

•安卓(Android)程序开发初级教程•
【视频+教程+源码】

视频区：

Android 开发经典视频教程合集

本视频教程由初级入门篇和实战篇结合而成，入门篇是本人收集网上反响比较好，很适合学者自学的视频教程，大家可以针对自己的风格选择个别教程学习，实战篇是汇集各个实战项目讲解的视频教程，大家可以挑选自己喜欢的项目，进行练手。排名不分先后，本资源会不定期更新。

| | |
|-----|---|
| 初级篇 | 视频教程 |
| | Android 游戏开发视频教程(Mars 老师) |
| | Java 4 Android 基础视频教程 |
| | 如何快速掌握 Android 开发技术系列【51CTO 技术公开课】 |
| | 轻松几步学 Android 开发视频教程 |
| | Android 开发视频教程《Java 语言基础》 |
| | 4 天走进 Android 开发的世界 |
| | Android 从程序员到架构师之路【共享版】 |
| | Android 技术轻松入门课程 |
| | Android 企业级应用开发系统课程-0 基础学习 Android 就是这么简单 |
| | Android 游戏开发--Are you Ready ? |
| | Android 高级应用开发-基础篇 |
| | Android 嵌入式底层开发【张凌华课程】 |
| | Android 深入浅出视频教程【51CTO 学院】 |
| | Android 开发 AsyncTask 的使用【孙炜】 |
| | Android 系统中通知栏的应用【阎赫】 |

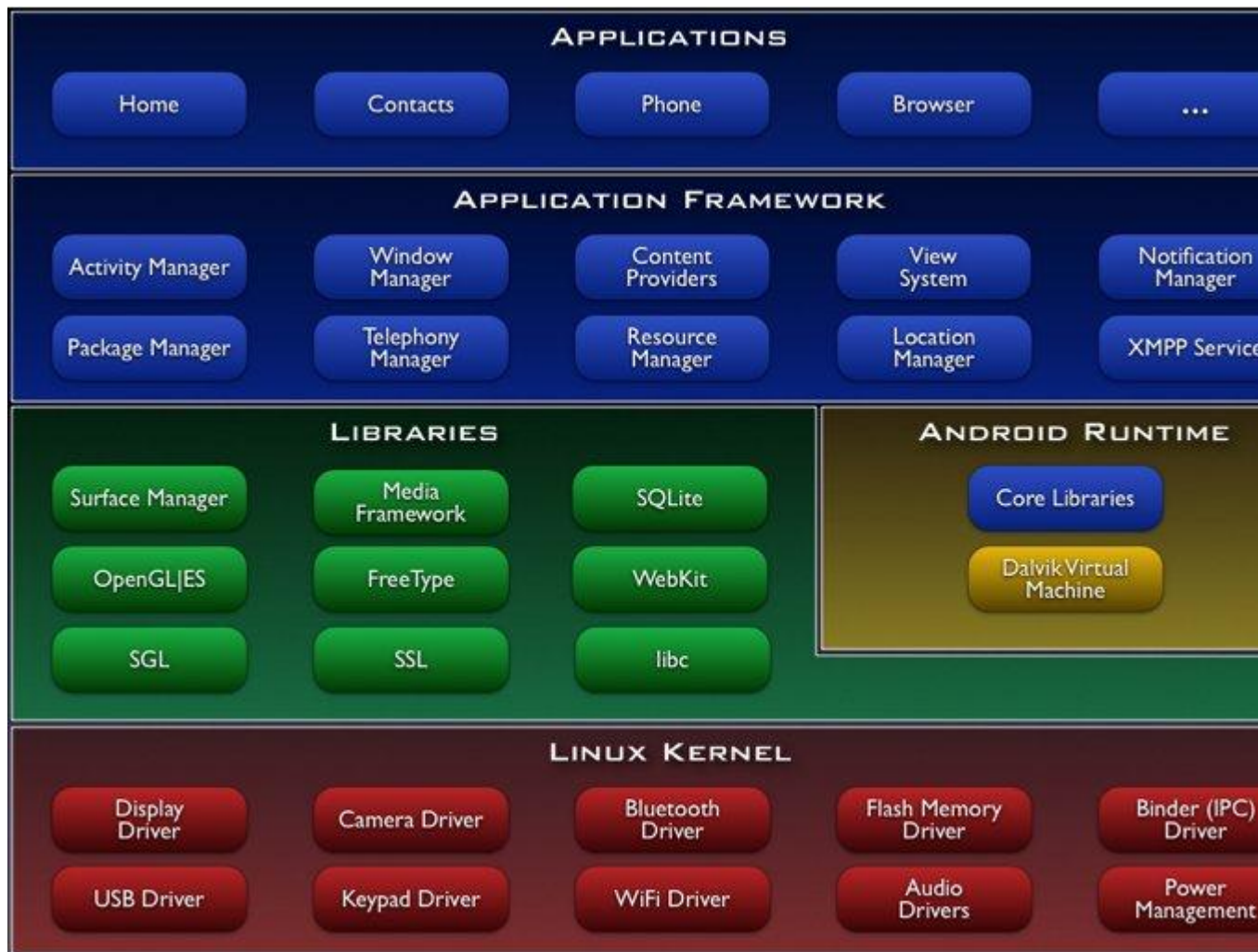
| | |
|-----|---|
| | Android 开发高级应用第一季 |
| | Android 开发高级应用第二季 |
| | Android 应用开发视频教程-第 2 季 |
| | Android 高级应用 1-java、android 起步 |
| | Android 高级应用 2-Eclipse IDE 使用提高 （ 7 课时 ） |
| | Android 高级应用 3-Android 编码风格 （ 7 课时 ） |
| | Android 高级应用 4-SVN 和 Git 版本控制 （ 16 课时 ） |
| | Android 高级应用 6-总观 Android 全局 （ 7 课时 ） |
| | Android 高级应用 7-简单了解 Android 程序制作 （ 13 课时 ） |
| | Android 高级应用 8-Android 的面孔 Activity （ 32 课时 ） |
| | Android 高级应用 9-Android 之 Service 详解 （ 14 课时 ） |
| | Android 高级应用 10-Android 之 UI 编程 （ 66 课时 ） |
| 实战篇 | Android 从零到实战 |
| | 疯狂软件新浪微博 Android 项目案例视频教程 |
| | Android 开发视频教程《游戏数独》(Mars 老师) |
| | Android 应用实战:淘宝网手机客户端全程实录(第一季) |
| | Android 开发视频教程--腾讯微博项目 |
| | Android 实现多款 3D 游戏 |
| | 微信开放平台-Android 应用接入视频教程 |
| | 杀毒软件的基本开发原理及实现 |
| | Android 开发 自定义未捕获异常 |
| | 大话优酷、酷六类视频网站 Android 客户端开发 |

| | |
|--|--|
| | Android 自定义控件与自定义动画实战【张科勇】 |
| | Android 开发：播放 sdcard 中的 mp3 文件 |
| | 像极客一样制作 Android ROM【李宁】 |
| | Android JNI 机制实现 java 调用 C 函数返实例【倪键树】 |
| | Android 开发 隐式 Intent 讲解【李韩飞】 |
| | Android 网络游戏开发-实战项目-一步一步教你开发 Android 网络游戏 |

教程(一)

平台简介

令人激动的 Google 手机操作系统平台-Android 正式发布了，这是一个开放源代码的操作系统，内核为 Linux. 作为开发者，我们所关心的是这个平台的架构以及所支持的开发语言。下面是这个平台的架构模型：



这个平台有以下功能：

+ Application framework: 可重用的和可替换的组件部分，在这个层面上，所有的软件都是平等的。

+ Dalvik virtul machine: 一个基于 Linux 的虚拟机。

+ Integrated browser: 一个基于开源的 WebKit 引擎的浏览器，在应用程序层。

+ Optimized graphics: 包含一个自定义的 2D 图形库和基于 OpenGL ES 1.0 标准的 3D 实现。

+ SQLite: 数据库

+ Media support: 通用的音频, 视频和对各种图片格式的支持(MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

+ GSM Telephony: GSM 移动网络, 硬件支持。

+ Bluetooth, EDGE, 3G, and WiFi: 都依赖于硬件支持。

+ Camera, GPS, compass, and accelerometer: 都依赖于硬件支持。

+ Rich development environment: 包含一套完整的开发工具集, 方便跟踪调试, 内存检测和性能测试, 而且提供了 Eclipse 的插件。

下面我们就来亲身体验一下 Android 程序的开发之旅。

先介绍一下开发环境, 下面是对系统及相关软件的要求 :

操作系统 :

Windows XP or Vista

Mac OS X 10.4.8 or later (x86 only)

Linux (tested on Linux Ubuntu Dapper Drake)

Supported Development Environments

Eclipse

Eclipse 3.2, 3.3 (Europa)

Android Development Tools plugin (optional)

Other development environments or IDEs

JDK 5 or JDK 6 (JRE alone is not sufficient)

Not compatible with Gnu Compiler for Java (gcj)

Apache Ant 1.6.5 or later for Linux and Mac, 1.7 or later for Windows

我使用 Eclipse 3.3 + JDK 1.6. + Ant 1.7 的组合。还有两个重要的就是: Android SDK 以及 Android 用于 Eclipse 中的插件。

Android SDK 的下载链接: <http://code.google.com/android/>

如果你是第一次使用这些软件，请注意安装顺序和设置好环境变量。一般的顺序是先安装 JDK 然后 解压 ant 压缩包，然后设置 java 环境变量和 ant 环境变量，然后是解压 Android SDK，再设置 Android SDK 的环境变量。总之就是把 JDK, ANT, Android SDK 的路径添加到 path 里。

Android for eclipse plug in 在安装过程很简单，通过网络安装插件就可以了，这个是 URL: <https://dl-ssl.google.com/android/eclipse/>

具体的配置过程，可以查看：

<http://code.google.com/android/intro/installing.html#otherides>

以下为一个 hello Android 的开发步骤：

创建一个项目：

创建一个新项目是很简单的，只要你安装了 Eclipse 插件，并且你的 Eclipse 软件版本在 3.2 或 3.3，你就可以开始开发了。

首先，看一下要创建"Hello, World"程序从高级层面上有哪些步骤：

1, 通过 File -> New -> Project 菜单，建立新项目"Android Project"

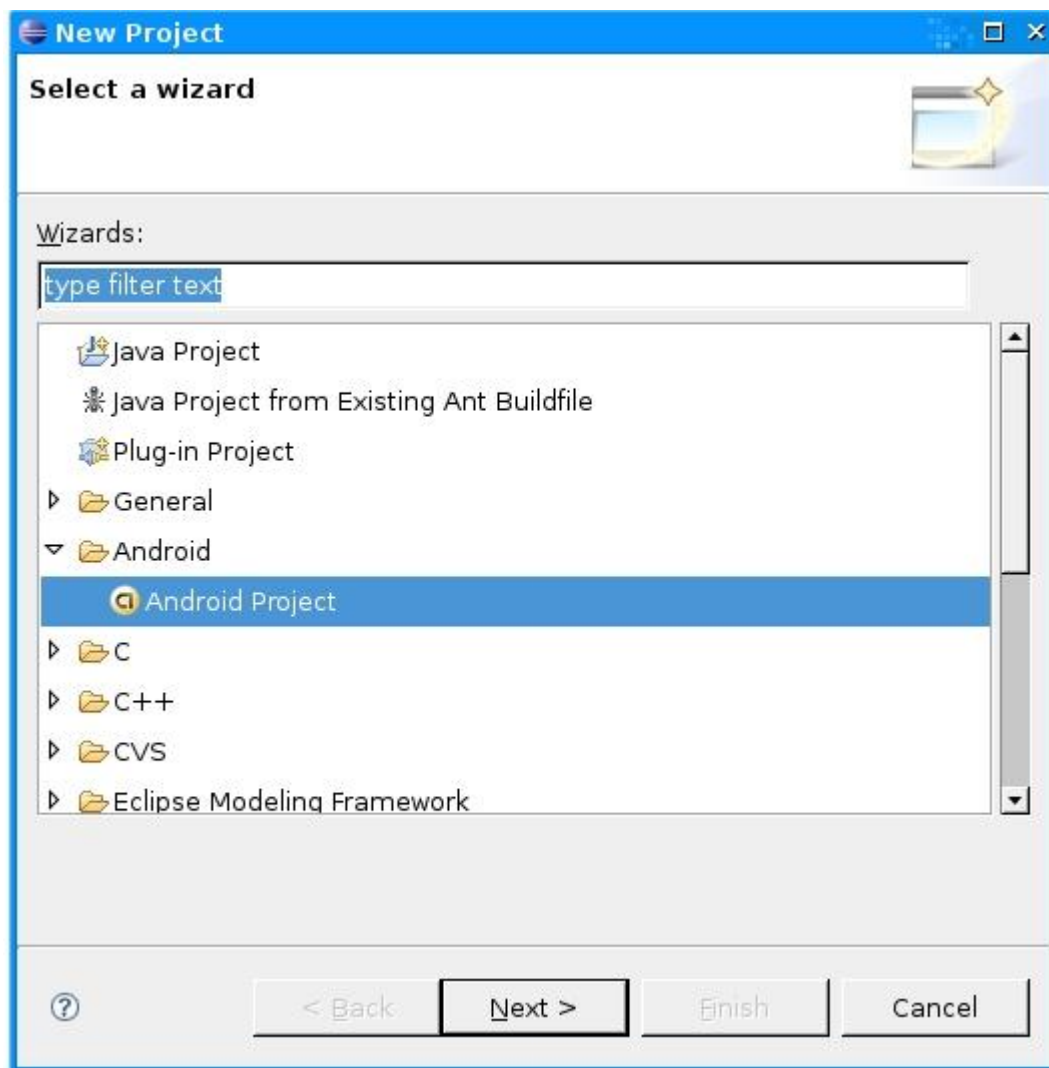
2, 填写新项目各种参数。

3, 编辑自动生成的代码模板。

尽此而已，我们通过下面的详细说明来完成每个步骤。

1, 创建一个新的 Android 项目

启动 Eclipse, 选择 File -> New -> Project 菜单, 如果你安装好了 Android 的 Eclipse 插件，你将会在弹出的对话框中看到"Android Project" 的选项。



选择 “Android Project” ,点击 Next 按钮。

2, 填写项目的细节参数.

下面的对话框需要你输入与项目有关的参数：

New Android Project
Creates a new Android Project resource.

Project name:

Contents

- ☒ Create new project in workspace
- ☐ Create project from existing source
- ☒ Use default location

Location:

Properties

Package name:

Activity name:

Application name:

? < Back Next > Finish Cancel

这个表格中详细介绍了每个参数的含义:

Project Name: 包含这个项目的文件夹的名称。

Package Name: 包名，遵循 JAVA 规范，用包名来区分不同的类是很重要的，例子中用到的是"com.google.android",你应该按照你的计划起一个有别于这个的路径的名称。

Activity Name: 这是项目的主类名，这个类将会是 Android 的 Activity 类的子类。一个 Activity 类是一个简单的启动程

序和控制程序的类。它可以根据需要创建界面，但不是必须的。

Application Name: 一个易读的标题在你的应用程序上。

在"选择栏"的 "Use default location" 选项，允许你选择一个已存在的项目。

3，编辑自动生成的代码。

当项目创建后，你刚才创建的 HelloAndroid 就会是包含下面的代码.

```
public class HelloAndroid extends Activity

{

    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle icle)

    {

        super.onCreate(icle);

        setContentView(R.layout.main);

    }

}
```

下面我们开始修改它

[构建界面]

当一个项目建立好以后，最直接的效果，就是在屏幕上显示一些文本，下面是完成后的代码，稍后我们在逐行解释。

```
public class HelloAndroid extends Activity {  
  
    /** Called when the activity is first created. */  
  
    @Override  
  
    public void onCreate(Bundle icle) {  
  
        super.onCreate(icle);  
  
        TextView tv = new TextView(this);  
  
        tv.setText("Hello, Android");  
  
        setContentView(tv);  
  
    }  
  
}
```

注意你还需要添加 `import android.widget.TextView;` 在代码开端处。

在 Android 程序中，用户界面是由叫做 Views 类来组织的。 一个 View 可以简单理解为可以绘制的对象，像选择按钮，一

个动画，或者一个文本标签(这个程序中)，这个显示文本标签的 View 子类叫做 TextView.

如何构造一个 TextView:

```
TexiView tv = new TextView(this);
```

TextView 的构造参数是 Android 程序的 Context 实例, Context 可以控制系统调用, 它提供了诸如资源解析, 访问数据库等

等。Activity 类继承自 Context 类，因为我们的 HelloAndroid 是 Activity 的子类，所以它也是一个 Context 类，所以我们能用"this"在 TextView 构造中。

当我们构造完 TextView 后，我们需要告诉它显示什么:

```
tv.setText("Hello, Android");
```

这个步骤很简单，当我们完成了这些步骤以后，最后要把 TextView 显示在屏幕上。

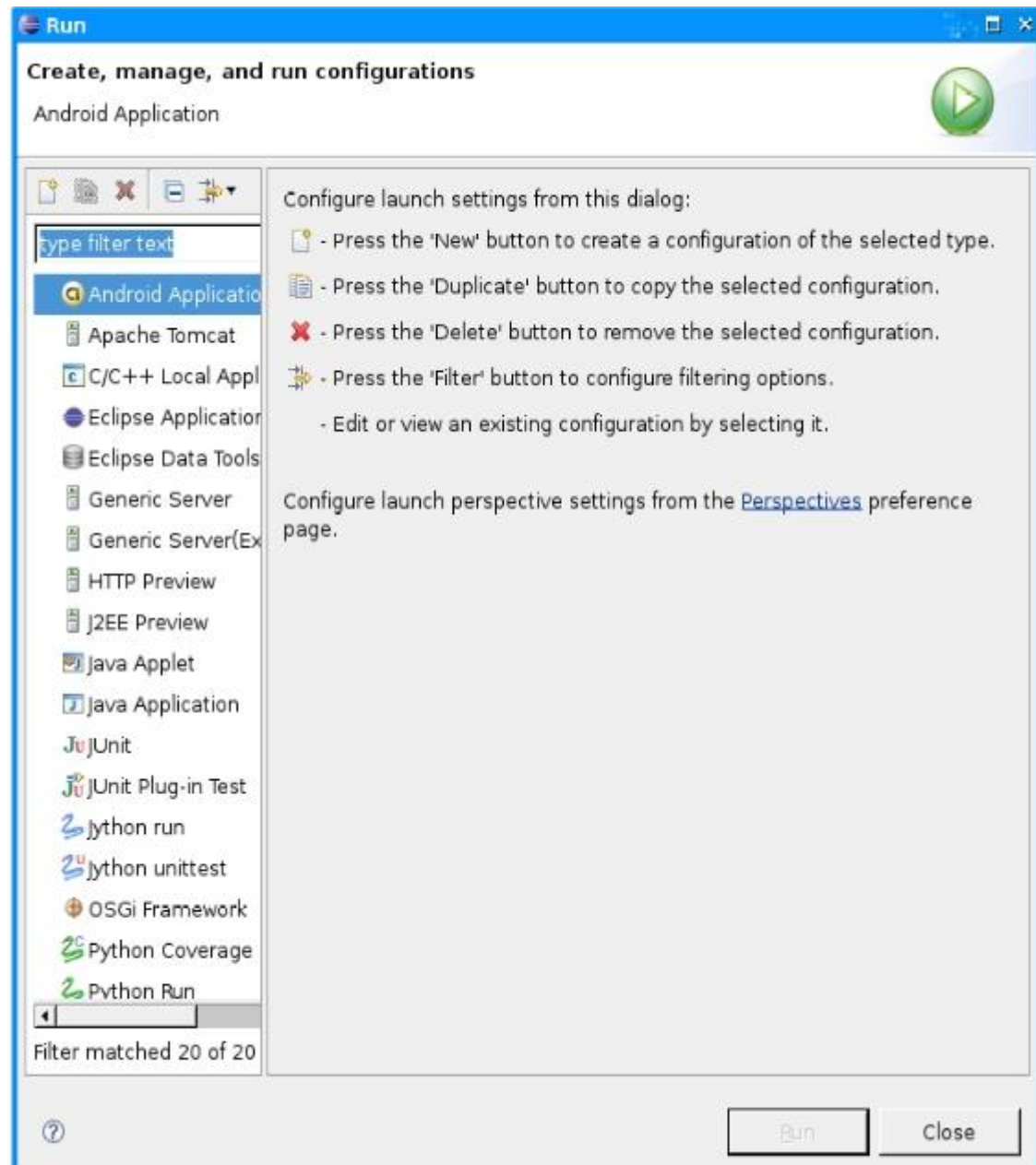
```
setContentView(tv);
```

Activity 的 setContentView()方法指示出系统要用哪个 View 作为 Activity 的界面, 如果一个 Activity 类没有执行这个方法，将会没有界面并且显示白屏。在这个程序中，我们要显示文本，所以我们传入已创建好的 TextView。

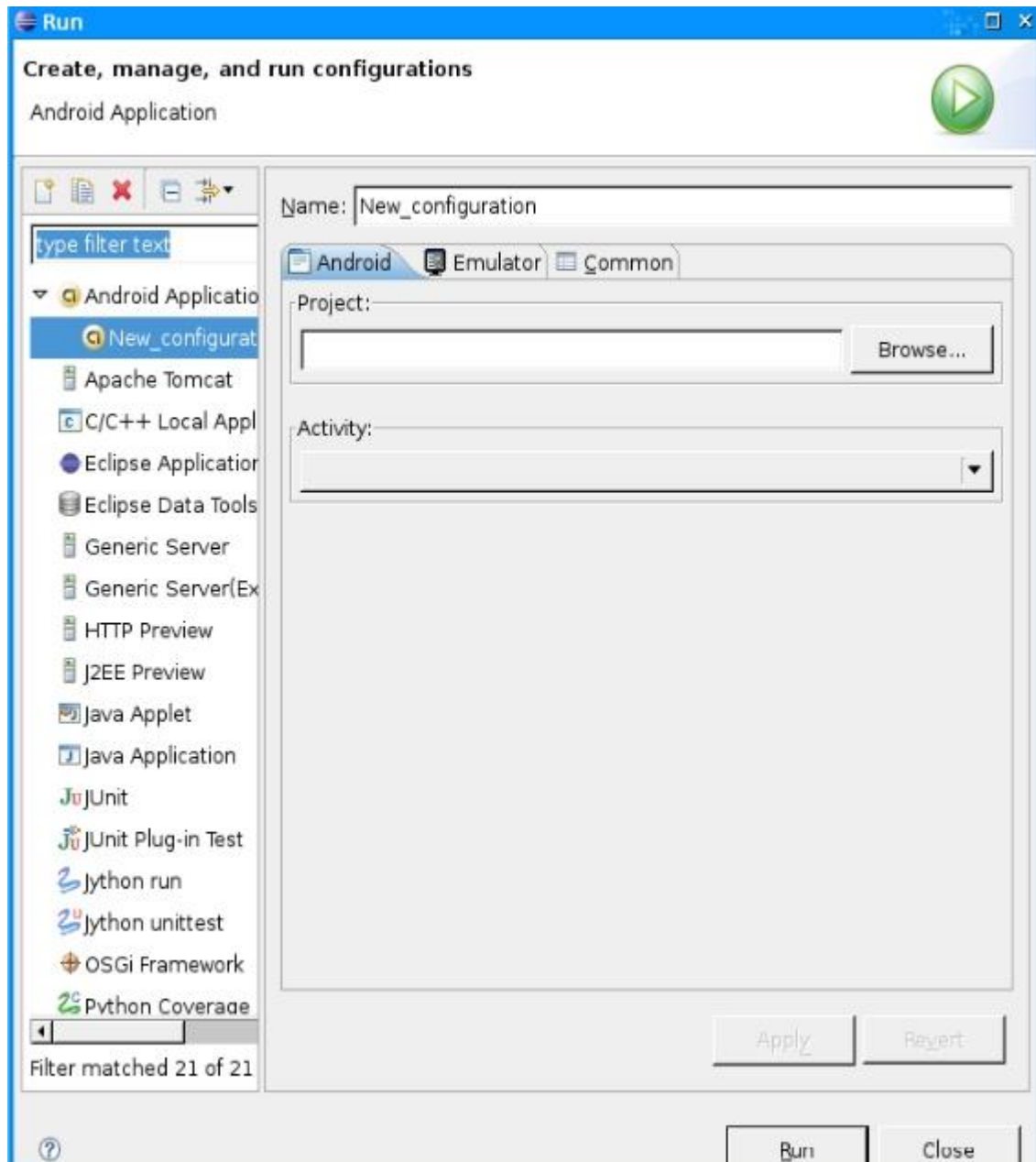
好了，程序代码已经写好，下面看看运行效果。

运行代码: Hello, Android

使用 Android 的 Eclipse 插件就可以很轻松的运行你的程序 , 选择 Run -> Open Run Dialog。你将会看到下面的对话框

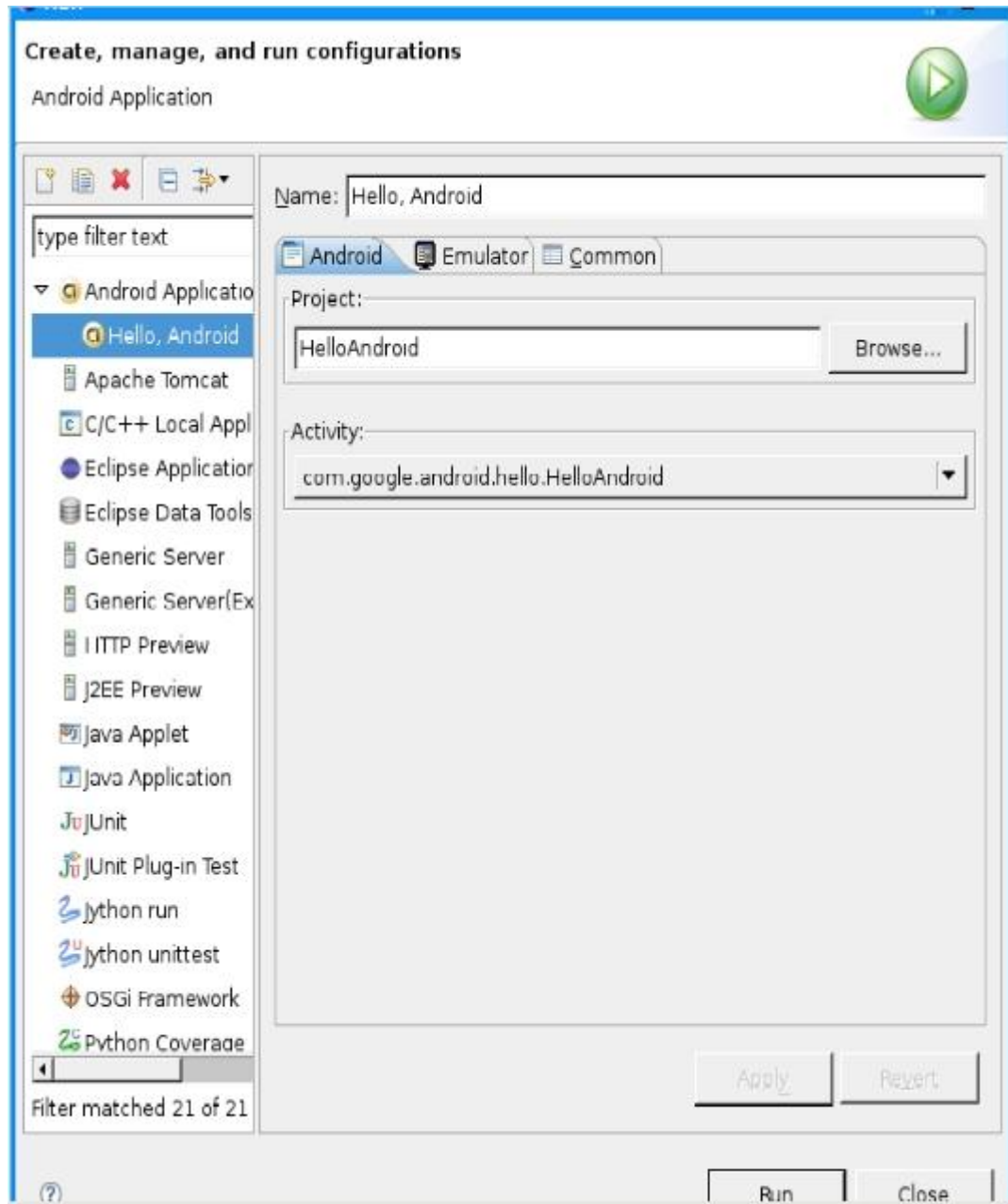


下一步, 高亮"Android Application" 标签, 然后按下左上角的图标(就是像一片纸带个小星星那个), 或者直接双击 "Android Application" 标签, 你将会看到一个新的运行项目, 名为"New_configuration".



取一个可以表意的名称 ,比如"Hello, Android", 然后通过 Browser 按钮选取你的项目 (如果你有很多个项目在 Eclipse 中 , 确保你选择要运行的项目), 然后插件会自动搜索在你的项目中的 Activity 类并且将所有找到的添加在"Activity"标签的下拉列表中。 我们只有 "Hello, Android"一个项目 , 所以它会作为默认选择。

点击"Apply" 按钮,下图



到这里，已经完成了，你只需要点击"Run"按钮，然后 Android 的模拟器将会启动，你的应用程序就会被显示出来。



下次继续介绍如何把界面的布局用 XML 表示以及使用命令行进行编译和运行 Android 程序。

将界面实现用 XML 编排

你刚刚完成的"Hello, World"例子我们称之为“程序化”的界面编排。意思就是说构建你的应用程序界面是直接使用的源代码。如果你已经完成过很多界面程序，你大概熟悉像此类的方式是多么脆弱：一个对布局小小的修改会对源代码造成很头疼的事情。如果忘掉与 View 的紧密结合，这个导致代码出错和浪费你的调试时间的界面问题也会很简单。

这就是为什么 Android 提供了一种可替换的界面构建方式：基于 XML 的布局文件。最简单的解释这个概念的方式就是展示一个例子。我们就用刚才创建的项目来进行演示，达到相同的界面效果。

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="Hello, Android"/>
```

AndroidXML 布局文件的大体的结构很简单。它是一个标签的树，任何一个标签就是 View 类的名字。在这个例子中，它是一个很简单的只有一个元素的树，一个 TextView。 你可以使用任何继承自 View 类的名字作为标签的名字。包

括在你的代码中自定义的 View 类。这个结构可以很容易的构建界面，它比你在源代码中使用的结构和语法更简单。这个模式的设计灵感来自于 Web 开发。就是可以将界面和应用程序逻辑分离的模式。

在这个例子中，也有些是 XML 的属性，下面是他们的含义：

| 属性 | 含义 |
|-----------------------|---|
| xmlns:android | 这是 XML 命名空间的声明,它是告诉 Android 的工具,你将要涉及到公共的属性已被定义在 XML 命名空间。 在每一个 Android 的布局文件的最外边的标签必须有这个属性。 |
| android:layout_width | 这个属性定义了屏幕上这个 View 可用的宽度是多少。 |
| android:layout_height | 这个属性定义了屏幕上这个 View 可用的高度是多少。 |
| android:text | 设置 TextView 所包含的文本内容 ,当前设置为“ Hello, Android” 信息 |

以上就是一个 XML 布局文件的样子，但是你需要放在哪里？它要放在你的项目目录的 res/ 文件夹下。“res”是“resources”的缩写，它是存放所有非代码资源的文件夹，包含象图片，本地化字符串和 XML 布局文件。

这些 Eclipse 的插件已经给你创建好了，在我们上面的例子中，我们没有使用它。在包浏览器中，展开 res 目录的 layout. 并且编辑 main.xml, 替换掉那个文本内容，然后保存。

现在，在包浏览状态，打开在代码文件夹中名为 R.java 的文件，你将看到下面的内容：

```
public final class R {  
    public static final class attr {  
    };  
    public static final class drawable {  
        public static final int icon=0x7f020000;  
    };  
    public static final class layout {  
        public static final int main=0x7f030000;  
    };  
    public static final class string {
```

```
        public static final int app_name=0x7f040000;  
    };  
};
```

一个项目的 R.java 文件是一个定义所有资源的索引文件。使用这个类就像使用一种速记方式来引用你项目中包含的资源。这个有点特别的强大像对于 Eclipse 这类 IDE 的代码编译特性，因为它使你快速的，互动式的定位你正在寻找的特定引用。

到目前需要注意的重要事情是叫做“ layout” 的内部类和他的成员变量“ main”，插件会通知你添加一个新的 XML 布局文件，然后从新产生这个 R.java 文件，比如你添加了新的资源到你的项目，你将会看到 R.java 也相应的改变了

最后重要的事情是你需要去修改你的 HelloAndroid 源代码，去使用新的 XML 布局你的界面。替换掉编码式的界面模式。下面是你的新代码的样子，你可以看到，代码变得更加简单了。

```
public class HelloAndroid extends Activity {  
  
    @Override  
  
    public void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.main);  
  
    }  
  
}
```

当你做这些改变的时候，不要仅仅复制，粘贴到你的代码中，尝试去体验 R.java 的代码编译特点。你会发现它对你有很大的帮助。

现在完成这些改变以后，你就可以重新运行你的程序，然后你会发现两种不同的界面编排方式会产生同样的效果。

调试你的项目

这个用于 Eclipse 中的 Android 插件作为 Eclipse 中的调试器也具有优秀的兼容性。要演示这些，让我们制造一个 bug 在代码中，改变你的 HelloAndroid 源代码象下面这样：

```
public class HelloAndroid extends Activity {

    @Override

    public void onCreate(Bundle icle) {

        super.onCreate(icle);

        Object o = null;

        o.toString();

        setContentView(R.layout.main);

    }

}
```

这次简单的变化会引起一个 NullPointerException 异常，如果你再次运行程序，你将会看到屏幕

要找到什么地方出错，需要设置个断点在你源代码的 “Object o = null;” 行后(你可以双击在 Eclipse 中显示行数的左部区域)，然后选择 Run -> Debu

g 选择最后一次的运行加载。你的程序将会重起模拟器，但是这个时候它会挂起，当它走到你刚才设置的断点的时候，在 Eclipse 的调试模式视图中，它就会停止在你的代码处。就像你可以在其他任意程序中做这个一样。。

使用其它的 IDE 工具创建项目

调试你的项目

这个用于 Eclipse 中的 Android 插件作为 Eclipse 中的调试器也具有优秀的兼容性。要演示这些，让我们制造一个 bug 在代码中，改变你的 HelloAndroid 源代码象下面这样：

```
public class HelloAndroid extends Activity {  
  
    /** Called when the activity is first created. */  
  
    @Override  
  
    public void onCreate(Bundle icle) {  
  
        super.onCreate(icle);  
  
        Object o = null;
```



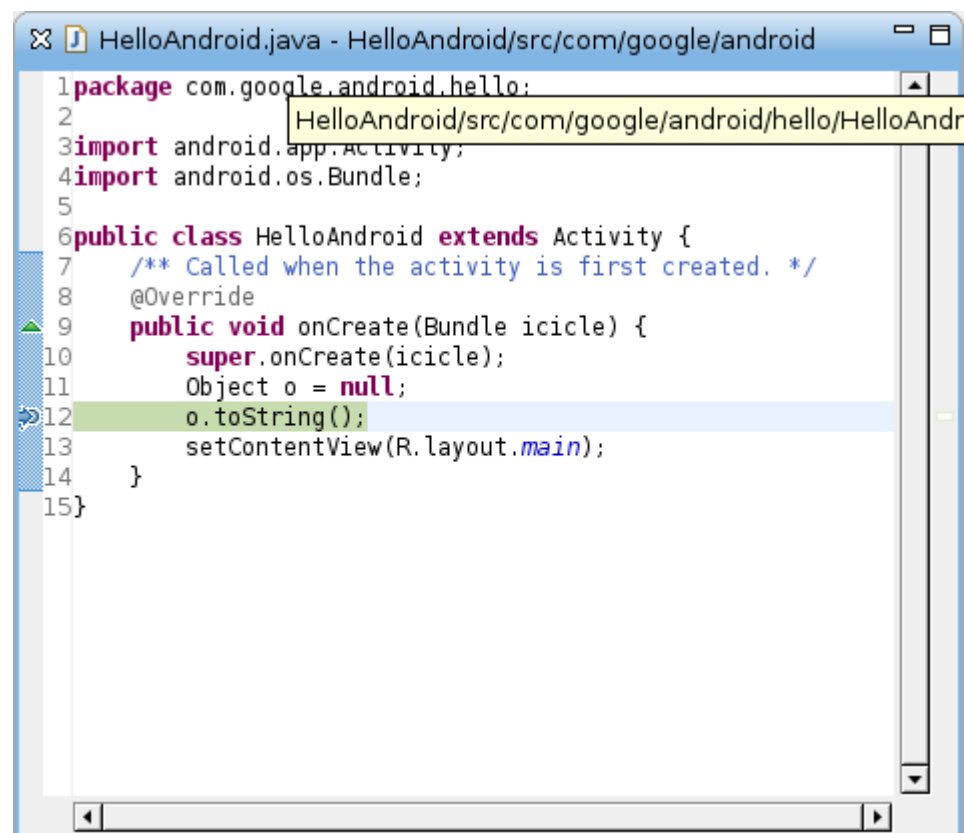
```
        o.toString();  
  
        setContentView(R.layout.main);  
    }  
}
```

这次简单的变化会引起一个 `NullPointerException` 异常，如果你再次运行程序，你将会看到下面的屏幕：



要找到什么地方出错，需要设置个断点在你源代码的 “Object o = null;” 行后(你可以双击在 Eclipse 中显示行数的左部区域)，然后选择 Run -> Debug 选择最后一行的运行加载。 你的程序将会重起模拟器，但是这个时候它会挂起，

当它走到你刚才设置的断点的时候，在 Eclipse 的调试模式视图中，它就会停止在你的代码处。就像你可以在其他任意程序中做这个一样。。



使用其它的 IDE 工具创建项目

如果你没有使用 Eclipse(比如你使用其他的 IDE 开发工具 ,或者简单的只使用文本编辑和命令行工具), 那么这个插件不会帮上你忙, 别担心, 你不会因为没有使用 Eclipse 而丢失任何开发能力。

其实 Android 的 Eclipse 插件也是将一套 Android SDK 的开发工具包装了。(这些工具像：模拟器，aapt, adb, ddms, 和其他的，可以参考 google 的相关文档)，因此，它也可以被其他工具包装，比如 ant.

在 Android SDK 中包含一个 Python 脚本，叫做“ activityCreator.py” ，它用于为了你的项目创建所有的源代码和目录的编译环境(个人理解)，也就是会产

生可用于 ant 编译的 build.xml. 这样就允许你的项目从命令行方式创建或者集成到你选择的 IDE 中。

比如,要创建一个 HelloAndroid 项目,就像刚才我们用 Eclipse 中的一样,你可以使用下面的命令:

```
activityCreator.py --out HelloAndroid  
com.google.android.hello.HelloAndroid
```

要编译你的项目,你要接着使用 ant 命令,当命令行提示成功时候,你会看到一个名为 HelloAndroid.apk 的文件在“ bin” 目录下。 这个.apk 文件是一个 Android 的包, 它需要使用“ adb” 工具安装进模拟器。

到这里为止,我向大家展示了 Android 平台的程序简单的开发实例,本人水平有限,翻译的时间仓促,一定有很多错误和疏漏,请多提修改意见。本系列教程的主要内容都是翻译自 Google 的 Android 开发文档。通过下面的链接查看全部内容。

<http://code.google.com/android/intro/hello-android.html>

经典代码资料汇总篇:

此处资料包含各类开发所需源代码,每个专题细分多个代码,欢迎各位下载使用。

[41 个经典 Android 软件开发工具打包下载](#)

[经典 Android 开发源代码大全](#)

[适合 Android 初学者的 9 个游戏源码](#)

[百度地图项目开发培训视频与源码](#)

[植物大战僵尸项目开发视频与源码](#)

[豆瓣阅读项目开发视频与源码](#)

[Google Android 开发入门与实战视频与源码](#)

[Android 开发功能实现实例源码](#)

[Android 开发经典实例源码](#)

[Android 益智类游戏开发源码](#)

[Android 应用开发程序源码](#)

[Android 开发工具及插件【官方最新版】](#)