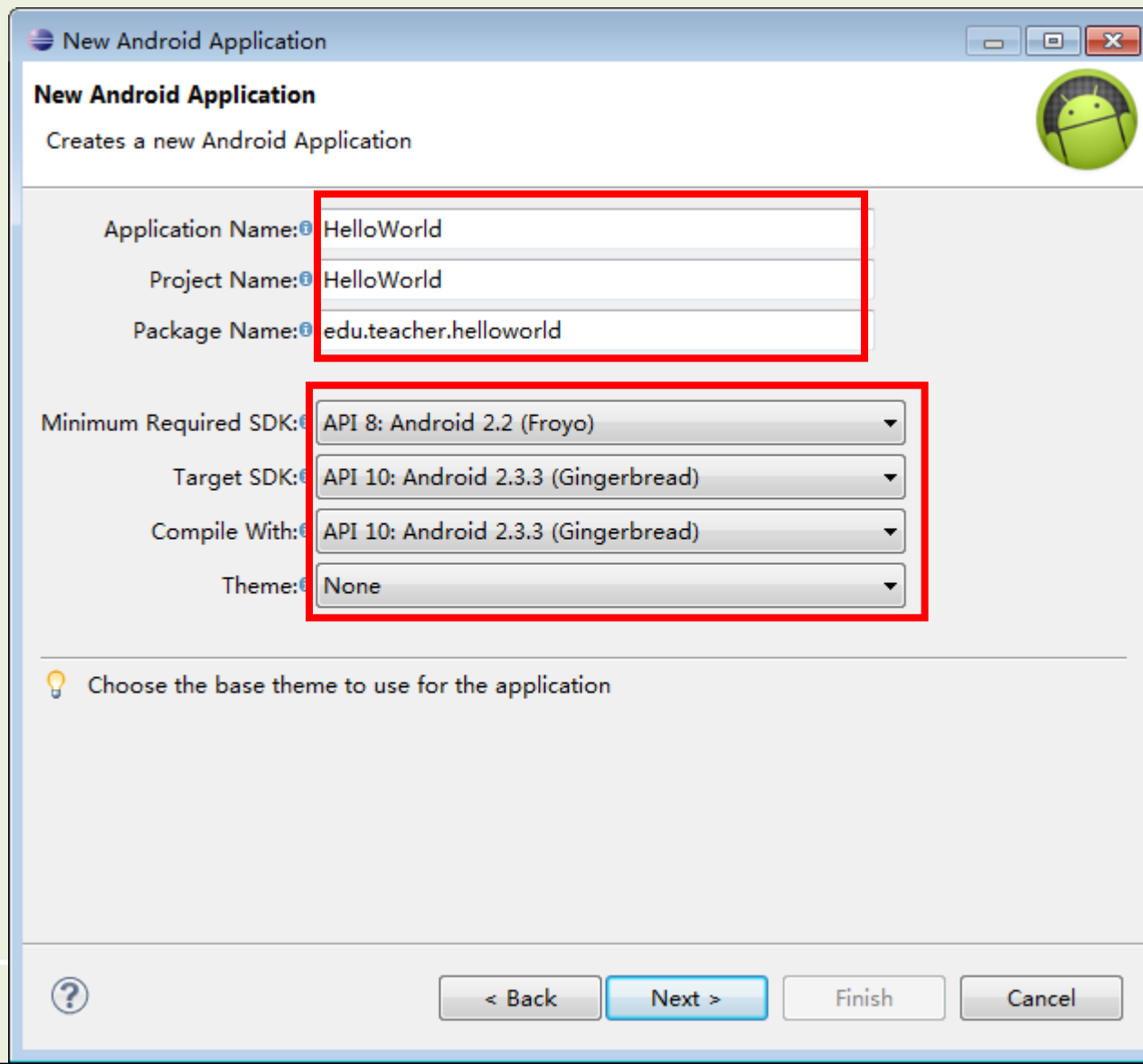
Theme:

< Back Next > Finish Cancel

- **Application Name:**
应用名称
- **Project Name:** 项目名称
- **Package Name:**
包名称
- 后面几项是选择此应用支持的 **Android SDK版本** 等

建立HelloWorld工程

- 根据实际情况填写和选择，设置好后单击“Next”进入下一步



New Android Application

New Android Application

Creates a new Android Application

Application Name: HelloWorld

Project Name: HelloWorld

Package Name: edu.teacher.helloworld

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 10: Android 2.3.3 (Gingerbread)

Compile With: API 10: Android 2.3.3 (Gingerbread)

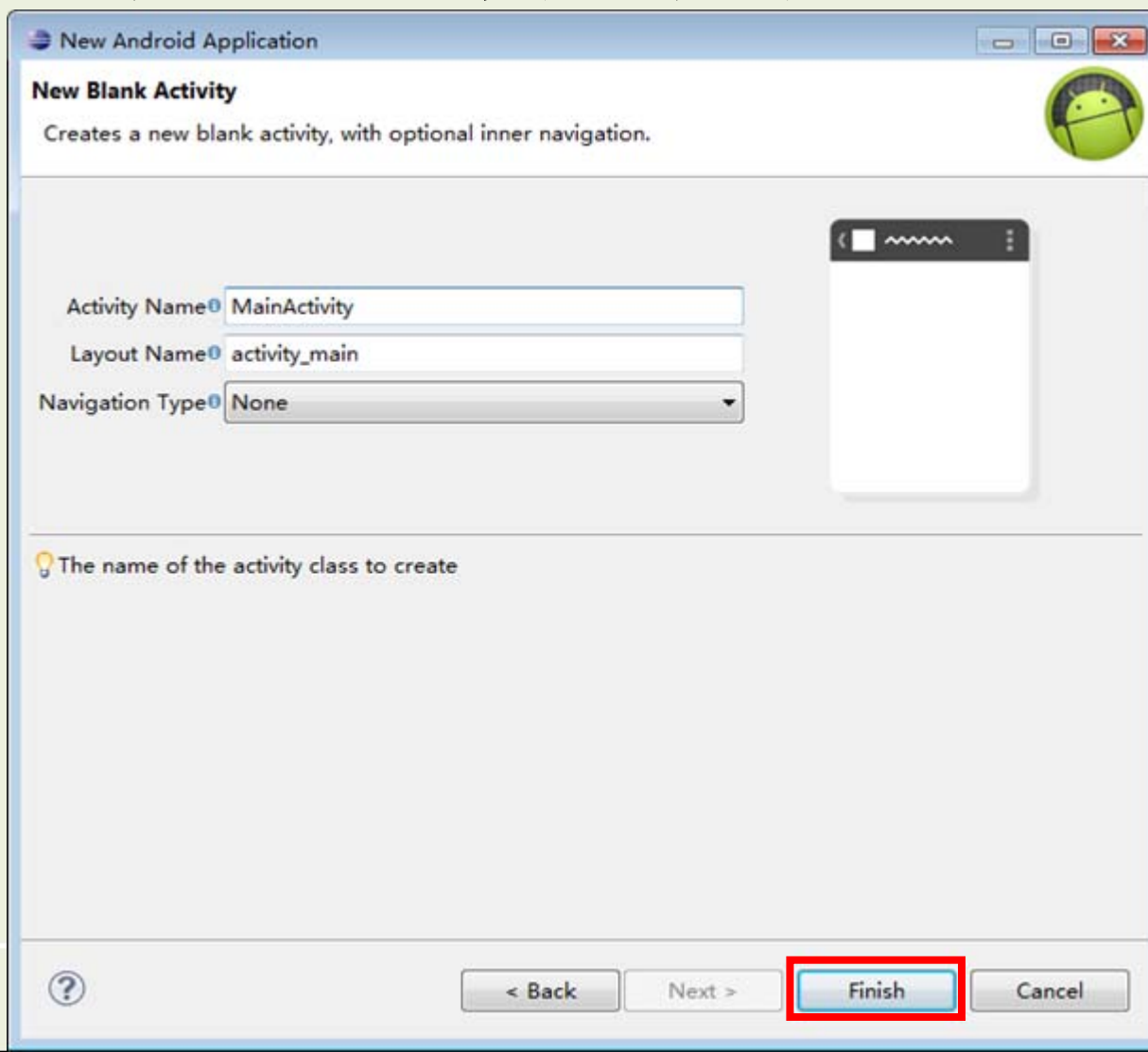
Theme: None

Choose the base theme to use for the application



< Back Next > Finish Cancel

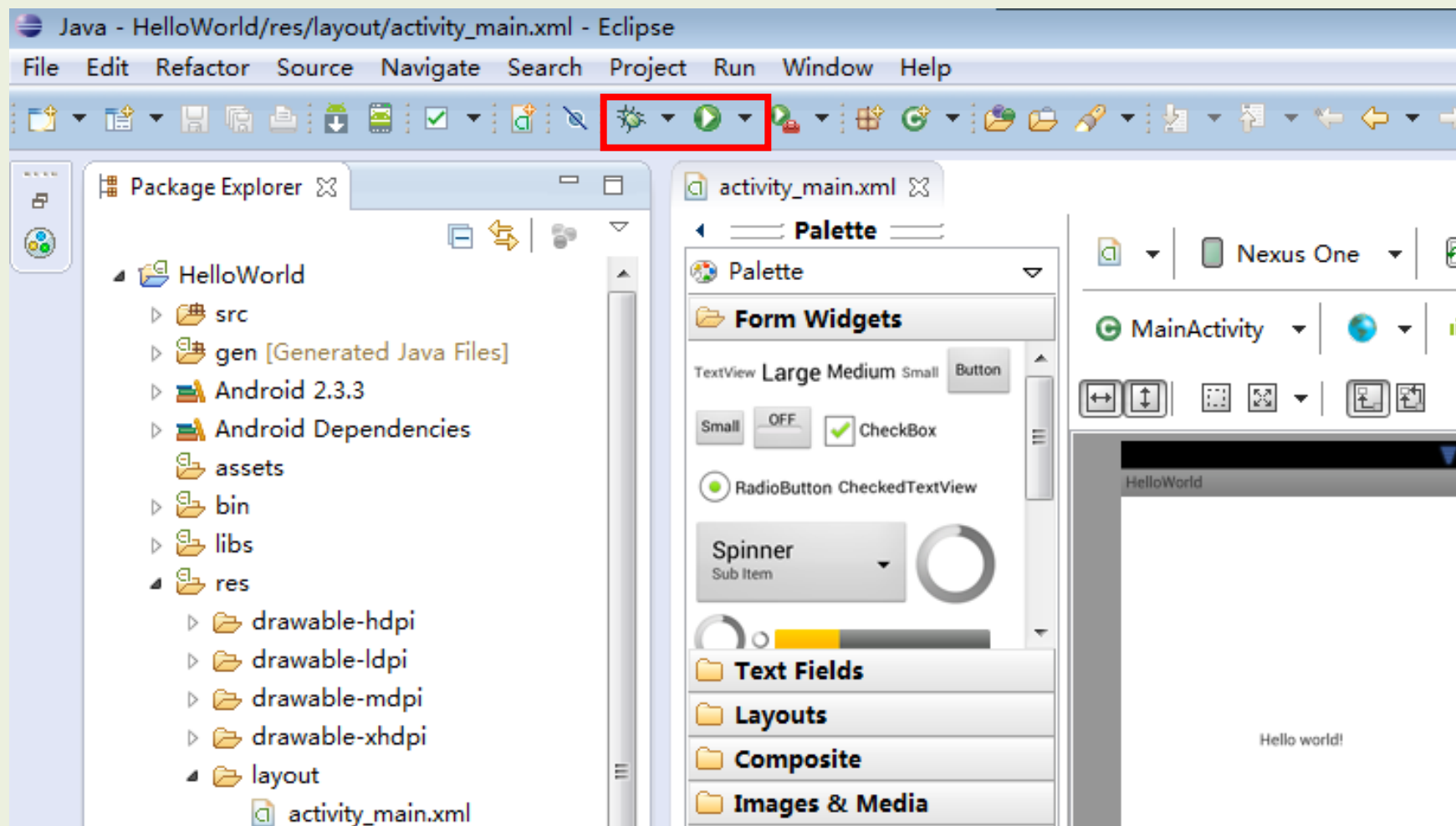
建立HelloWorld工程

- 接下来的几个窗口直接单击“Next”选择默认值，直到最后一个窗口，单击“Finish”，完成工程创建



运行HelloWorld

■ 选择Debug  或者Run  工具栏按钮启动程序



运行HelloWorld

■ 模拟器执行画面



简单修改

■ 修改前的strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">HelloWorld </string>
    <string name="hello_world">Hello World!</string>
    <string name="menu_settings">Settings</string>

</resources>
```

■ 修改后的strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">第一个Android应用</string>
    <string name="hello_world">你好，工程2010班的同学们！</string>
    <string name="menu_settings">Settings</string>

</resources>
```


运行HelloWorld

■ 模拟器执行画面（strings.xml修改后）



真实设备上运行HelloWorld

- 要在真实设置上（以手机为例）调试或者运行Android应用，需要设置
 - 安装手机的Windows下的USB驱动程序。可通过“豌豆荚”软件自动搜索安装
 - 打开手机的USB调试功能。选择手机中的“**设置**”功能，在如图所示项中进行修改：

①



②

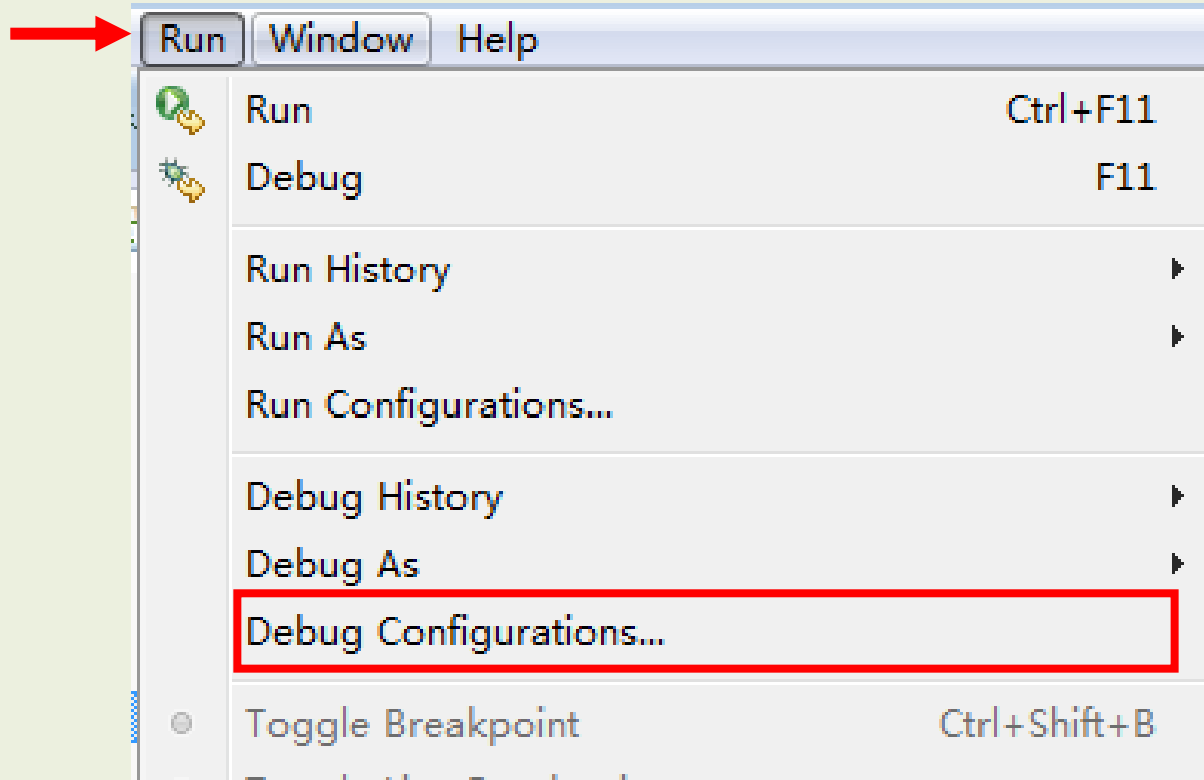


③



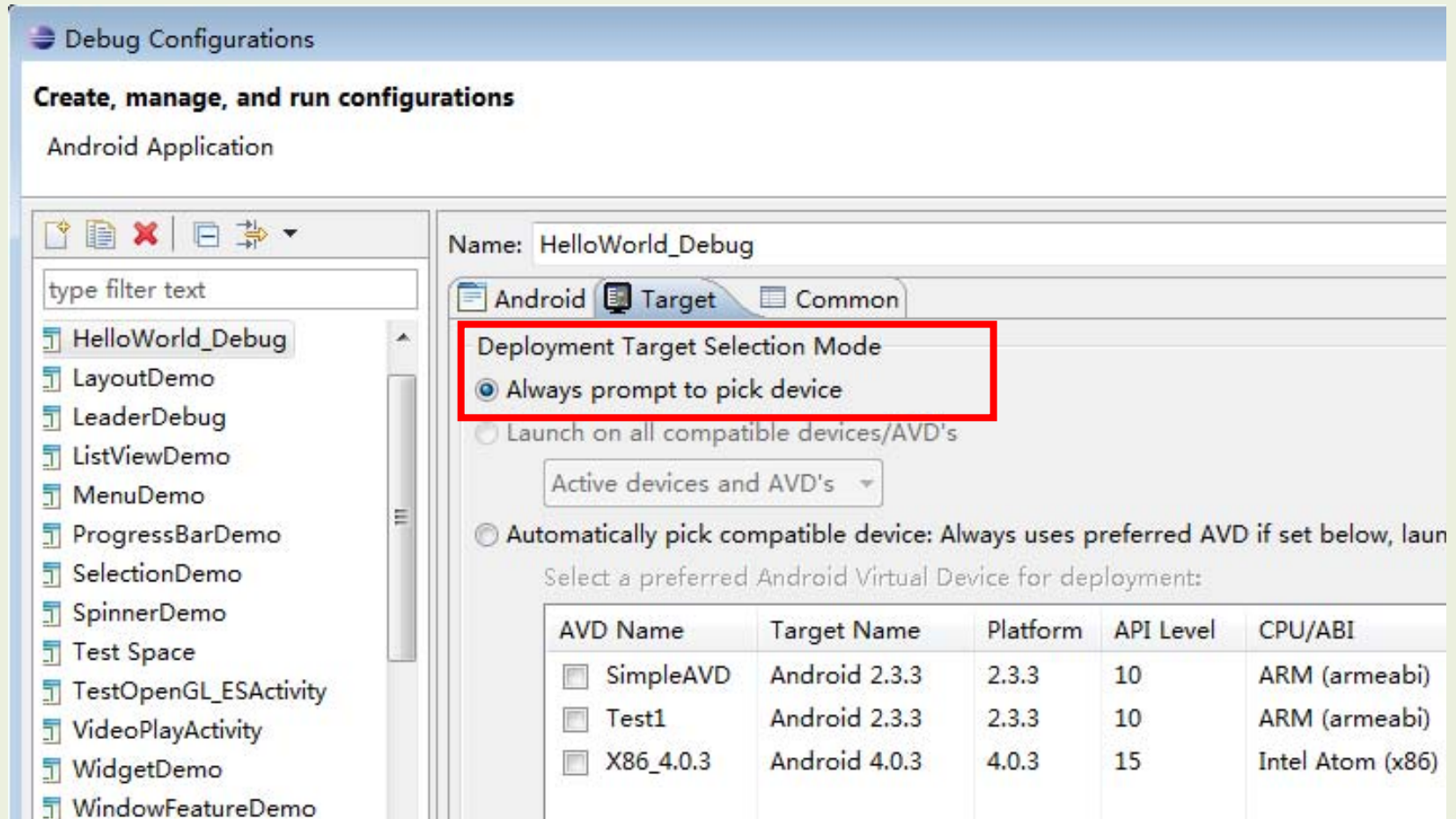
真实设备上运行HelloWorld

- 在Eclipse环境中进行配置（设置运行模拟器的选项）



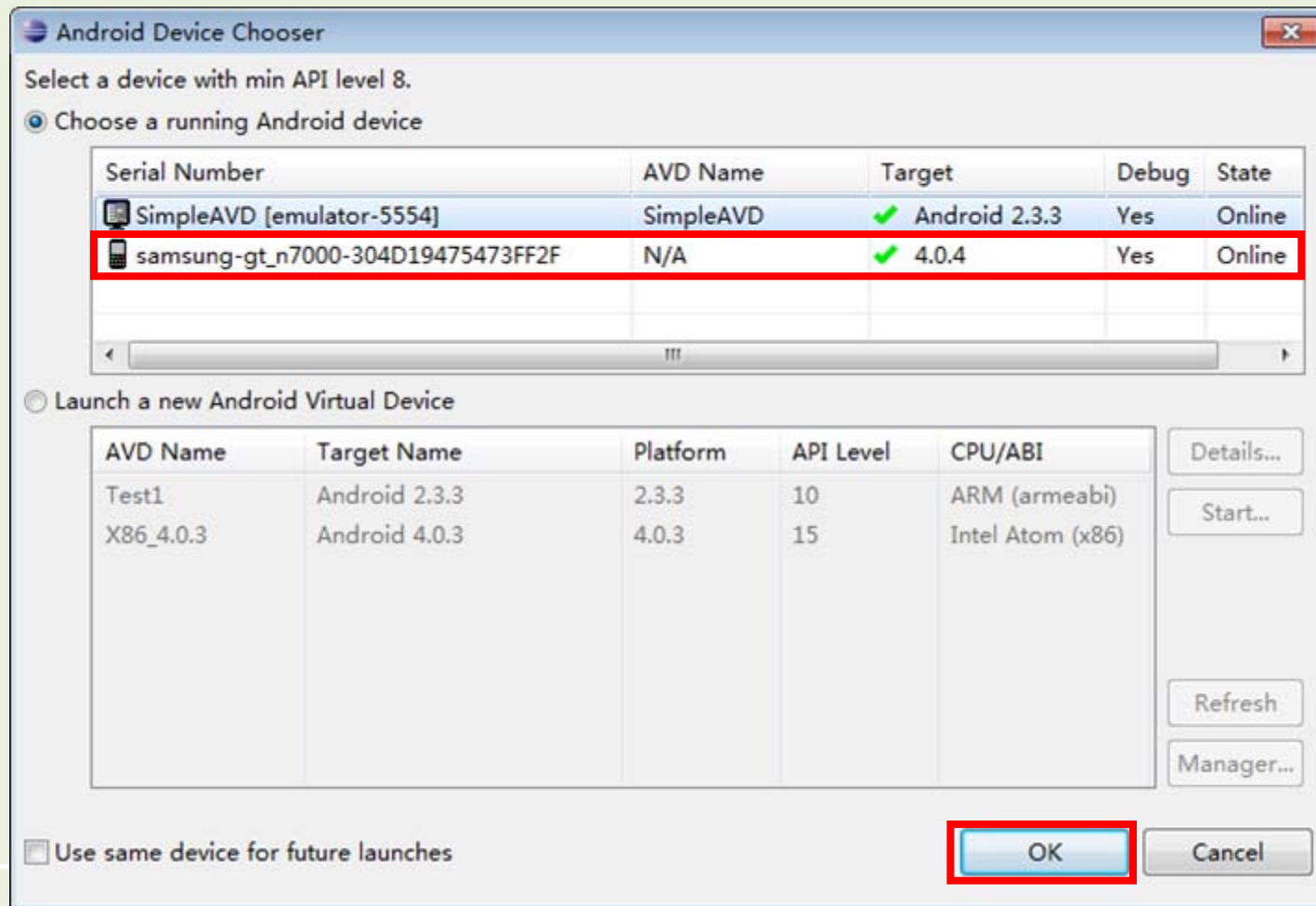
真实设备上运行HelloWorld

- 在Eclipse环境中进行模拟器环境设置。如图。设置好后点击“Apply”按钮



真实设备上运行Hello World

- 把手机通过USB数据线连接电脑，在Eclipse中启动HelloWorld项目，会出现选择Android Device的窗口，选择相应手机设备名后点击“OK”按钮，然后在手机中查看运行效果



真实设备上运行Hello World

- 此例在手机上的运行效果和模拟器上效果相似。截图略

Android应用程序目录结构

HelloWorld

src

Java源文件目录

edu.teacher.helloworld

MainActivity.java

gen [Generated Java Files]

edu.teacher.helloworld

R类, 资源管理文件

BuildConfig.java

R.java

Android 2.3.3

Android Dependencies

assets

任意资源文件

bin

libs

res

图片资源文件

drawable-hdpi

ic_launcher.png

drawable-ldpi

drawable-mdpi

drawable-xhdpi

布局资源文件

layout

activity_main.xml

menu

activity_main.xml

values

应用程序基础
信息配置文件

strings.xml

styles.xml

字符串、颜色、
尺寸、类型、主题.....
资源文件

项目属性
配置文件

AndroidManifest.xml

proguard-project.txt

project.properties

Android应用程序目录结构

■ 额外说明:

- gen: 存放编译器自动生成的java代码; 这个目录下的文件是系统自动维护的; 不要手动编辑这些代码
- Android 2.3.3: android的jar包
- assets: 存放资源文件, 不会在R.java中生成ID
- res: 存放资源文件, 会在R.java中自动生成资源的ID
 - drawable-~~X~~dpi系列文件夹, 包含四个大小版本的图片
 - layout: 以后经常打交道的目录。存放布局文件, 用于界面显示。每一个Activity都有一个对应的布局文件 (xml文件)
 - values: string.xml文件是页面中文字保存的地方。还包含一些其它文件, 都是应用程序中需要用到的值, 会在R.java中生成ID
- **AndroidManifest.xml**: app的主配置文件。配置application、activity及其相关属性等信息。例如: 当你新创建了Activity类, 就要在这个文件中配置

AndroidManifest.xml文件的作用

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.teacher.helloworld"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="10" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="edu.teacher.helloworld.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

应用的包名

项目sdk版本

整个应用的图标

整个应用的标签

要配置的Activity类

Activity标题

配置Activity

配置Activity过滤器

整个应用启动时的activity

HelloWorld布局文件

- 在建立项目时，系统会自动生成一个布局资源文件
 - 在原来的老版本中，默认文件名为`main.xml`
 - 在现在的新版本中，在建议工程时可以自己命名，缺省文件名是`activity_main.xml`
- 窗口布局资源文件示例如下页所示
 - 需要注意的是，在xml文件中，用`<!-- -->`来注释代码
 - 在版本和编码方式声明之后，是具体的界面元素，首先是布局方式。`LinearLayout` 代表线性布局，而`TextView`代表包含在线性布局中的文本框
 - Android中要注意一点，在对空间的变量设置值的时候，需要用双引号将值括住

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#aaffff"
>
    <TextView
        android:id="@+id/TextView_1"
        android:layout_width="100px"
        android:layout_height="wrap_content"
        android:background="#bbffff"
        android:text="@string/name1"
    />
    <TextView
        android:id="@+id/TextView_2"
        android:layout_width="@dimen/size1"
        android:layout_height="wrap_content"
        android:background="@color/red_color"
        android:text="@string/name2"
    />
</LinearLayout>
```

■ 窗口布局资源文件在JAVA中使用：

- 用法：R.layout.布局文件前缀名
- 范例： setContentView(R.layout.main);

```
public void onCreate(Bundle savedInstanceState){  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
}
```

HelloWorld Activity

Activity: HelloWorld类

```
public class HelloWorld extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState); //构造函数  
        setContentView(R.layout.main); //与xml相关联  
    }  
}
```

在这个大的函数里面有个onCreate函数，就相当于C/C++中的main函数；super.onCreate表示调用父类的构造函数；setContentView表示将.java文件和.xml界面相关联，也就是将后台代码和前台界面相关联。