



安卓开发环境与常用布局

第一章

Android简介与开发环境搭建

本章目标

- 3G的概念
- Android的改变
- Android开发环境的搭建
- 编写第一个Android程序
- Android程序的目录结构

手机发展史

1. 3G的概念

3G全称为：3rd Generation，中文含义即为第三代数字通信，是指将无线通信与国际互联网等多媒体通信结合的新一代移动通信系统

2. 符合3G标准的技术

CDMA2000

WCDMA

TD-SCDMA

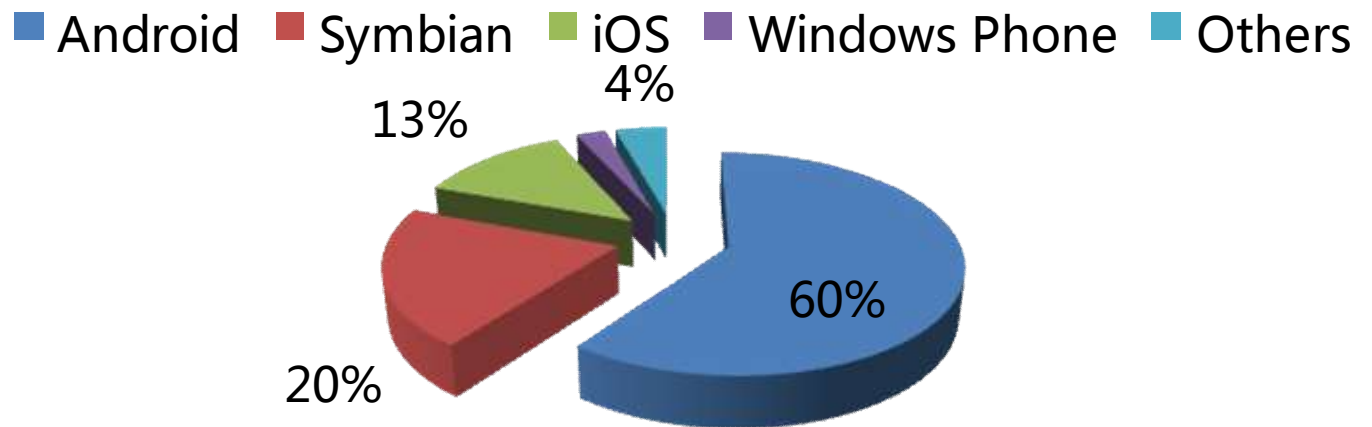


移动通信专家

Android的概念

Android是Google于07年11月5日宣布的基于Linux平台开源的手机操作系统名称，该平台由操作系统、中间件、用户界面和应用软件组成

3% **智能手机操作系统分布比例**



Android的升级

Android系统今后将继续每半年一次的升级步伐，分别定在每年的夏天和年终。新一代Android系统都将以甜点命名.比如:

1.5版叫做Cupcake (纸杯蛋糕)

1.6版为Donut (甜甜圈)

2.1版的Éclair (闪电泡芙，一种法式奶油夹心甜点)

2.2版的Froyo (冻酸奶)

2.3版的Gingerbread (姜饼). 添加了sip通话的支持，在省电上下了功夫，
黑色主题

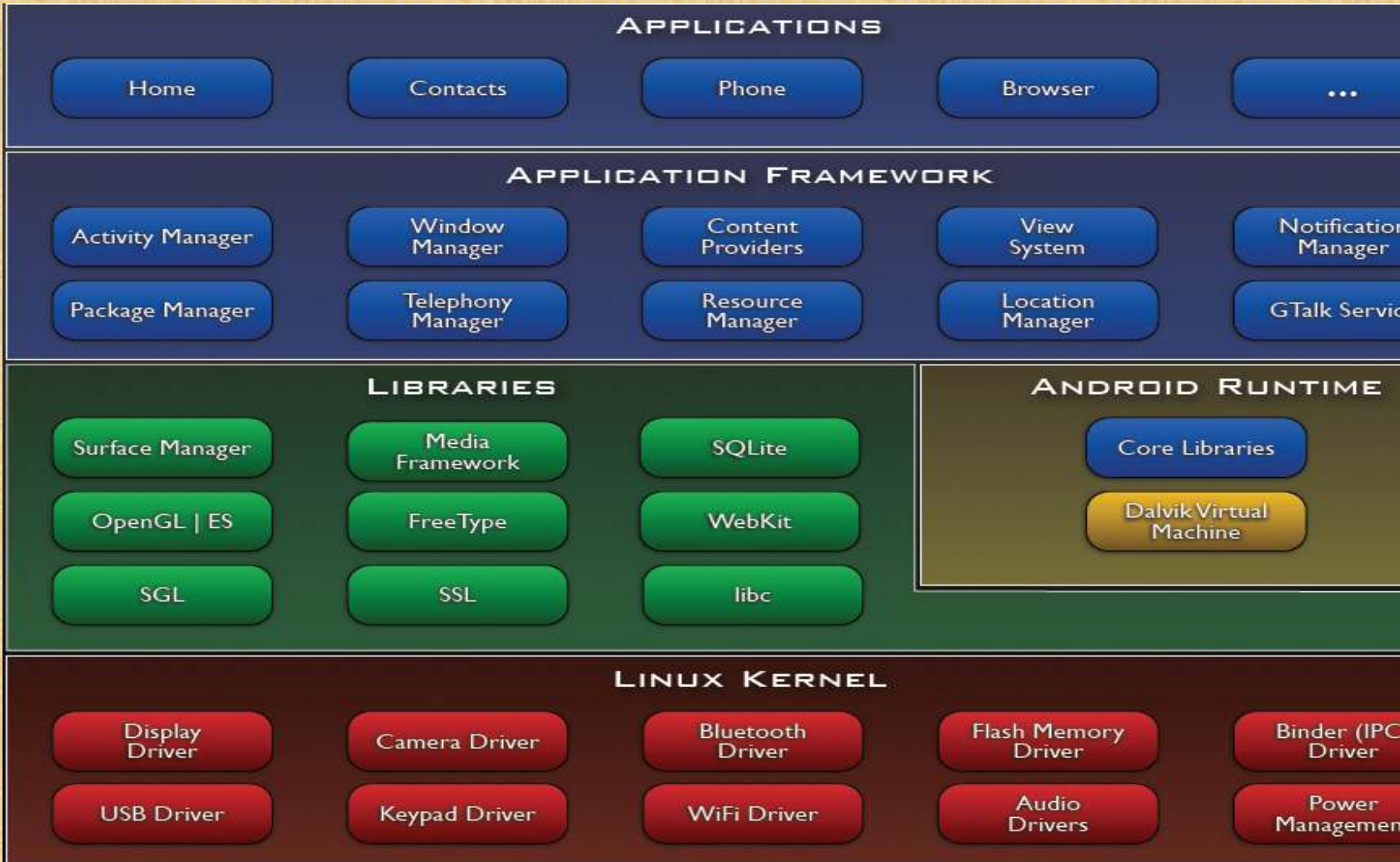
3.0版的Honeycomb (蜂巢) 主要是针对平板电脑，为平板和手机的整合做
，有了对设备的判断开关

4.0版的Ice cream SandWich(简称ics，冰激凌三明治) 统一版本，电视手机
融合

4.1版的Jelly Bean(果冻豆)整合新的google的服务

4.2版的Jelly Bean修正了4.1的bug

Android架构体系



Android特点

1.开源

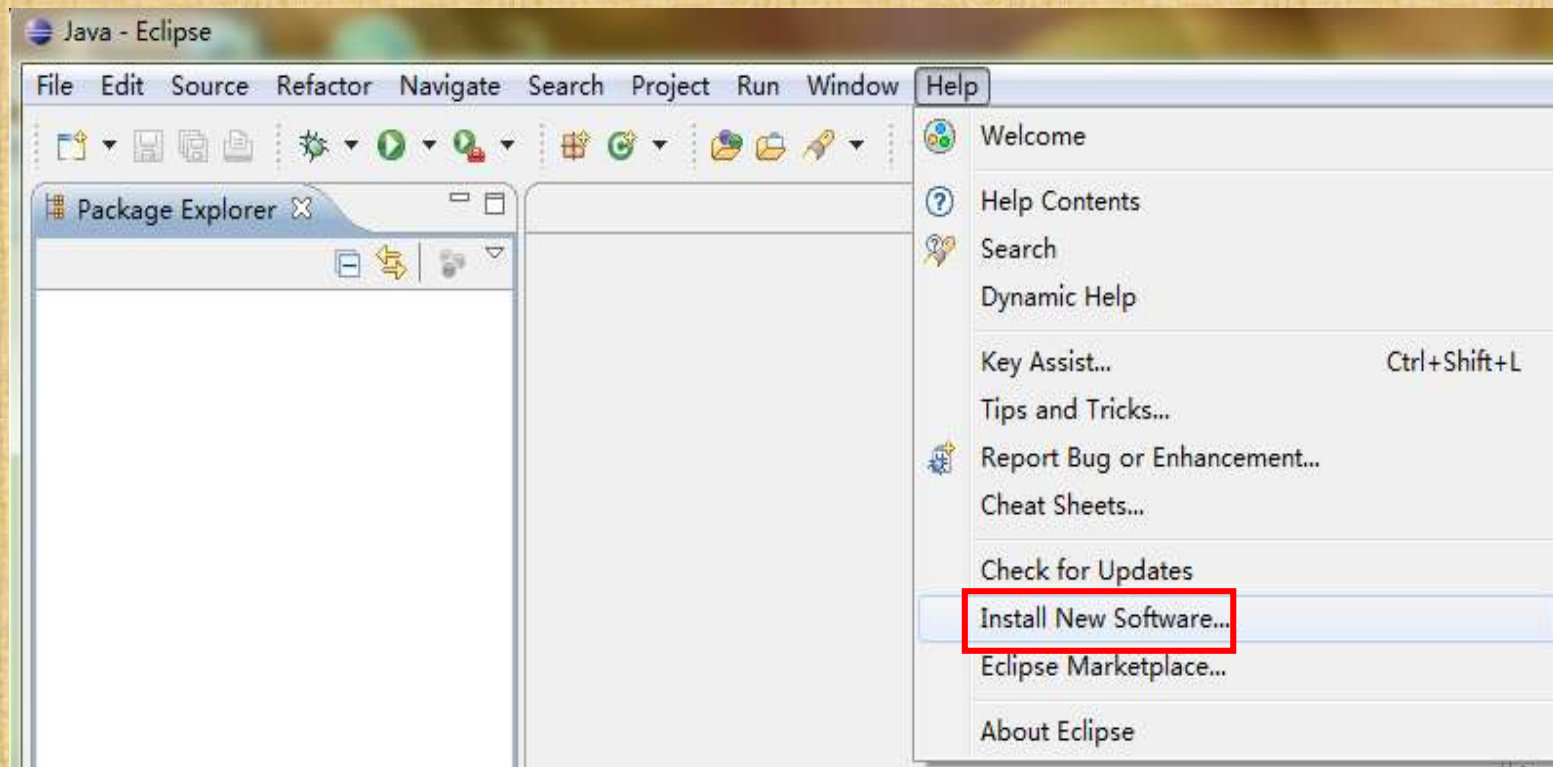
2.基于JAVA语言

3.Eclipse IDE

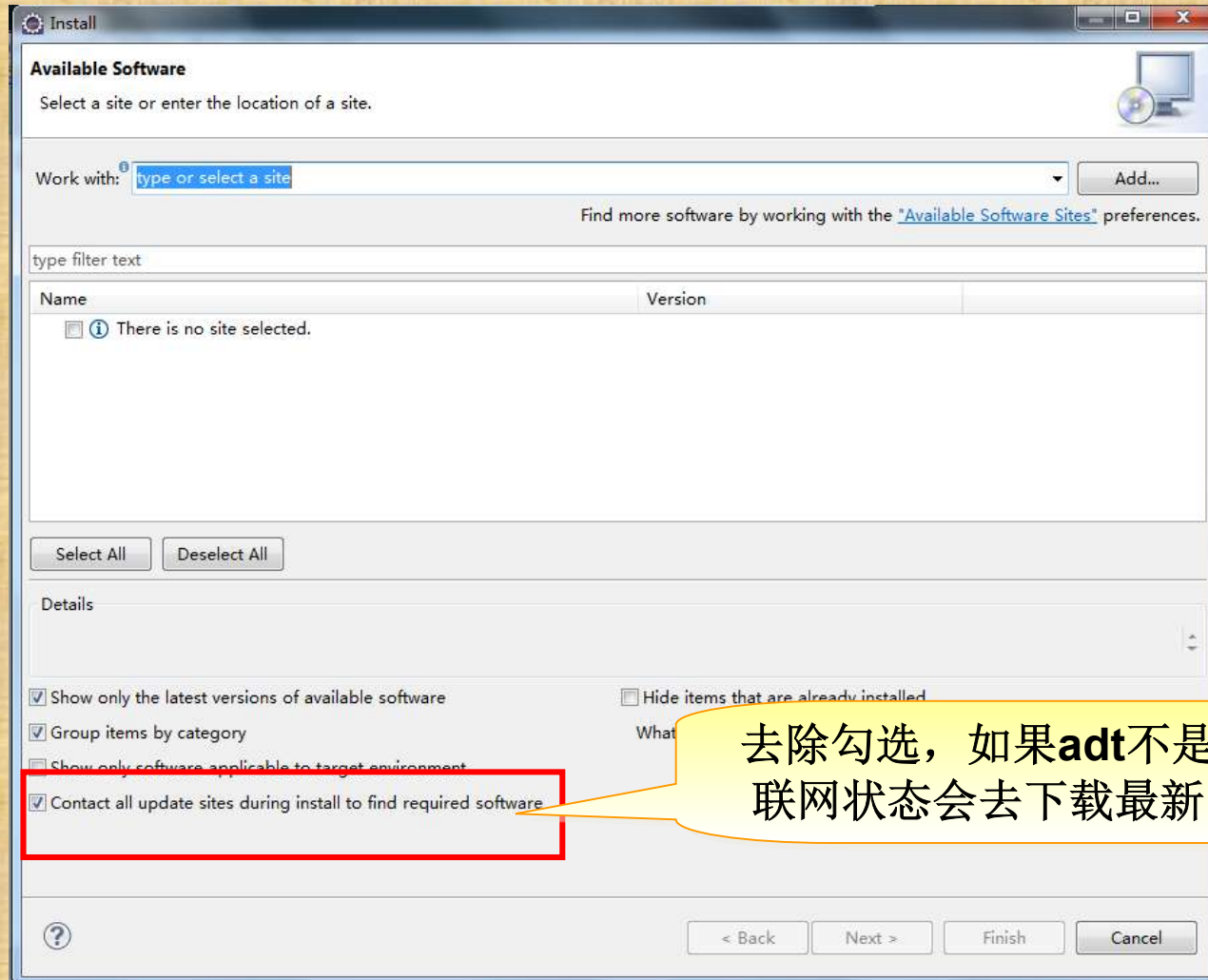
Android开发环境

- 1.JDK1.5以上（仅有jre不够）
- 2.Eclipse 3.5
- 3.ADT插件
- 4.SDK工具包

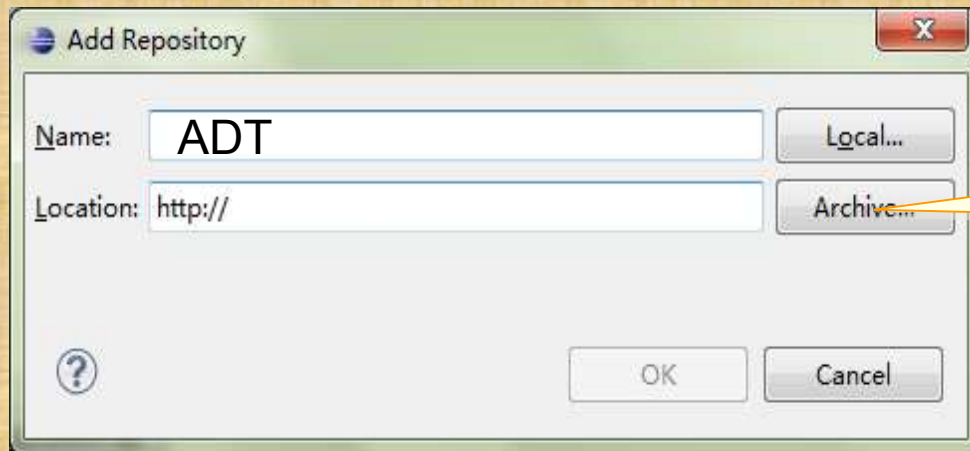
ADT安装步骤1



ADT安装步骤2

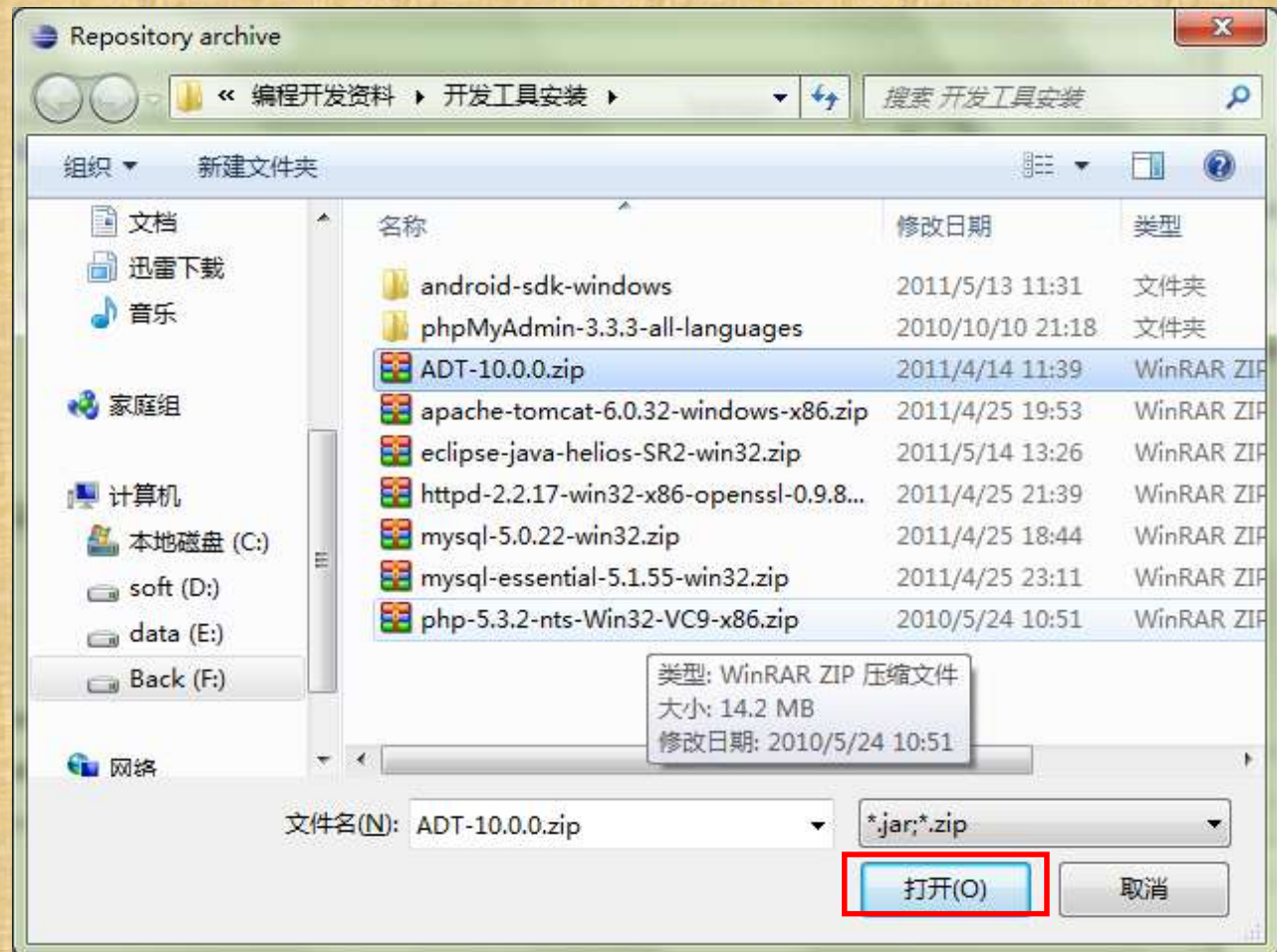


ADT安装步骤3

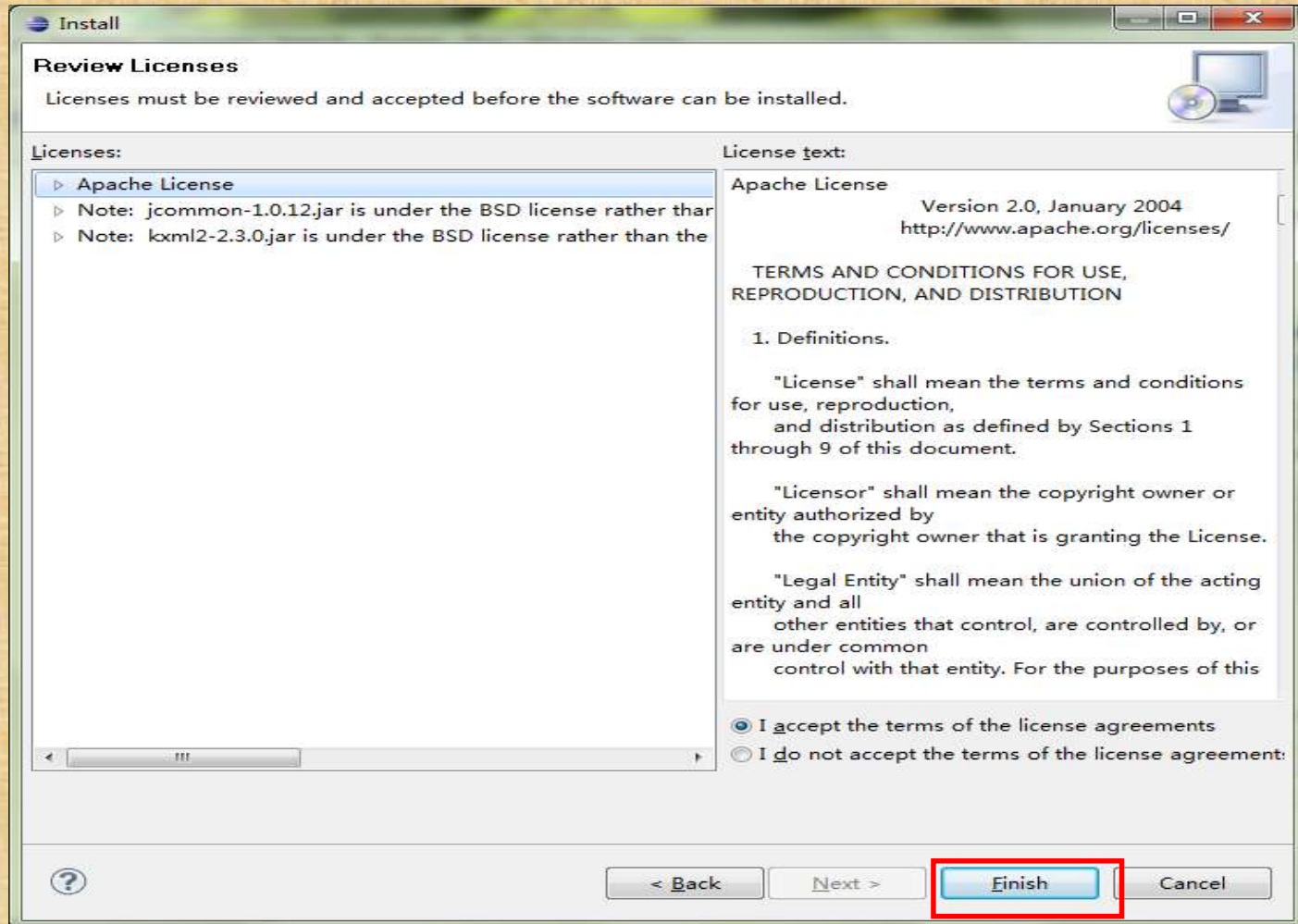


选择**ADT**文件所在的目录

ADT安装步骤4



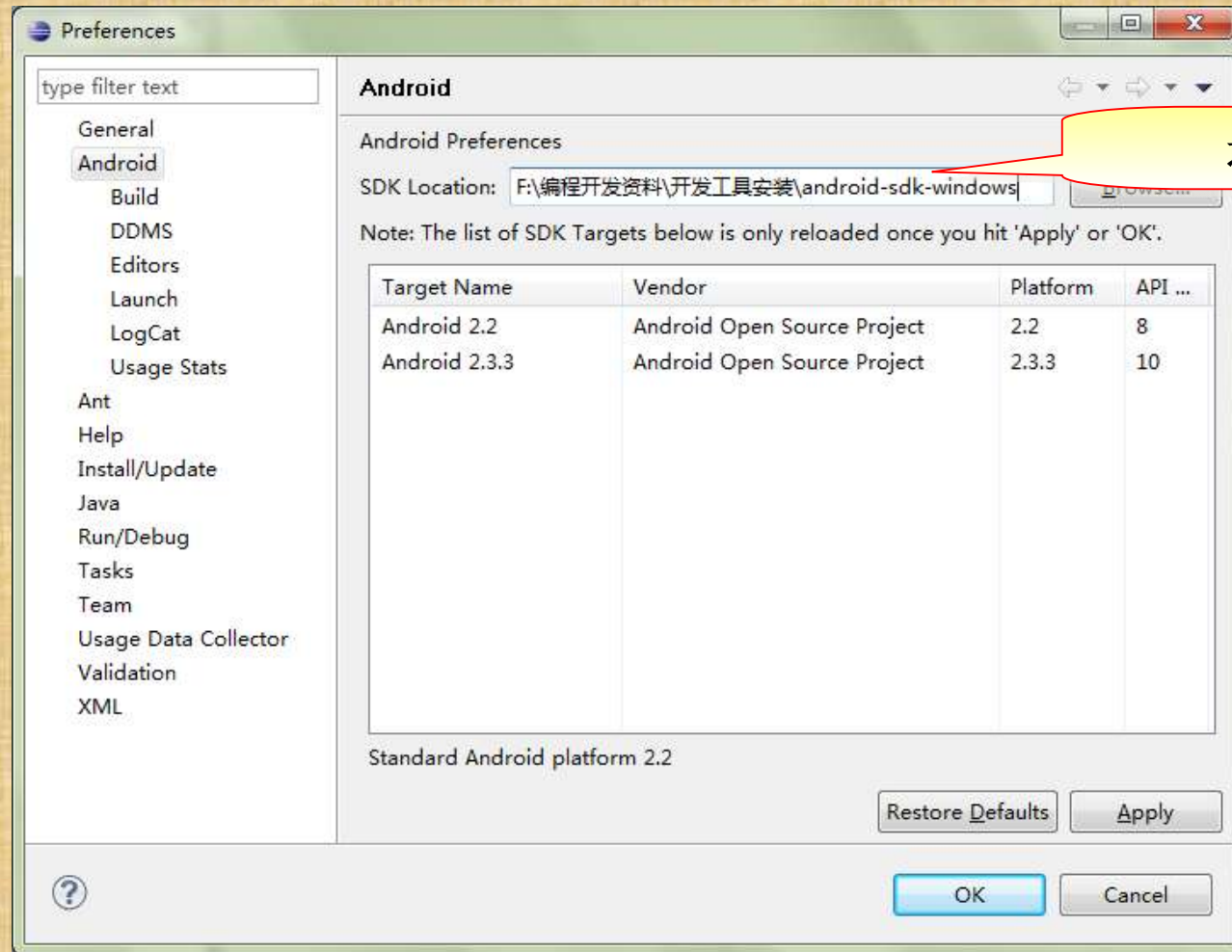
ADT安装步骤5



重启Eclipse
如果工具栏
出现上面的
图标，说明
你的ADT安
装成功

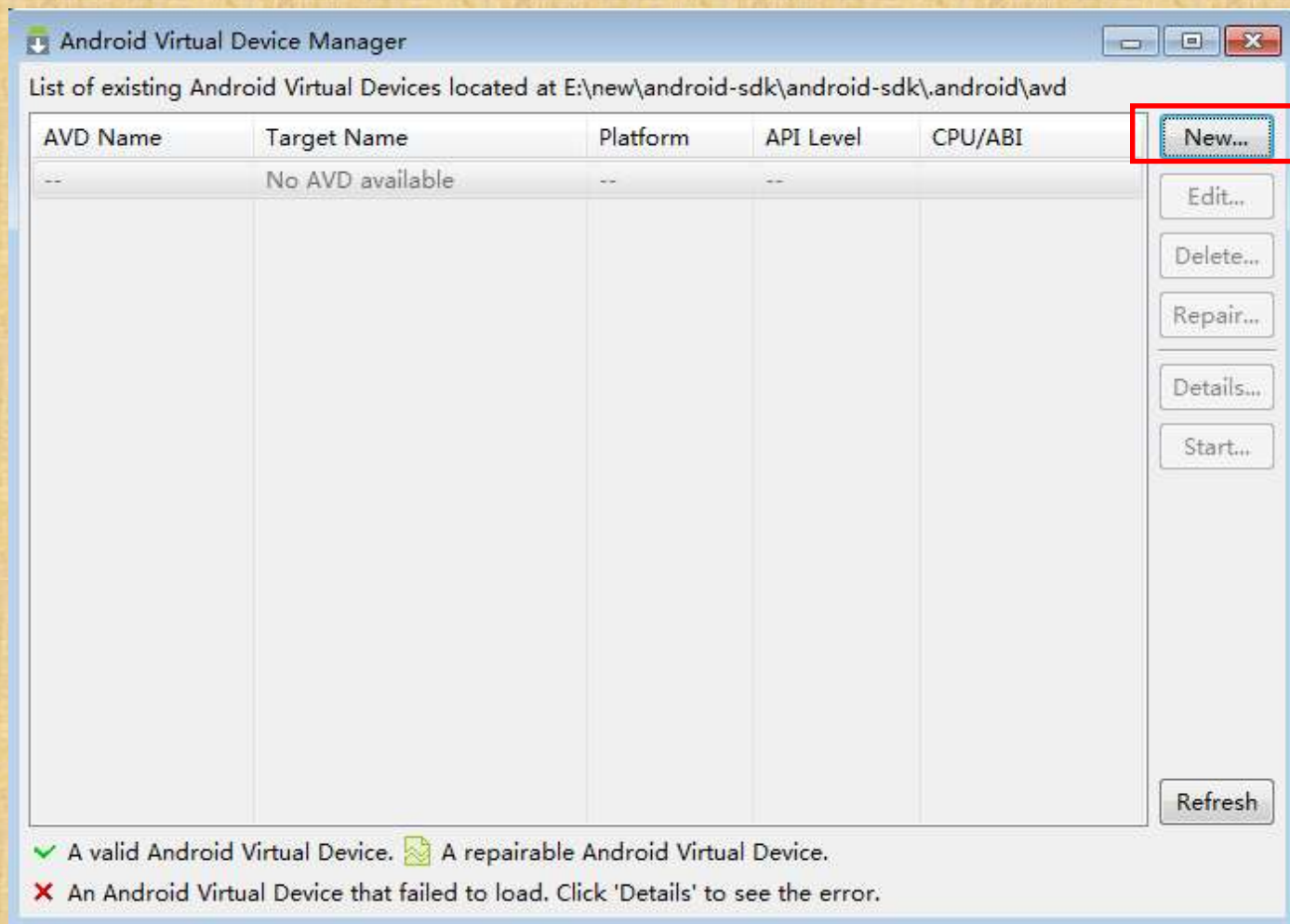


SDK安装步骤1



选择SDK所在的目录

创建模拟器



创建模拟器

Create new Android Virtual Device (AVD)

Name: jky

Target: Android 2.2 - API Level 8

CPU/ABI: ARM (armeabi)

SD Card:

Size: 100 File: Browse...

Snapshot:

Enabled

Skin:

Built-in: HVGA Resolution: x

Hardware:

Property	Value	
Abstracted LCD density	160	
Max VM application h...	24	

Override the existing AVD with the same name

Create AVD Cancel

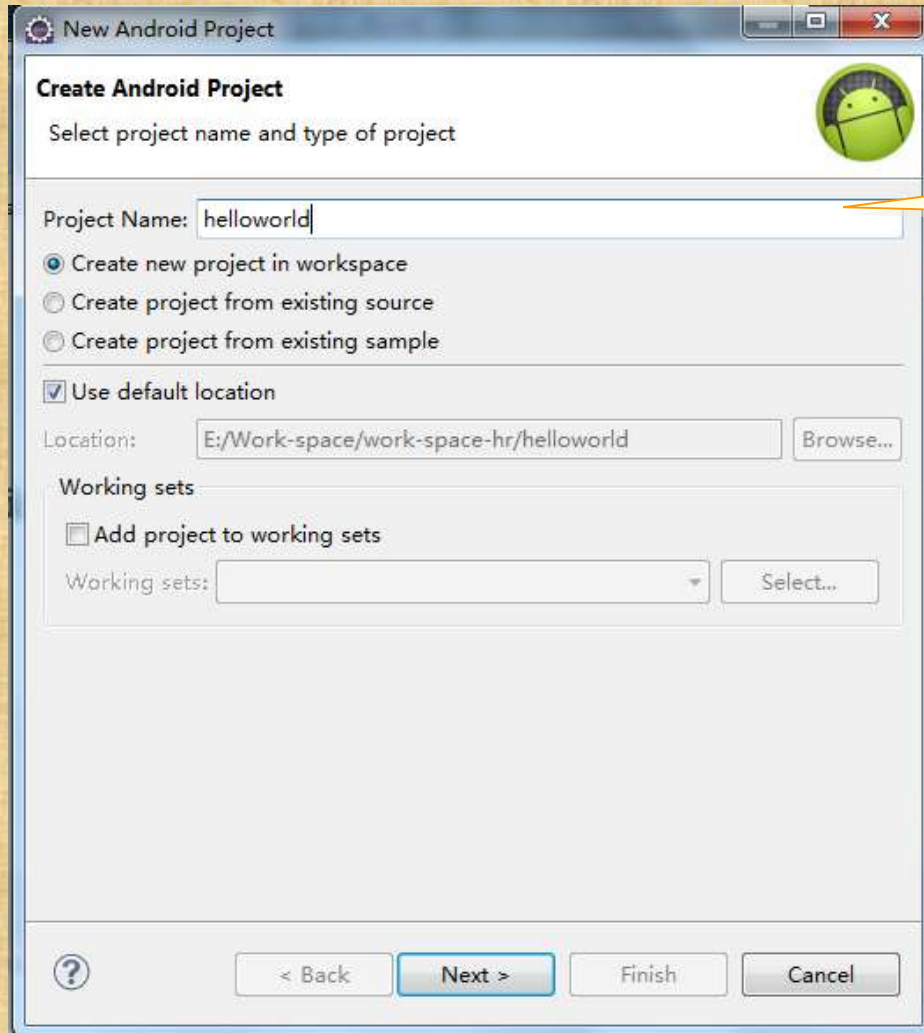
模拟器名

模拟器系统版本

Sdcard的大小，不要指定

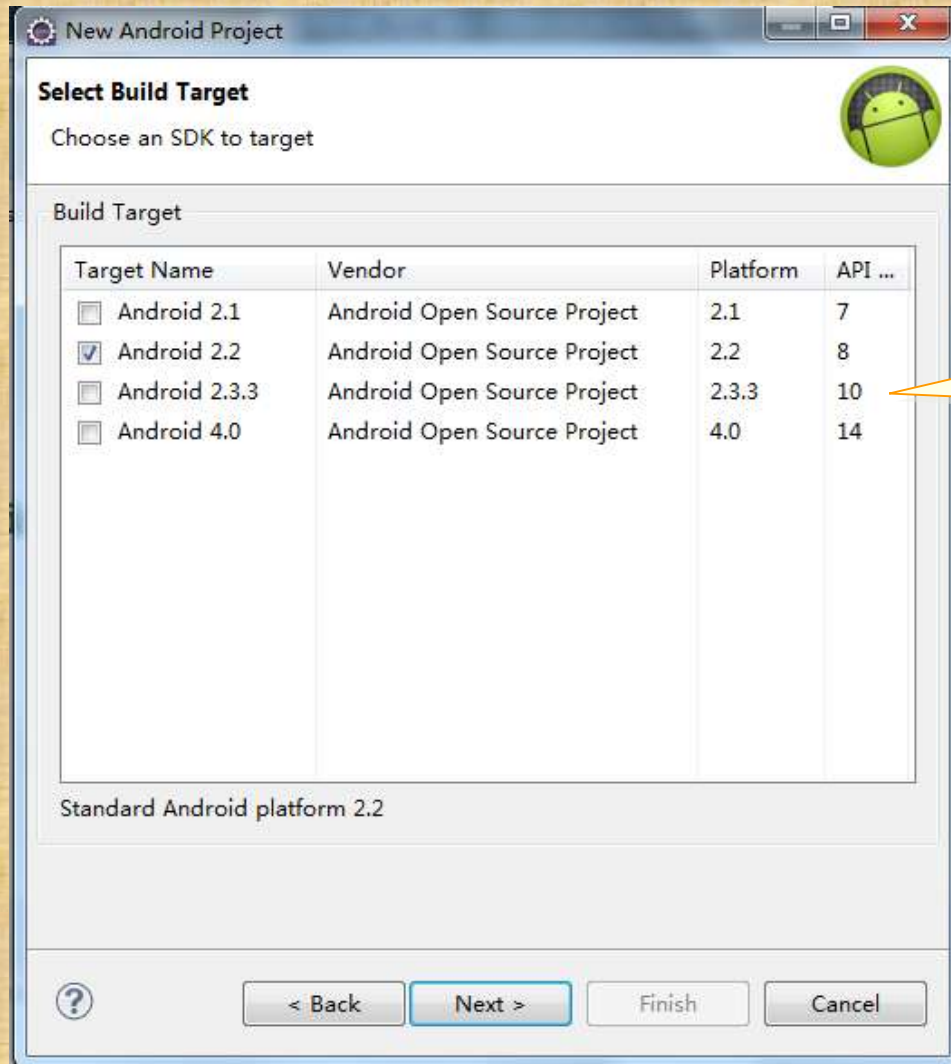
模拟器屏幕的大小

HelloWord的创建步骤1



工程名小写（有意义）

HelloWord的创建步骤2



选择工程所使用的android的版本，2.2是bug比较少的一个版本

HelloWord的创建步骤3

New Android Project

Configure the new Android Project

Application Name: Helloworld

Package Name: com.huaa.helloworld

Create Activity: MainActivity

Minimum SDK: 8

Create a Test Project

Test Project Name: helloworld-test

Test Application: HelloworldTest

Test Package: com.huaa.helloworld.test

< Back Next > **Finish** Cancel

工程名
包名：应用的唯一标示
主显示
最小的**sdk**

Android工程目录

src/ java源代码存放目录

gen/ 自动生成目录

gen 目录中存放所有由Android开发工具自动生成的文件。目录中最重要的就是R.java文件。这个文件由Android开发工具自动产生的。Android开发工具自动根据你放入res目录的xml界面文件、图标与常量，同步更新修改R.java文件。正因为R.java文件是由开发工具自动生成的，所以我们应避免手工修改R.java文件。R.java在应用中起到了字典的作用，它包含了界面、图标、常量等各种资源的ID。通过R.java，应用可以很方便地找到对应资源。另外编译器也会检查R.java文件中声明的资源是否被使用到，没有被使用到的资源不会编译进软件中，这样可以减少应用在手机占用的空间。

res/ 资源(Resource)目录

在这个目录中我们可以存放应用使用到的各种资源，如xml界面文件，图片或数据文件。

AndroidManifest.xml 功能清单文件

这个文件列出了应用程序所提供的功能，在这个文件中，你可以指定应用程序使用的服务(如电话服务、互联网服务、短信服务、GPS服务等等)。另外当你新添加一个Activity的时候，也需要在这个文件中进行相应配置，只有配置好后，才能启动此Activity。

Android工程目录

res/drawable 专门存放png、jpg等图标文件。在代码中使用 `getResources().getDrawable(resourceId)` 获取该目录下的资源。

res/layout 专门存放xml界面文件，xml界面文件和HTML文件一样，主要用于显示用户操作界面。

res/values 专门存放应用使用到的各种类型数据。不同类型的数据存放在不同的文件中，如下：

- **strings.xml** 定义字符串和数值。

res/xml 该目录下专门存放XML资源文件。

res/raw 该目录下的文件将直接被复制到设备上。编译软件时，这些数据不会被编译，它们被直接加入到程序安装包里。

Main布局

```
<LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
  android:layout_width="fill_parent"  宽: 填充父窗口  
  android:layout_height="fill_parent"  高:  
  android:orientation="vertical" >    方向: 垂直
```

线性

```
<TextView  
  android:layout_width="fill_parent"  
  android:layout_height="wrap_content"  包裹内容  
  android:text="@string/hello" />
```

文本

```
</LinearLayout>
```

AndroidManifest文件

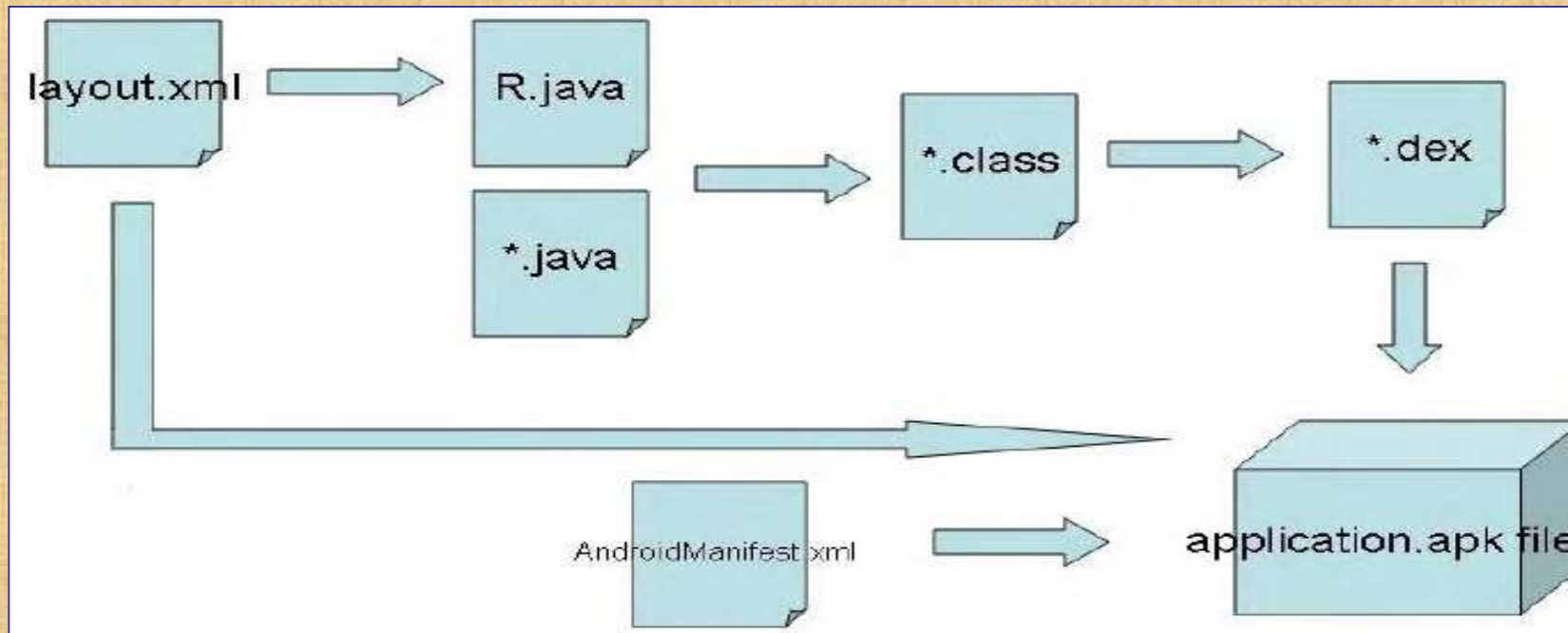
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.huao.lsn1" 包名
  android:versionCode="1" 版本号
  android:versionName="1.0" >
  <uses-sdk android:minSdkVersion="8" /> 安装系统最低版本号
  <application
    android:icon="@drawable/ic_launcher" 应用程序名称
    android:label="@string/app_name" > 应用程序图标
    <activity
      android:label="@string/app_name" 标题名
      android:name=".LsnActivity" >
      <intent-filter >
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```


HelloWorld程序

```
public class MainActivity extends Activity {  
    protected void onCreate(Bundle savedInstanceState)  
        super.onCreate(savedInstanceState);  
        //设置界面的布局  
        setContentView(R.layout.main);  
    }  
}
```

设置界面的
布局

Android应用程序打包过程



我们通过加压工具就可以查看

Dalvik VM 和JVM的对比

区别	jvm	dalvik vm
编译后文件格式	.java->.class->.jar	.java->.class->.dex->.odex

优点
在编译时优化代码，而不是在运行时，将多个文件整合成一个，整体减少文件个数，减少i/o操作，提高类查询的速度，常量池的引入

字节码格式	零字节地址格式	二/三地址的混合形式
-------	---------	------------

执行的效率要高些，零地址指令更加紧凑，但完成操作需要更多的load/store指令（指令分配次数和内存访问次数），二/三地址占用内存多些，但操作更少，访内存执行速度是一个瓶颈

基于的架构	基于栈的架构	基于寄存器的架构
-------	--------	----------

总结



1. 说说手机的发展史？
2. ANDROID是什么？
3. 开发ANDROID需要哪些工具？
4. 怎么去搭建ANDORID开发环境？
5. 怎样去创建一个ANDROID应用？

作业



1. 自己动手搭建android开发环境
2. 编写HelloWorld应用
3. 将应用的标题改为你好世界

第二章 常用布局

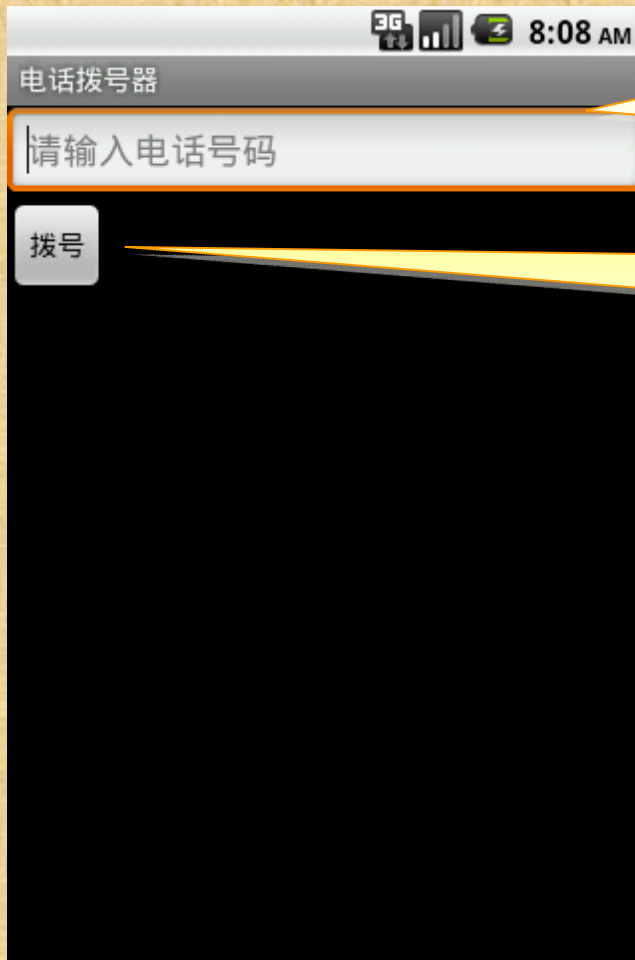
本章相关词汇

单 词	说 明
<code>EditText</code>	编辑框
<code>Button</code>	按钮
<code>Intent</code>	意图
<code>SmsManager</code>	短信管理器
<code>LinearLayout</code>	线性布局
<code>RelativeLayout</code>	相对布局
<code>TableLayout</code>	表格布局
<code>FrameLayout</code>	帧布局
<code>AbsoluteLayout</code>	绝对布局

本章目标

- 电话拨号器
- 常用布局

电话拨号器



EidtText (文本编辑框)

Button (按钮控件)

布局分析



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
```

线性布局-垂直

```
<EditText android:id="@+id/et_number"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="@string/enter_alert"
    android:inputType="phone" />
```

编辑框 控件

```
<Button android:id="@+id/bt_dial"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/dial"/>
</LinearLayout>
```

按钮 控件

核心代码

分析

```
public class MainActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
  
        //初始化控件  
        et_number = (EditText) findViewById(R.id.et_number);  
        bt_dial = (Button) findViewById(R.id.bt_dial);  
  
        //给按钮添加点击事件  
        bt_dial.setOnClickListener(new MyOnClickListener());  
    }  
}
```

继承Activity类

重写onCreate方法

- 设置显示的布局
- 布局是一个资源，引用：
R.layout.布局名

- 得到布局里面的控件：R.id.控件的id
- 通过findViewById方法初始化控件

核心代码



```
private class MyOnClickListener implements
OnClickListener{
    //点击回调该方法
    public void onClick(View v) {
        //得到输入的电话号码
        String number = et_number.getText().toString();

        //拨打电话 （调用系统里面已经存在的拨号组件）

        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_CALL);
        intent.setData(Uri.parse("tel:"+number));
        startActivity(intent);
    }
}
```

意图

设置动作名称

设置数据

激活组件

短信发送器



核心代码



```
SmsManager smsManager = SmsManager.getDefault();
```

得到短信发送器

```
smsManager.sendTextMessage(number,//短信的发送目标号码  
null,//短信中心号码  
sms_content,//短信内容  
null,//短信是否发送成功的意图  
null);//短信是否被对方收到的意图
```

发送短信

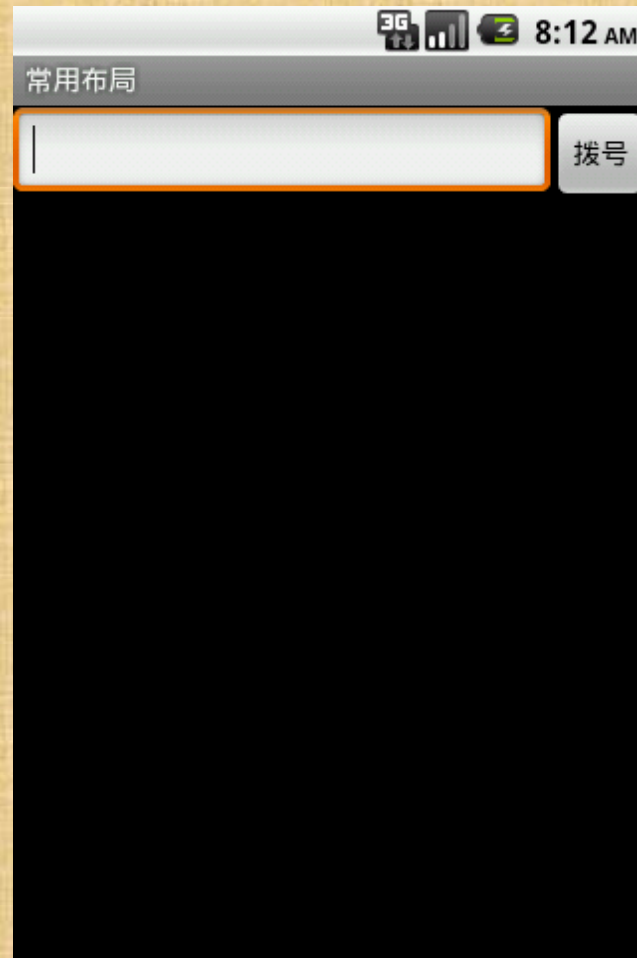
```
Toast.makeText(getApplicationContext(),R.string.success  
Toast.LENGTH_LONG);
```

土司提示

短信发送权限

```
<uses-permission android:name="android.permission.SEND_SMS"/>
```


线性布局 (LinearLayout)



相对布局 (RelativeLayout)



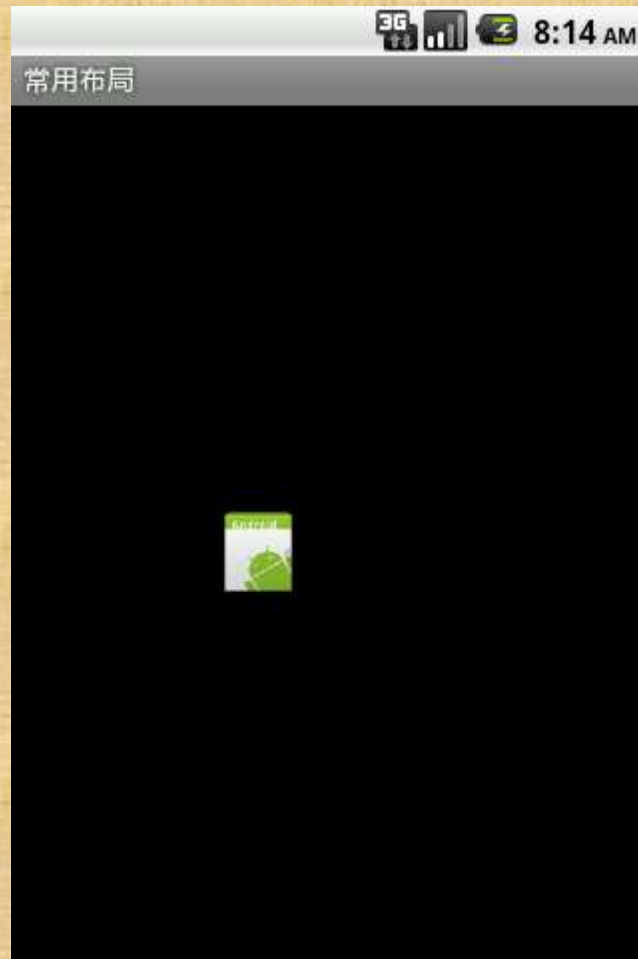
表格布局 (TableLayout)



帧布局 (FrameLayout)



绝对布局 (AbsoluteLayout)



总结



1. 如果要打电话，激活的组件是在那一层？
2. 今天我们学了那些布局？

作业



1. 实现打电话界面的创建
2. 用今天学的布局来实现右图效果：

