



【eoeAndroid特刊】

## 第三期：Android Market及应用发布

发布版本： Ver 1.0.0(build 2009.06.06)

© Copyright 2009 eoeAndroid.com. All Rights Reserved.

### 本篇简介

本期是eoeAndroid特刊的第三期，Android Market并不仅仅是Google的一个应用程序商店。虽然应用程序也很重要，但是在谷歌的Android Market中，它们却仅仅是微不足道的变量。这一平台最大的特点就在于，所有的开发者注册后都可以上传内容，而所有上传的内容都会能够立刻在Android Market上找到，这改变了以往只能通过付费获得内容的方式，为了让开发者更好、更直接全面的了解Android Market及其相关方面的内容，本期特刊主题是Android Market，其内容包含如下文章：

- 1、Google Checkout简介&开通流程  
--介绍什么是Google Checkout以及开通Google Checkout的流程。
- 2、Android Market简介&注册过程  
--介绍Android Market以及其注册流程
- 3、准备发布：一份核查清单  
--介绍发布应用前需要做的准备工作
- 4、应用程序签名  
--发布前，需要对你的应用进行签名
- 5、发布你的应用  
--本篇介绍如何发布你的应用
- 6、对应用程序进行版本管理  
--本篇介绍如何管理版本以及应用的升级
- 7、使用Android Market管理面板  
--本篇介绍如何查看管理面板中的统计数据和发布应用的信息
- 8、Android Market开发者发布协议  
--本篇让你更清楚的了解其发布协议。

希望本期内容能让更多的开发者熟悉Market及其相关内容，指导其注册、使用；实践社区协调作业的流程和氛围；自我提高，加深对Market相关内容的了解；

**致谢：**

本期专题得到如下同学的大力支持和积极响应，谢谢你们辛苦劳动，谢谢你们为Android发展和普及做出的贡献。

- IceskYsl: iceskysl@gmail.com,
- haiyangjy: wecan.jin@gmail.com,
- 404: xuliicom@gmail.com,
- zhoubo: zhoubo526@gmail.com,
- **apcwowo@hotmail.com** ,
- edison0951

**活动发起地址：**

【eoeAndroid特刊】第三期：Android Market及应用发布

<http://www.eoeandroid.com/viewthread.php?tid=358&extra=page%3D1>

目录:

第三期: Android Market及应用发布.....	1
本篇简介.....	1
1. Google Checkout简介&开通流程.....	5
1.1 简介.....	5
1.2 特色.....	5
1.3 注册.....	6
2. Android Market简介&注册过程.....	8
2.1 简介.....	8
2.2 特色.....	8
2.3 申请帐号.....	9
3. 准备发布: 一份核查清单.....	14
3.1 在您考虑发行应用程序之前.....	14
3.1.1 在真机上彻底地测试您的应用程序.....	14
3.1.2 考虑在您的应用程序里边添加一份终端用户许可协议.....	15
3.1.3 在应用程序的manifest文件中指定一个图标和名称.....	15
3.1.4 关闭日志记录和调试功能并清除不必要的文件或数据.....	15
3.2 在您最终编译应用程序之前.....	15
3.2.1 为您的应用程序设定版本.....	15
3.2.2 生成一份合适的密钥.....	16
3.2.3 如果您的应用程序用到MapView, 需注册一个Maps API Key.....	16
3.3 编译您的应用程序.....	17
3.4 当您的应用程序编译完成之后.....	17
3.4.1 签名您的应用程序.....	17
3.4.2 测试经编译过的应用程序.....	17
4. 应用程序签名.....	18
4.1 概述.....	18
4.2 签名策略.....	18
4.3 签名的基本设置.....	19
4.4 调试模式下的签名.....	19
4.4.1 Eclipse 用户.....	20
4.4.2 Ant 用户.....	20
4.4.3 调试证书的期限.....	20
4.4.4 签名正式发布软件.....	20
4.5 在发布模式下编译.....	20
4.5.1 Eclipse 用户.....	21
4.5.2 Ant 用户.....	21
4.5.3 获得适合的私有密钥.....	21
4.6 签名你的应用程序.....	22
4.7 保护你的私有密钥.....	23
5. 发布你的应用.....	24
5.1 发布到Android Market上.....	24
5.2 在Android Market上发布更新.....	24
5.3 利用Intents来启动设备上的Market应用程序.....	25
6. 对应用程序进行版本管理.....	27
6.1 文档包含.....	27
6.2 也包含.....	27
6.3 设定应用程序的版本信息.....	27
6.4 指定最低的系统API版本.....	28
7. 使用Android Market管理面板.....	29
7.1 查看应用列表和统计信息.....	29
7.2 上传应用的基本步骤.....	30
7.3 升级应用的过程.....	32
7.4 其他问题.....	32

- 8. Android Market开发者发布协议..... 33
  - 8.1 导言..... 33
  - 8.2 接受本协定..... 33
  - 8.3 价格和付款。..... 34
  - 8.4 对于你 Market的用处..... 34
  - 8.5 授权资助..... 35
  - 8.6 品牌特征和宣传..... 36
  - 8.7 产品 Takedowns..... 36
  - 8.8 您的开发证书..... 36
  - 8.9 隐私和信息..... 37
  - 8.10 终止本协定..... 37
  - 8.11 免责声明..... 37
  - 8.12 责任范围..... 37
  - 8.13 赔偿..... 37
  - 8.14 对协议的更改..... 38
  - 8.15 常规合法条款..... 38
- 9. 其他..... 39
  - BUG提交..... 39
  - 参加翻译..... 39
  - 关于eoeAndroid..... 39

正文开始》》

## 1. Google Checkout简介&开通流程

by: IceskYsl (<http://iceskysl.lsters.com/>)



Google好好找，Google Checkout轻松购！

想要更即时、更安全，而且更便利的线上购物方式吗？就在这里！

要开通Android Market，是离不开Google Checkout的，原因有：

- 注册Android Market开发者帐号，需要缴纳25美元的注册费，而其通过Google Checkout直接使用信用卡进行支付的；
- 要想提交付费的应用，需要申请 Google Checkout Merchant(零售商)服务，其使用Google Checkout收款；

如果你已经了解并使用过Google Checkout，那本篇文章你可以大致浏览即可，如果你尚未开通自己的Google Checkout帐号，你可以按照本文内容申请开通自己的Google Checkout帐号。

### 1.1 简介

Google Checkout服务类似易趣的贝宝(PayPal)，网上零售商可以将Checkout系统整合到自己的支付平台，在线购买者可以通过它实现安全付款。完成每次支付后，Google Checkout向商家收取0.20美元的手续费以及交易商品价格2%的费用。目前，美国超过90%的零售商注册了google checkout服务，包括DVD Empire, Jockey, Starbuck's, Levi's, Timberland, Buy.com等品牌零售网站。在中国，Google checkout的用户群体主要集中于外贸行业的个人及企业，当然也包括喜欢在国外买东西的人士。

### 1.2 特色

您不用再建立多个帐户和密码。

使用Google Checkout™，您可以迅速轻松地透过网路商店购买，还可再同一处追踪所有订单和运送情形。

您可以放心购物。

我们的防诈骗政策保护您透过Google Checkout安全购物，而且我们也不会向卖家透露您的购物记录或完整的信用卡号码。

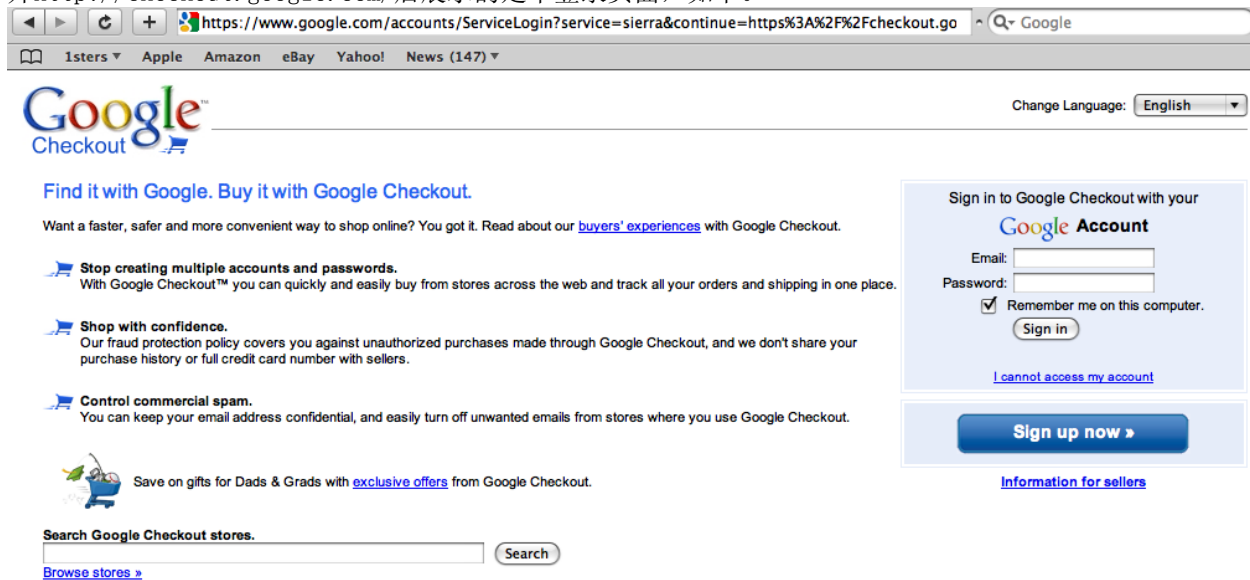
控制广告垃圾邮件。

您可以不公开您的电子邮件地址；另外，如果您不希望使用Google Checkout购物时的商家寄电子邮件给您，也很容易就能够关闭此功能。

Google的在线安全支付系统Google Checkout吸引了人们的高度关注并看好发展。网上零售商对Google Checkout在线支付系统的看法是，Google Checkout能否成功将最终取决于它为商家带来的顾客转化率高高低。

### 1.3 注册

要开通Google Checkout，你首先需要一个Google帐号，如果你尚未登录，则打开<http://checkout.google.com/>后展示的是个登录页面，如下。



Find it with Google. Buy it with Google Checkout.

Want a faster, safer and more convenient way to shop online? You got it. Read about our [buyers' experiences](#) with Google Checkout.

**Stop creating multiple accounts and passwords.**  
With Google Checkout™ you can quickly and easily buy from stores across the web and track all your orders and shipping in one place.

**Shop with confidence.**  
Our fraud protection policy covers you against unauthorized purchases made through Google Checkout, and we don't share your purchase history or full credit card number with sellers.

**Control commercial spam.**  
You can keep your email address confidential, and easily turn off unwanted emails from stores where you use Google Checkout.

Save on gifts for Dads & Grads with [exclusive offers](#) from Google Checkout.

Search Google Checkout stores.

[Browse stores »](#)

Sign in to Google Checkout with your **Google Account**

Email:

Password:

☒ Remember me on this computer.

[I cannot access my account](#)

[Information for sellers](#)

使用自己的Gmail帐号登录后，将看到开通Google Checkout页面，如下：

https://checkout.google.com/buyerSignup?continue=https%3A%2F%2Fcheckout.google.com%2Fmain%3Fhl%3Dzh

1sters Apple Amazon eBay Yahoo! News (147)

# Google Checkout

## Welcome to Google Checkout !

Sign up for a free Google Account in order to use Google Checkout : A faster, safer and more convenient way to shop online.

Email: **eoemobile@gmail.com** [Sign in as a different user](#)

Location: **Hong Kong**

Card number: **信用卡号码**  
Re-enter your card number

Expiration date: **8** / **2010** CVC:  [What's this?](#)

Cardholder name: **持卡人姓名(英文)**

Address Format: [Switch to domestic format](#)

Billing Address: **持卡人地址信息**

Area: **Select area**

Phone number: **电话号码**  
Required for account verification.

My shipping address is: ☒ My billing address ☐ A different address

☐ Send me Google Checkout special offers, market research, and newsletters.

在上面的界面中输入您个人的信息，需要注意的其location中并不包含中国大陆【原因很复杂，很无奈】，我们只能通过一些手段处理下，总结下需要注意的事项有：

- 国家里可能没有china，选择HongKong；
- 信用卡用visa和master卡比较容易通过；
- 申请成功后，可能会从你的卡里冻结1—2美金，作为验证卡的过程；
- 地址，姓名，都必须填写英文；

更多帮助信息，请参考：

<http://checkout.google.com/support/sell/?hl=zh>

## 2. Android Market简介&注册过程

导读：

**Android Market**是个开放的平台，开发者可以上传和销售自己的应用，用户可以随时随地的下载、安装和评价应用。本文将对**Android Market**做相关介绍，并演示其注册过程，通过本文学习，你将了解**Market**并掌握如何开通自己的**market**帐号。

### 2.1 简介

配合首款Android手机T-Mobile G1的上市，Google于2008年8月28号发布了**Android Market**，并将其定位为开放的内容分享系统，也就是应用程序超市。使用这个系统可以帮助使用**Android**操作系统的手机用户在该平台寻找、购买、下载及评级使用**Android**操作系统的手机应用程序及其他内容。这个平台类似于Apple给Iphone用户提供的**AppStore**，可以连接Google在线服务器，获取最新的应用程序。

Google Android 平台负责人 Eric Chu 称，**Android Market**对应用程序开发者最大的吸引力是，他们可以很轻松地进入这个"市场"。他写道："与 YouTube 相似，开发者仅需三个简单的步骤即可在上传内容：注册为'商人'、上传内容并加以描述、发布。我们选择'**Market**'这个字眼而不是'**Store**'的原因是，我们觉得应当为开发者创造一个开放的、畅通无阻的环境来发布内容。"

Google表示不会像苹果那样对iTunes AppStore内的软件应用进行严格的监管，这也是Android Market称为“Market”的意图。软件开发商只需要注册并一次性支付25美元的申请费，之后就可以在Android Market上销售他们的软件，并从每次交易中获得70%的利润，这类似于苹果的销售模式，不同的是，交易所得的其它利润将属于运营商而不是Google。



该软件平台刚推出时只有50种应用软件，截至到Google2009年6月5日的开发者技术大会上李开复博士的介绍，已经超过4900个应用。

### 2.2 特色

与苹果相同，**Android Market**的所有内容也被存储在其开发商谷歌的服务器上。而且谷歌也对付费和市场非常重视，但是谷歌也有自己的独到之处：（本段内容来源网上，版权不详。）



## 市场而非商店

Android移动平台负责人埃里克-朱(Eric Chu)表示, Android Market是一个“开放的内容分发系统, 它可以帮助使用Android操作系统的手机终端用户寻找、购买、下载和安装各种各样的内容。”这是一个令人兴奋的平台。与微软和苹果平台不同, Android Market可以为用户提供多样化的内容, 既有来自各大媒体公司的内容, 也有来自业余开发者的程序, 从而消除二者之间的隔阂。谷歌拥有足够的技术实力来完成这种服务。他们专业的服务器和搜索算法将会使得Android Market的优势更为突出。

## Web 2.0社区功能

Android Market在默认状态下就整合有Web 2.0社区功能。谷歌一直以来都致力于为人们提供相关的工具以帮助用户建立社区。谷歌表示, Android Market将会配有与YouTube类似的反馈和评分系统。虽然苹果的应用商店也拥有这两项功能, 但是, 如何运营这些功能才是最重要的。由于苹果的控制更为严格, 因此谷歌的透明将会吸引更多的用户。

## 发布更为简便

要在苹果应用商店发布一款iPhone应用商店相当繁琐。苹果将会对每个iPhone应用程序进行审查, 这就使得许多开发者不得不等上几天, 甚至几周的时间。有时, 开发人员已经开发出了新版的应用程序, 但是老版的软件却还没有被审核通过。谷歌则表示, Android Market的内容发布流程与YouTube类似: 你只需要注册成功, 然后上传内容并对其加以描述之后即可发布。

## 数据分析工具

iPhone开发者虽然也可以从苹果那里获得每日的消费统计, 但是这些都是非常基本的数据。除了用户ID、所购应用程序的ID、付款方式以及付款金额之外, 就没有任何的有用信息了。开发者从一开始就对这项服务颇有微辞, 而这也恰恰给了谷歌以机会。他们在数据统计方面有着丰富的经验, 同时也能够提供相应的软件来为开发者提供更加复杂的分析。

## 巨大的市场

只用了一个月的时间, 苹果应用商店的销量就达到了惊人的3000万美元。如果以这样的速度来计算, 苹果应用商店全年的销售额将达到3.6亿美元。苹果自己也认为他们应用上点将会很快成为一个过亿的大生意。但是, 他们的客户群却仅仅局限于600万iPhone用户。尽管iPhone用户的一向以出手大方而著称, 但是对更多的手机和更多的内容开放的Android Market平台完全有能力超越苹果应用商店。

但是Android Market的成功还未板上钉钉, 在它真正壮大之前, 谷歌还有几个问题需要解决。例如, 并不是所有的Android Market的功能都会从发布当日起开始生效。谷歌方面表示, 首批的Android手机将会配有Android Market的测试版, 用户至少可以通过这一版本获得免费的应用程序。而新版的支持付费内容及其他功能的Android Market将会“随即”推出。为用户提供测试版的应用程序可不是个明智之举。

另外, Android Market希望能够提供尽可能多的内容, 因此谷歌不会对内容进行监控。由于缺乏对垃圾信息的过滤, Android Market将会毫无疑问破坏用户体验。

尽管还存在许多问题, 但是我们相信, 拥有上述的五大优势的Android Market将会获得一个良好的开端。而且, 但从功能性上来讲, 这项服务无疑是新鲜独特的, 这就使得它具有足够的实力与苹果应用商店和微软Zun Marketplace竞争。

## 2.3 申请帐号

要想将自己开发的应用上传到Google Android Market上, 你必须申请market帐号, 过程如下。

打开<http://market.android.com/publish/signup>页面, 输入基本信息

## Getting Started

Before you can publish software on the Android Market, you must do three things:

- Create a developer profile
- Pay a registration fee (US\$25.00) with your credit card (using Google Checkout)
- Agree to the [Android Market Developer Distribution Agreement](#)

### Listing Details

Your developer profile will determine how you appear to customers in the Android Market

Developer Name

1sters.com

Will appear to users under the name of your application

Email Address

1sters.com@gmail.com

Website URL

http://www.1sters.com

Phone Number

Include country code and area code. [why do we ask for this?](#)

Email updates

☒ Contact me occasionally about development and Market opportunities.

[Continue »](#)

填写完后，点击“Continue”进入支付引导页面。



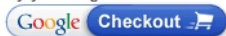
[1sters.com@gmail.com](#) | [Home](#) | [Help](#) | [Android.com](#) | [Sign out](#)

## Register as a developer

**Registration fee: US\$25.00**

Your registration fee enables you to publish software in the market. The name and billing address used to register will bind you to the [Android Market Developer Distribution Agreement](#). So make sure you double check!

Pay your registration fee with



Fast checkout through Google

[Continue »](#)

选择“Continue”进入支付界面，需要输入你的信用卡号和CVC码，需要注意其国家选项中尚无“china”，这个不要紧，选择“Hong Kong”就可以了。





https://checkout.google.com/view/buy?o=shoppingcart&shoppingcart=942196210315103

☐ Keep my email address confidential.  
Google will forward all email from Android Market to iceskysil@gmail.com. [Learn more](#)


☒ I want to receive promotional email from Android Market.

Pay with: Your current payment method is not valid  
Please enter a new payment method to complete this purchase.

Location: Hong Kong

Card number: 4392-2670-  
   

Expiration date: 09 / 2010 CVC:  [What's this?](#)

 The CVC is a 3-digit number in the signature panel on the back of your Visa card.

Name: yad

Address Format: [Switch to domestic format](#)

Address: 6/F, IE  
enzhen Jark

City/Town: shenzhen

Area: HONG KONG

Phone number: 13  
Will be used in case of order problems and for account verification.

OK


Place your order now -- \$25.00

**Billing Information & Privacy**  
Your credit card will be charged by Google. "GOOGLE \* Android Market" will appear by the charge on your credit card statement. [Learn more](#)

资料填写完成后，下到其确认页面

← → ↻ ☆ <https://checkout.google.com/view/buy?o=shoppingcart&shoppingcart=942196210315103>

iceskysl@gmail.com | [Help](#) | [Sign out](#)



Change Language English (US)

Order Details - Android Market, 1600 Amphitheatre Parkway, Mountain View, CA 94043 US

Qty	Item	Price
1	Android - Developer Registration Fee for iceskysl@gmail.com	\$25.00

Estimated Tax : \$0.00

**Total: \$25.00**

☒ Keep my email address confidential.  
Google will forward all email from Android Market to iceskysl@gmail.com. [Learn more](#)

☒ I want to receive promotional email from Android Market.

Pay with: VISA xxx-1417 - [Change](#)


[Place your order now -- \\$25.00](#)

**Billing Information & Privacy**  
Your credit card will be charged by Google. "GOOGLE \* Android Market " will appear by the charge on your credit card statement. [Learn more](#)

确认后开始支付，支付完成后显示成功消息

← → ↻ ☆ <https://checkout.google.com/confirmation?t=957706465087903>

iceskysl@gmail.com | [Help](#) | [Sign out](#)




**✓ Thanks yaoshanglang, you're done!**  
Your order has been sent to Android Market. [Return to Android Market >](#)

**Message from Android Market:**

Thanks for your interest in publishing your applications to Android Market. Please return to the [Android Market Developer Site](#) to finish your registration

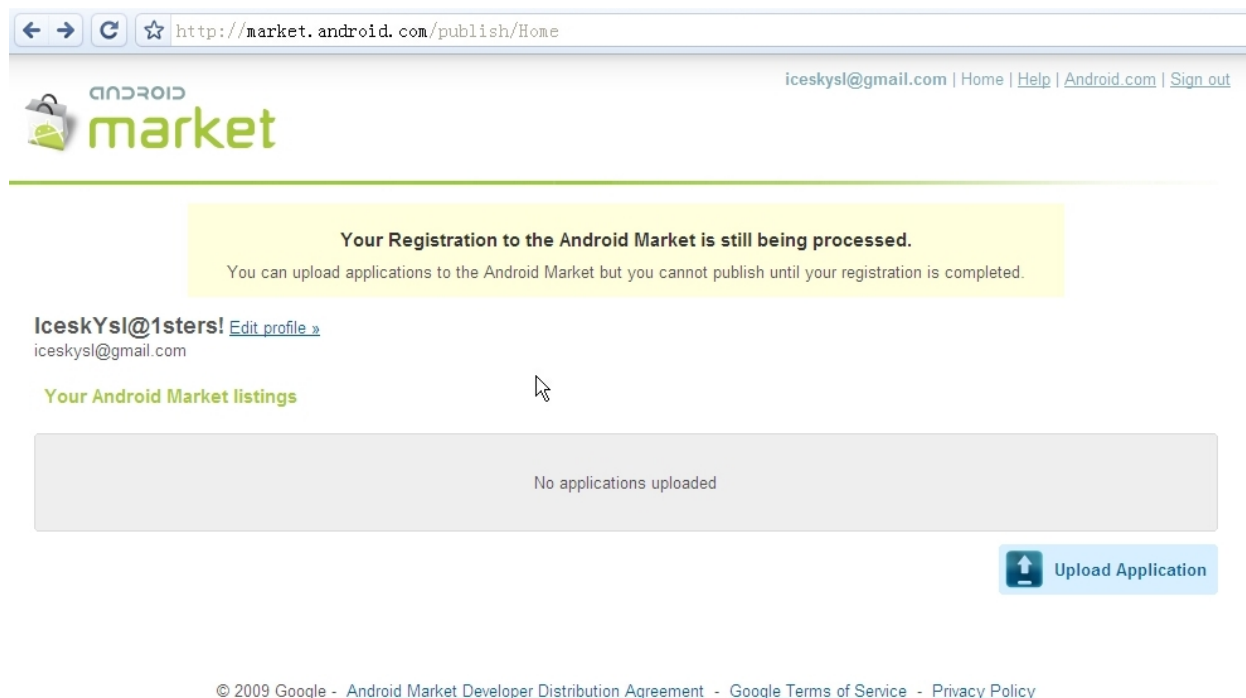
**How do I track my order?**  
[Get up-to-date order progress](#) on checkout.google.com

**Buy in seconds every time you shop.**  
Just look for the Google Checkout button on [Android Market](#) or [Google Product Search](#).



©2009 Google [Terms of Use](#) - [Privacy Policy](#) - [Google Home](#)

紧接着就可以发布你的应用了。



到这里我们已经申请了自己的Android Market帐号，值得称赞的是，Android Market的帐号不需要审核，支付完成后，就开通了，就可以发布自己的应用了。

### 3. 准备发布：一份核查清单

(Preparing to Publish)

译者：404

原文地址：<http://developer.android.com/guide/publishing/preparing.html>

在发布一个应用程序之前，需要对它进行适当地测试然后将其打包，并确保最终能够运行在用户的Android移动设备上。

如果您计划发布应用程序并使其能够安装在Android设备上，您需要为您的应用程序做一些准备工作。为了您能够顺利地发行应用程序，本文着重突出了一些需要仔细核查的要点事项。

如果您打算将应用程序发布到Android商店，也可以参见 [Publishing on Android Market](#) 以便了解发行应用程序的具体事宜。

欲了解发布应用程序的一般方式，参见 [Publishing Your Applications](#) 这篇文档。

在您考虑发行应用程序之前：

1. 在真机上彻底地测试您的应用程序
2. 考虑在您的应用程序里边添加一份终端用户许可协议
3. 在应用程序的manifest文件中指定一个图标和名称
4. 关闭日志记录和调试功能并清除不必要的的数据或文件

在您最终编译应用程序之前：

5. 为您的应用程序设定版本
6. 生成一份合适的密钥
7. 如果您的应用程序用到MapView，需注册一个Maps API Key

编译您的应用程序...

当您的应用程序编译完成之后：

8. 签名您的应用程序
9. 测试经编译过的应用程序

#### 3.1 在您考虑发行应用程序之前

##### 3.1.1 在真机上彻底地测试您的应用程序

尽可能彻底且广度地测试您的应用程序是很重要的。Android 提供了多种测试类和工具可以帮助您做到这一点，您可以使用[Instrumentation](#)来运行Junit和其他测试用例，还可以使用测试工具如[UI/Application Exerciser Monkey](#)。

- 为了确保您的应用程序面向终端用户可用，您应该尽量提供一个或多个物理移动设备以便您的应用程序能够按预期运行。然后，您应该在這些真机上测试您的应用程序，且在实际的网络条件下。在物理设备上测试您的应用程序尤为重要，因为那样可以验证用户界面元件的尺寸是否正确（尤其是触摸屏UI）以及应用程序的性能和耗电情况是否在可接受的范围之内。

- 如果您没有可用于应用程序运行的移动设备，您可以使用模拟器选项诸如`-dpi`, `-device`, `-scale`, `-netspeed`, `-netdelay`, `-cpu-delay` 等等来模拟屏幕，网络性能及其他和终端设备相似的参数或属性。然后，您还可以测试应用程序的UI和性能。不过，我们还是强烈推荐您在发布应用程序之前在实际的终端设备上进行测试。
- 如果您使用 [T-Mobile G1](#) 作为终端来运行您的应用程序，确保UI能掌握屏幕方向的变化。

### 3.1.2 考虑在您的应用程序里边添加一份终端用户许可协议

为了保护您个人，组织，以及知识产权，您可能想要为您的应用程序提供一份终端用户许可协议（EULA）。

### 3.1.3 在应用程序的manifest文件中指定一个图标和名称

在应用程序中的manifest文件中指定图标和名称非常重要，因为它们将作为应用程序的图标和名称呈现给用户。它们还将显示在设备的主屏中，以及Manage Applications, My Downloads或别处。另外，其他发行机构可能将这些图标和名称呈现给用户。

为了指定一个图标和名称，您可以在manifest文件的`<application>`里边定义`android:icon` and `android:label`属性。

至于图标设计，您应该试着尽可能让它接近Android内置应用的风格。

### 3.1.4 关闭日志记录和调试功能并清除不必要的文件或数据

为了发行应用程序，您应该确保调试功能已关闭且已经从应用程序项目中移除了那些不必要的文件或数据。

- 从manifest文件的`<application>`里边去掉`android:debuggable="true"` 属性。
- 从应用程序项目中清除日志文件，备份文件，和其他多余的文件。
- 检查私有或独立数据，必要的话将它们移除。
- 在源代码中禁用所有能够调用 [Log](#) 的方法。

## 3.2 在您最终编译应用程序之前

### 3.2.1 为您的应用程序设定版本

在您编译应用程序之前，必须确保已经为它定义了版本号，即在应用程序的manifest文件的 `<manifest>`里边为`android:versionCode` 和 `android:versionName` 属性指定适当的值。请仔细考虑在您应用程序的升级策略中实行数字版本计划。

如果您已经发布一个应用程序的早期版本，必须更新当前应用程序的版本号。您必须将应用程序的manifest文件 `<manifest>`中的`android:versionCode` 和 `android:versionName` 属性的值递增到适当的值。



欲了解为应用程序定义版本号的详细信息，参见[Versioning Your Applications](#)。

### 3.2.2 生成一份合适的密钥

如果您已经遵循了以上步骤，您的应用程序已完成了编译并可以准备签名。这时候在.apk内部，您的应用程序应该有了合适的版本号，且您已经按如上所述清除了额外的文件和私有数据。

在您签名应用程序之前，必须确保有一个合适的私钥。欲了解关于生成私钥的完整信息，参见[Obtaining a Suitable Private Key](#)。

一旦您获得了一份合适的私钥，将在如下地方用到它：

- 如果您的应用程序用到了MapView元件，参照下面的方式注册一个Maps API Key。
- 为了发行您的应用作签名用，稍后讨论。

### 3.2.3 如果您的应用程序用到MapView，需注册一个Maps API Key

欲了解关于获取一份Maps API Key的完整信息，参见 [Obtaining a Maps API Key](#)。

如果您的应用程序用到了一个或多个MapView 元件，在这些MapView(s)将能够从Google Maps检索数据之前，您需要为应用程序注册Google Maps服务并获取一份Maps API Key。要实现这一点，您要为地图服务中心提供一份证书署者的MD5签名。

开发期间，您可以注册一份通过SDK 工具生成的debug key作为临时的Maps API Key来使用。不过，在应用程序发布之前，您必须基于私钥注册一份新的Maps API Key。

如果您的应用程序用到了MapView 元件，请注意以下几个要点事项：

1. 您必须在编译应用程序之前取得一份Maps API Key，因为您必须将此Key加入到每个MapView元件特定的属性中——`android:apiKey`——在您的应用程序的布局文件里边。如果您直接从代码中实例化MapView对象，您必须将Maps API Key作为构造器的参数传入。
2. 应用程序中的MapView元件引用的Maps API Key必须已经使用签名用的证书注册过（在Google Maps中）。这点在您发布应用程序时尤为重要——MapView元件所引用的Key必须是通过您将用于签名应用程序的发行证书注册过的。
3. 如果您之前已经使用SDK 工具生成的调试证书来注册过一份临时的Maps API Key，记得必须使用发行证书来注册获得一份新的Maps API Key。您还必须记得改变MapView元件来引用新的Maps API Key，而不是关联通过调试证书注册的那个Key。假使您不那样做，您的MapView元件将没有权限下载地图数据。
4. 如果您将要改变用于签名应用程序的私钥，必须记得从Google地图服务中心获取一份新的Maps API Key。如果您未取得这份新的Maps API Key，或者没有将这份新的Maps API Key应用到所有的MapView元件中，任何关联旧Key的MapView元件将没有权限下载地图数据。

欲了解更多关于签名应用程序和私钥的信息，参见[Signing Your Applications](#)。



### 3.3 编译您的应用程序

当您已经参照前面章节中所述的那样准备好了应用程序；此时，您可以编译应用程序用于发行。

### 3.4 当您的应用程序编译完成之后

#### 3.4.1 签名您的应用程序

用您的私钥签名您的应用程序。使用正确的方式签名您的应用程序是相当重要的。请参考[Signing Your Applications](#)这章了解完整信息。

#### 3.4.2 测试经编译过的应用程序

在您发行经编译过的应用程序之前，您应该彻底地在终端移动设备上（以及在现实的网络条件下，假如可以）进行测试。尤其是，您应该确保UI中的任何MapView元件能够正常地接收地图数据；如果不行，返回至 [Register for a Maps API Key](#) 这一章并纠正这个问题。同时，您必须确保应用程序能够结合服务端数据正常地工作，并且应用程序必须是通过正确的身份认证来使用这些数据。

测试通过后，您就可以准备发行您的应用程序供持有移动设备的用户使用了。

## 4. 应用程序签名

译者：apcwowo

原文：<http://developer.android.com/guide/publishing/app-signing.html>

这个文档主要讲述了在你发布程序给移动设备用户前如何对你开放的应用程序进行签名。

### 4.1 概述

Android系统要求所以安装的程序必须有应用程序开发者认证的数字签名的私人密钥。Android系统用这个证书来识别作者并建立与应用程序的信用关系。证书不是用来控制哪个用户哪个应用程序可以安装。证书并不需要官方机关的签名。这是完全许可的。典型的情况是Android用自签名证书。

- 所以的应用程序必须签名。如果应用程序没有签名，系统将无法安装此程序。
- 你可以使用自签名证书来签名你的软件，不需要官方机构的认证。
- 当你发布你的程序给最终用户时，你必须有一个合适的密钥来签名它，你不能发布一个用SDK工具生成的调试密钥签名的应用程序。
- 系统在安装程序的时候检查签名证书是否过期，如果在安装应用后签名证书过期，这个应用程序还是会正常运行的。
- 你可以使用标准工具 ---Keytool 和Jarsigner----来生产密钥来签名你的apk文件。

Android系统不会去安装或运行一个没有签名的应用程序，这适应于任何在Android系统上运行的任何应用程序。无论是实际设备还是模拟器，基于上面的理由，你必须签名你的应用程序，这样你才可以能在模拟器或设备上调试它。

当你调试时Android SDK工具可以帮助你签名你的应用程序。Eclipse的ADT插件 和Ant 构建工具提供了两个签名模式 调试模拟 和 发布模式。

- 在开发和测试的期间，你会在调试模式编译。在调试模式，构建工具使用的是Keytool功能模块，它在JDK中包含，通过已知的别名和密码去创建一个密钥库和密钥，在每一个编译中，这工具用调试的密钥去签名你的应用程序的.apk文件,因为密码是可知的，工具不会在你编译的时候提示你密钥库和密钥的密码。
- 当你准备发布你的程序时，你必须在发布模式编译然后用私有密钥去签署.apk文件，通过以下方法来做到这点：
  - 在命令行使用Keytool 和Jarsigner. 这种方法，你先编译一个没有签名的apk文件。然后你必须手动通过Jarsigner（或其它相同的工具）为apk文件签名私有密钥。如果你没有一个合适的私人密钥，你可以先用Keytool生成一个密钥库，然后在拿Jarsigner来签名你的应用程序。
  - 使用ADT 向导。如果你的开始环境是在有ADT插件的Eclipse中，你可以使用导出向导去编译你的程序，生成你的私有密钥，签名应用程序，所以的过程都会在导出向导中处理。

### 4.2 签名策略

程序的签名在很多方面会影响开发应用程序的方法，特别是，如果你计划发布多个应用程序一般来说，建议的策略是，所有的应用程序拿相同的证书去签名。贯穿你应用程序的期望周期，需要做的理由有几个：

- 应用程序升级 ----当你对你的应用程序进行升级, 你会拿相同的证书签名更新程序, 当系统正在安装一个应用程序的更新程序, 如果你想在新的版本中无缝升级, 如果其中任何一个证书的新版本与任何旧版本的证书匹配, 系统允许你去更新。如果你签名的版本没有被匹配, 你可能需要为应用程序指定一个不同的包名。在这种情况下, 用户将安装一个新版本的完整的应用程序。
- 应用程序模块性----**Android**系统允许相同证书签名的应用程序运行在同一个进程中。如果应用程序请求, 那系统将以单一的一个应用程序来看待, 以这种方式, 你可以开发模块化的应用程序, 而且可以更新每一个独立的模块, 如果你需要。
- 代码/数据的共享----**Android**系统提供了基于签名的执行权限。当用特别的证书签名后, 一个应用程序可以暴露功能给另一个应用程序。通过拿相同证书签名多个应用程序和基于签名执行权限的检查, 你可以在安全的方式下共享代码和数据。

另外决定以那种策略来签名文件需要重点考虑的是如何拿一个有效周期的密钥来签名你的应用程序。

- 如果你的应用程序支持更新。你应该确保你的密钥的有效周期能满足应用程序期望的生命周期。建议有效周期为**25年**或者更多, 如果你的密钥的有效期过期, 用户将无法无缝更新应用程序的新版本。
- 如果你要拿同样的密钥来签名多个不同的应用程序, 你应该确保你密钥的有效周期超过所以版本应用程序的生命周期, 包括将来应用程序需要添加的程序。
- 如果你计划在**Android**的商店发布你的应用程序。你密钥的有效周期到**2033年10月22日**, 商店强制要求这个属性确保用户可以无缝更新应用程序的新版本。

正如设计你的应用程序, 注意上面所提到得点确保拿一个合适的证书去签名你的应用程序。

### 4.3 签名的基本设置

为了支持生成密钥库和调试用的密钥, 所以第一步你应该确认**SDK**的构建工具**Keytool**可用。在大多数情况下, 你要确保告诉**SDK**构建工具如何去找**Keytool**, 你可以通过设置环境变量**JAVA\_HOME**来引用一个合适的**JDK**, 另外你还可以把**Keytool**的**JDK**版本号加入**PATH**的环境变量中来解决这个问题。

如果你在**Linux**的版本下开发, 使用的是原始的**GNU**编译器编译**JAVA**, 你必须确保使用正确的**Keytool**的**JDK**版本, 而不是**GCI**版本。如果你的**Keytool**的路径已经放到了**PATH**环境变量中。它是一个指向**/usr/bin/keytool**的符号链接, 这种情况下, 你应该确保这个符号链接的目标是指向正确的**Keytool**..如果你要发布你的应用程序, 你的机器上还需要有**Jarsigner**, **Jarsigner**和**Keytool**都包含在**JDK**中。

### 4.4 调试模式下的签名

**Android**构建工具提供了在调试模式下你签名机制, 这样可以让开发人员更容易的去调试应用程序。同时还满足了**Android**系统在模拟器和设备上安装程序时必须签名**apk**文件的要求。当在调试模式下时, **SDK**工具调用 **Keytool**来创建密钥库和密钥。

通过预设的名字跟密码**SDK**工具来创建密钥库跟密钥。

- 密钥库名字 - "debug.keystore"
- 密钥库密码 - "android"
- Key alias - "androiddebugkey"
- 密钥别名 - "androiddebugkey"
- 密钥密码 - "android"
- CN - "CN=Android Debug,O=Android,C=US"

如果需要, 你可以改变调试模式下的密钥库跟密钥的名字也可以自定义一个。在**Eclipse/ADT**的环境下, 你可以在菜单**Windows>prefs>Android>Build**下设置, 然而, 任何自定义的调试用的密钥库/密钥必须使用相同的名字跟密码作为默认调试的密钥。

注意：你不能拿调试模式下签名的证书来发布正式的应用程序。

#### 4.4.1 Eclipse 用户

如果你在有ADT插件的Eclipse的环境下开发而且已经通过上面说的设置好了Keytool.默认模式是在调试模式下签名文件。当你运行或调试你的程序，ADT将拿调试模式的证书签名你的.apk文件，然后在模拟器上安装你的程序，但这个前提是，ADT能成功访问到Keytool.

#### 4.4.2 Ant 用户

如果你在Ant下构建你的.apk文件。通过设置debug选项来设置调试签名模式为可用。如果你使用Android的工具build.xml文件来生成。当你在Ant的调试模式下编译你的程序，build脚本会生产密钥库跟密钥来签名你的.apk文件，其它需要注意的地方可以参见[Developing In Other IDEs: Building in debug mode](#)

#### 4.4.3 调试证书的期限

自签名证书用来签名调试模式下的应用程序（Eclipse/ADT Ant 默认选项），它们的使用周期为生成日期后的365天。

如果一个证书过期，你会得到构建错误，在Ant上构建，你会看到如下错误：

```
debug:
[echo] Packaging bin/samples-debug.apk, and signing it with a debug key...
[exec] Debug Certificate expired on 8/4/08 3:43 PM
```

在Eclipse/ADT中，你也会在Android的控制台中看到类似的错误。

解决这个问题，可以简单的把调试密钥库删除掉。它默认保存在AVD目录，在OS/Linux中为~/.android/avd 在XP中是C:\Documents and Settings\\*.android\

在Window Vista在C:\Users\\*.android下。

在下次构建的时候，构建工具将会再生成新的密钥库跟调试密钥。

需要注意的是，如果你开发机中的日期非阳历，那构建工具可能产生不正确过期的证书，当你编译你的程序时会发生错误，关于可行方案，可以参见疑难解答专题[I can't compile my app because the build tools generated an expired debug certificate.](#)

#### 4.4.4 签名正式发布软件

当你准备发布程序给其他用户，你必须：

- 1.在发布模式下编译程序
- 2.得一个合适的私有密钥
- 3.拿私有密钥签名你的应用程序

在下面的章节中会讲如何做到这几点：

如果你使用的是有ADT插件的Eclipse，你可以使用导出向导用私有密钥来签名你的应用程序。导出向导甚至可以允许你生产一个新的密钥库跟私有密钥。可以参见：[Compiling and signing with Eclipse ADT](#)

### 4.5 在发布模式下编译

准备发布你的应用程序，你第一步必须在发布模式编译你的程序。Android构建工具会编译你的应用程序，但不拿调试密钥签名你的程序。

注意：你不能发布没有签名或拿调试密钥签名的应用程序

#### 4.5.1 Eclipse 用户

在Eclipse中导出没有签名的.apk文件。在右键点击工程并选择**Android Tools>Export Unsigned Application Package**,然后只需要指定保持文件的位置。（或者你可以在Eclipse中打开AndroidManifest.xml文件，打开预览标签，点击**Export an unsigned .apk**）你也可以在导出向导中结合编译与签名 参见：[Compiling and signing with Eclipse ADT](#).

#### 4.5.2 Ant 用户

如果你用的是Ant，你需要在Ant 命令中特别指定构建目标为发布。例如你可以运行目录下build.xml，命令就像下面一样：

```
$ ant release
```

脚本会编译构建无签名的应用程序

#### 4.5.3 获得适合的私有密钥

准备签名你的应用程序。你第一步必须确保你有一个合适的私有密钥，一个合适的私有密钥指的是：

- 你是否持有它。
- 通过应用程序能识别是个人，团队 还是组织。
- 密钥的周期是否超过了应用程序或应用程序套件的生命周期。建议密钥的周期为25年或更长。

如果你计划在Android商店在发布你的应用程序。注意你密钥周期必须大2033年10月22,如果你密钥的周期在这个日期之前那你将无法上传你的应用程序。

- Android SDK工具生成的不是调试用密钥。

如果你没有一个合适的密钥。你必须拿Keytool去生成一个。确保你Keytool工具是可用的可参见上面在Basic Setup中描述的。

拿Keytool工具生成一个自签名的密钥。可以在keytool的命令行传递下面列表中的参数：

注意：在你运行Keytool前，确保你读了[Securing Your Private Key](#), 讨论如何保护你密钥的安全和这么做对你和用户的重要性。尤其是，当你生成你的密钥时，你应该选择给密钥库跟密钥选择一个比较强壮的秘密。

Keytool 选项	描述
-genkey	生成一对密钥（公开的和私有的）
-v	启用详细输出
-keystore <keystore-name>.keystore	包含私有密钥的密钥库的名字
-storepass <password>	密钥库的密码 作为一个安全防范措施。不要在命令行包含这个操作除非你工作在一个安全的电脑上。如果没有提供。Keytool会提示你输入密码，以这种方式，你的密码不会保存在Shell的历史中。
-alias <alias_name>	密钥的别名
-keyalg <alg>	通过哪种算法来生产密钥，DSA 和RSA都支持。
-dname <name>	一个区别名字来说明谁创建了这个密钥。该值在自签名证书中作为issuer和subject字段。 注意你不需要在命令行中指定这个值，如果没有提供，Jarsigner会提示你输入区别名字段(CN,OU 等等)
-validity <valdays>	密钥有效周期天数 注意：建议是10000或更大。
-keypass <password>	密钥的密码

作为一个安全防范措施。不要在命令行包含这个操作除非你工作在一个安全的电脑上。如果没有提供。Keytool会提示你输入密码，以这种方式，你的密码不会保存在Shell的历史中。

下面是一个例子用Keytool命令来生产一个私有密钥：

```
$ keytool -genkey -v -keystore my-release-key.keystore  
-alias alias_name -keyalg RSA -validity 10000
```

运行上面例子中的命令。Keytool会提示你提供密钥库和密钥的密码和密钥的区别名称字段。然后它会生产一个密钥库文件叫my-release-key.keystore.这个密钥库和密钥通过你输入的密码来保护。这个密钥库包含一个密钥，有效期为10000天。你给它设置的别名，将会在签名你的应用程序时引用这个密钥库。关于Keytool的更多信息，可以参见：<http://java.sun.com/j2se/1.5.0/docs/tooldocs/#security>

## 4.6 签名你的应用程序

当你准备确实签名你发布的.apk文件时。你的使用Jarsigner工具。确保这个工具在你的机器上是可用的。在Basic Setup已经有描述。同时确保你的密钥库包含的私有密钥是可用的。签名你的应用程序。你运行Jarsigner，引用你要签名应用程序的.apk文件和含有私有密钥的密钥库。下面的表说明各个选项的功能。

Jarsigner 选项	描述
-keystore <keystore-name>.keystore	包含私有密钥的密钥库
-verbose	支持详细输出
-storepass <password>	密钥库的密码 作为一个安全防范措施。不要在命令行包含这个操作除非你工作在一个安全的电脑上。如果没有提供。Keytool会提示你输入密码，以这种方式，你的密码不会保存在Shell的历史中。
-keypass <password>	私有密钥的密码 作为一个安全防范措施。不要在命令行包含这个操作除非你工作在一个安全的电脑上。如果没有提供。Keytool会提示你输入密码，以这种方式，你的密码不会保存在Shell的历史中

如何用Jarsigner去签名一个叫my\_application.apk的应用程序包。通过上面的密钥库创建。

```
$ jarsigner -verbose -keystore my-release-key.keystore  
my_application.apk alias_name
```

运行上面的命令行。Jarsigner会提示你提供密钥库和密钥的密码。它然后修改.apk文件，意思是将签名这个apk文件。

注意，你不能拿不同的密钥在不同的时间签名一个.apk文件。

去验证你的.apk文件是否签名。你可以使用下面的指令：

```
$ jarsigner -verify my_signed.apk
```

如果.apk文件已经签名，Jarsigner会打印“jar verified”，如果你想看更多的信息。你可以试一下下面的指令：

```
$ jarsigner -verify -verbose my_application.apk
```



或

```
$ jarsigner -verify -verbose -certs my_application.apk
```

上面的命令，添加了一个-certs的操作。它将显示“CN=”字段来显示谁创建了这个密钥。

注意：如果你看到得是“CN=Android Debug”它的意思是这个apk文件是拿Android SDK生成的调试用密钥签名的。你如果你想发布你的程序，你必须拿私有密钥代替调试密钥来签名它。

关于Jarsigner的更多信息：参见：<http://java.sun.com/j2se/1.5.0/docs/tooldocs/#security>

当你使用的是有ADT的Eclipse.你可以使用导出向导来签名你的apk文件。导出向导会配

合keytool 和Jarsigner，允许你通过图形界面取代命令行来签名应用程序。应用导出向导使用的

是Keytool Jarsigner，所以你应该确保它们能在你机器上正确访问。具体的描述在上面[Basic Setup for Signing](#).

去创建一个签名的apk文件。在Package Explorer的工程上单击右键选择**Android Tools > Export Signed Application Package**(或者你可以打开AndroidManifest.xml文件，打开预览，单击**Use the Export Wizard**) 窗口上会出现任何在导出过程中的错误。如果没有错误发生。导出过程继续执行。这将引导你完成应用程序的签名。包含的步骤有选择哪个密钥去签名apk文件。或创建一个新的密钥库和私有密钥。

当你完成了导出向导。你就有了一个可以发布的签名apk文件了。。

## 4.7 保护你的私有密钥

对于你跟用户来说维护你密钥的安全是很重要的。如果你允许某人使用你的密钥。如果你把密钥库跟密码放在一个不安全的地方。第三方会找到它们并使用。这样用户对作者的识别和信用将受到伤害。

如果第三方在你不知道和为允许的情况下使用你的密钥。每个人都可以签名和发布恶意的应用程序来破坏作者的应用程序。这样的人也可以签名和发布应用程序用你的身份，攻击其他应用程序或系统本身，或破坏或窃取用户数据。

- 你开发的声誉依赖的是你如何保护好私有密钥，在所有时间，直到密钥过期。下面有保护密钥的几点建议
- 对于密钥库跟密钥选择强壮的密码
- 当你拿Keytool来生成你的密钥时。不要在命令行操作-storepass和-keypass，如果你这么做了，您的密码将在您的shell的历史中，任何用户在您的计算机可以访问
- 同样的，当你拿Jarsigner签名你的应用程序时，也不要再在命令行操作-storepass和-keypass
- 不要给或借给任何人你的私有密钥。不要让未经授权的人知道您的密钥库和密钥密码。

一般来说，如果你按照上面的预防常识来生产 使用 保存你的密钥，这将是安全的。

## 5. 发布你的应用

(Publishing Your Applications)

翻译: haiyangjy (<http://www.haiyangjy.com/>)

发布应用意味着, 你要测试你的应用, 并且正确的打包应用, 并将它呈现给使用Android系统的用户, 让用户下载使用。

如果你按照“准备发布你的应用”中列出来的步骤执行的话, 那么处理到最后应该能得到一个编译并且签名的.apk文件。在.apk文件内部, 应用包含了正确的版本好, 如果用到了MapView的话, 那么应用也需要包含Maps API Key。现在你的程序就可以发布了。

接下来的部分主要说明了怎么样发布你的应用让用户使用。

### 5.1 发布到Android Market上

Android Market 是一种线上服务, 它使得用户可以很方便的寻找并且从Android Market 下载应用到自己的基于Android的设备上, 它使得开发者很方便地发布应用到终端用户。

为了将你的应用发布在Android Market 上, 首先你需要同意注册条款并用Google帐号注册这项服务, 一旦你注册了, 你可以在任何时候将你开发的应用上传到Android Market , 并且没有次数限制, 一旦你准备好了, 那么就可以直接发布应用。一旦你发布了, 用户就可以看到你的应用, 并且可以下载, 而且还可以用安装在基于Android设备里边的Market应用程序来打分。

如果你注册成为Android Market 的开发者开始发布应用, 那么就访问Android Market 的官方网站:

<http://market.android.com/publish>

如果你打算在Android Market上发布你的软件, 你需要确保你的应用满足以下列出来的要求。当你上传应用到Android Market的服务器的时候, 这些要求会被硬性检查。

Android Market强制的强制要求

- 你的应用必须进行密钥签名。并且密钥的有效期必须为2033年10月22日后。
- 应用的manifest文件里的<manifest>元素中必须定义android:versionCode 和 android:versionName属性。服务器利用android:versionCode属性来处理你的应用是否更新过, 并且会把android:versionName属性的值做为程序的版本显示给用户。
- 你的程序必须在manifest里边包含 android:icon和android:label attribute属性。

### 5.2 在Android Market上发布更新

beta版的Android Market在当你发布一个应用更新后并不支持主动提示你的用户。这个功能将会马上提供, 但是现在用户必须手动更新程序。当你发布了应用更新后, 你可以提醒用户应用的更新已经可以用了, 并且提供给他们从Android Market升级应用的方法。

下边是我们提供的一个建议方法, 提示用户, 当应用的新版本发布的时候。

1. 让你的应用程序时不时的访问一下web-service。这个web-service应当返回两个值: 一个是程序最新的版本号, 也就是android:versionCode中定义的值, 另外一个是一个URI, 这个URI随后会被传到intent里边, 通过这个intent, 你就可以启动Market应用, 给用户搜寻最新版本的应用。



你的web-service返回的URI必须是正确格式的，以便用来搜索Android Market中的应用。具体可以参照“用Intents来启动Market应用。这个URI必须把要升级程序的包名作为查询参数，因为程序的包名在Market里边是唯一的。查询应用的URI格式为：`http://market.android.com/search?q=pname:<package>` 或者 `market://search?q=pname:<package>`

2. 你的应用这时可以把自身的版本号和从web-service处获得的版本号做一个比较。如果得到的版本号比较大的话，你的程序应该弹出来一个对话框来提示用户当前应用有一个新的版本可用。对话框应该提供下载和取消的按钮。

3. 如果用户点击了开始下载的按钮后，你的程序应该调用startActivity(), 传进来的包含ACTION\_VIEW 和从web service处获得URI的Intent. 这个Intent启动设备上Market程序，并且利用传进来的查询参数构造一次查询请求。当查询结果显示的时候，用户可以看到关于新版应用更加详细的信息，并且可以下载这个应用。

注意，由于这个URI是从web-service获得的，而不是硬编码到程序当中，所以你可以在用改变程序的情况下可以轻松的改变Market启动的方式。

### 5.3 利用Intents来启动设备上的Market应用程序。

基于Android的设备包含一个预装的Market应用，这个应用可以让用户访问Android Market。在Market应用当中，用户可以浏览并且搜索可用的应用、阅读评价、并且可以下载安装。

通过给系统发送一个Intent你可以启动Market应用程序。比如你可以帮助用户来定位并下载更新已经安装过的程序。或者让用户知道和次程序相关的程序是否可以下载。

为了启动Market应用，你发出一个ACTION\_VIEW的Intent，并且将Market可以处理的URI字符串作为data放在Intent里边。在大多数情况下，你的应用应该调用startActivity() 来发送ACTION\_VIEW的Intent。

你在Intent当中提供的URI可以让系统正确的路由intent。intent当中的action指出了当启动Market后，Market应该怎么来执行。目前，当启动Market的时候，你可以让Market基于你提供的查询参数初始化一个查询，比如你可以通过一下的参数来查询程序：

- 包名
- 开发者的姓名
- 应用名、开发者名、描述中的字符串
- 以上的组合

下表格提供了目前Market应用支持的URI以及对应的actions

操作结果	URI和Anction	注释
通过应用的java包名来查询，并且显示出来结果	<code>http://market.android.com/search?q=pname:&lt;package&gt;</code> or <code>market://search?q=pname:&lt;package&gt;</code>	查询应用的包名，只返回精确的匹配。
通过开发者的名字查询应用，并显示结果	<code>http://market.android.com/search?q=pub:"&lt;Developer Name&gt;"</code> or	通过开发者的姓名进行匹配，只返回精确匹配结果。

	<code>market://search?q=pub:"&lt;Developer Name&gt;"</code>	Searches all public fields (application title, developer name, and application description) for all applications. Returns exact and partial matches.
通过子字符串来查询应用，并显示结果	<code>http://market.android.com/search?q=&lt;substring&gt; or market://search?q=&lt;substring&gt;</code>	查询所有程序公开的所有字段（程序的标题，开发者的名字，程序的描述）返回精确和部分匹配的所有结果。
利用多参数查询，并显示结果	<p>Example:</p> <code>http://market.android.com/search?q=world pname:com.android.hello pub:Android</code>	返回适合的所有结果。

## 6. 对应用程序进行版本管理

(Versioning Your Applications)

翻译: Edison0951

原文地址: <http://developer.android.com/guide/publishing/versioning.html>

- 您的应用程序必须拥有版本信息
- 您应该在应用程序的Manifest文件中设置版本信息
- 应用程序的版本信息将影响用户的更新
- 考虑到应用程序在将来会持续发行, 在开发中您应该尽早地确定应用程序的版本控制策略。

### 6.1 文档包含

1. 设定应用程序的版本
2. 指定最低的系统API版本

### 6.2 也包含

1. 准备发布您的应用程序
2. 发布在Android商店
3. The AndroidManifest.xml File

版本信息在您应用程序的升级/维护策略中是非常重要的一部分

- 用户需要知道已经安装在他们设备中的应用程序版本和可用于安装的新版本的具体信息
- 其它的一些应用程序-包括作为一个套件发布的-需要查询系统来获取版本信息从而决定兼容性和依赖性。
- 发布您的应用程序的服务可能也需要查询您的应用程序的版本信息, 以便它们能够显示给用户。为了能够确定兼容性和建立上下级的依赖关系, 一个发布服务可能也需要去检查应用程序的版本信息。

Android系统本身不会去检查一个应用程序的版本信息, 例如强迫限制更新和兼容性等等。相反, 只有用户或者应用程序本身会去检查应用程序的任何版本限制。

如果一个应用程序在Manifest中设置了minSdkVersion属性, Android系统会去检查系统的兼容性。这也允许应用程序去确定它所兼容的最低系统API。更多信息请查看[设定应用程序的版本信息](#)

### 6.3 设定应用程序的版本信息

您可以通过设置应用程序的Manifest文件的属性来定义应用程序的版本信息。您只需要设定下面两个属性就可以了

- **android:versionCode**--用一个整数来表示应用程序的版本。用整数的目的便于其它应用程序能自动评估, 比如确定上下级依赖关系。您可以设定任何您想要的整数, 只需要确保后续发布的应用程序的版本值更大。虽然系统不会去检查这个, 但是必须遵循后续发布的应用程序版本值要递增的规则。您可以设置versionCode为1来作为您应用程序的第一个版本, 只要有一次更新, 这个值就依次递增, 无论更新是大或者小。也就是说android:versionCode的值与显示给用户的版本信息(见android:versionName)没有必然的联系。应用程序和发布服务都不应该把这个值显示给用户。
- **android:versionName**-用一个字符串值来显示应用程序的版本信息, 而且它应该被显示给用户。用字符串是便于用<major>.<minor>.<point>描述应用程序版本, 也可以用其它相关的定义来描述。与

android:versionCode相比，这个值除了显示给用户，没有任何内在的目的。发布服务也可能用android:versionName去显示给用户

您可以在manifest文件的<manifest>元素中设定这两个值，下面是一个展示如何在<manifest>元素中设置android:versionCode和android:versionName的例子

```
<?xml version="1.0" encoding="utf-8"?>
  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.package.name"
    android:versionCode="2"
    android:versionName="1.1">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
      ...
    </application>
  </manifest>
```

在这个例子中，android:versionCode的值显示当前.apk是应用程序的第二次发布，同时在android:codeName中的字符表明这仅仅是一个小的更新。

Android框架提供一个API让应用程序在系统中查询您的应用程序的版本信息。为了获取版本信息，应用程序用[PackageManager](#)的[getPackageInfo\(java.lang.String, int\)](#)方法。

## 6.4 指定最低的系统API版本

如果您的应用程序需要确定ANDROID平台的最低版本，您可以在应用程序的MANIFEST文件中设定版本为API Level。这样做可以确保您的应用程序运行在一个兼容的Android系统上。

- **android:minSdkVersion**-用一个整数来显示ANDROID平台的版本，在准备安装一个应用程序时，系统会去检查这个属性的值并与系统的版本比较。如果android:minSdkVersion的值大于系统的版本，系统将会阻止应用程序的安装。如果您没有在manifest中设置这个属性，系统会认为您的应用程序兼容所有的平台。

定义应用程序所需的最低系统平台，可以在<manifest>下添加一个<uses-sdk>元素，然后将属性设置为android:minSdkVersion。

如果需要更多的信息，可以参考[Android System Image 1.1 Version Notes](#)。

## 7. 使用Android Market管理面板

by: IceskYsl (<http://iceskysl.lsters.com/>)

注册并开通Android Market功能后, 你就可以进入自己的Market管理面板中了, 在这个面板中你可以上传自己的应用, 查看应用列表以及每个应用被下载、激活的统计数据等, 本篇文章针对此做相关的说明。

### 7.1 查看应用列表和统计信息

上传并发布应用后, 就可以被Andoird用户查看、下载和安装了, 这个时候你就可以在自己的Market管理界面中看到相关的统计信息了, 如下图。

The screenshot shows the Android Market Developer Console interface. At the top, there's a navigation bar with the Android Market logo and user information. Below this, a section titled 'Your Android Market listings' displays a table of applications. The first application, 'eoeAppInstaller(安装器) v1.5.1', is highlighted with a red background and numbered 1 through 7. The table columns include application details, ratings, total and active installs, price, and status. At the bottom right, there is a red button labeled 'Upload Application' with a plus icon and numbered 7.

Application	Rating	Total Installs	Active Installs	Percentage	Price	Status
eoeAppInstaller(安装器) v1.5.1 Applications: Tools	(75) ★★★★★	5207 total 2431 active installs	46%	Free	Published	
eoe手机信息助手(InfosAssistant) v1.1.0 Applications: Tools	(15) ★★★★★	645 total 180 active installs	27%	Free	Published	
ImApps vVer 1.4.3 Applications: Tools	(54) ★★★★★	2413 total 674 active installs	27%	Free	Published	
iTracks v1.0.0 Applications: Travel	(28) ★★★★★	1668 total 297 active installs	17%	Free	Published	

上图列出了四个应用（每行一个）以及每个应用的统计信息, 以第一个应用为例, 我们可以看看其提供了哪些统计信息, 按照我图上标识1..7的区域, 对应的说明如下。

区域1: 标识这是一个应用, 我这里的就是eoeAppInstaller（安装器）这个应用;

区域2: 显示的是这个应用的图标, 名字, 版本已经其所属分类;

区域3: 显示的是这个应用被用户评价的次数以及评分的平均值（以星的形式表示）

区域4: 显示的是这个应用被下载的次数（这里是5027次）以及安装激活的次数（这里是2431）以及激活量和下载量的比例（这里显示的是46%）;

区域5: 这个应用的价格。

区域6: 这个应用目前的状态（发布、暂停等）

区域7: 就是上传新应用的入口按钮。

## 7.2 上传应用的基本步骤

在管理界面内单击“Upload Application”按钮进入上传应用界面，如下图：

iceskysl@1sters! [Edit profile »](#)  
iceskysl

### Upload an Application

**Upload assets**  
Application .apk file  没有选择文件

**Listing details**  
Language | English (en\_US) | [add language](#)  
Title (en\_US)   
Description (en\_US) 

这里是应用的介绍

8 characters (325 max)

  
Application Type   
Category   
Price ☒ Free ☐ USD

### Publishing options

**Copy Protection**

- ☒ Off (Application can be copied from the device)
- ☐ On (Helps prevent copying of this application from the device. Increases the amount of memory on the phone required to install the application. Once copy protection is enabled, it cannot be disabled.)

**Locations**

- ☒ All Current and Future Locations
- ☐ Australia
- ☐ Austria
- ☐ Belgium
- ☐ Canada
- ☐ Czech Republic
- ☐ Denmark
- ☐ France
- ☐ Germany
- ☐ Greece
- ☐ Hong Kong
- ☐ Ireland
- ☐ Italy
- ☐ Japan
- ☐ Netherlands
- ☐ Norway
- ☐ Poland
- ☐ Portugal
- ☐ Singapore
- ☐ Spain
- ☐ Sweden
- ☐ Switzerland
- ☐ Taiwan
- ☐ United Kingdom
- ☐ United States

### Contact information

**Website**

**Email**

**Phone**

☒ This application meets [Android Content Guidelines](#)

☒ I acknowledge that my software application may be subject to United States export laws, regardless of my location or nationality. I agree that I have complied with all such laws, including any requirements for software with encryption functions. I hereby certify that my application is authorized for export from the United States under these laws. [\[Learn More\]](#)

Publish

Delete

Save

上图中，用数字标识的区域分别表示：

区域1：选择你已经正确签名的apk包，并上传；

区域2：添些不同语言的介绍信息，如名字，描述，分类，价格等；

区域3：选择发布属性选项，如是否需要防copy以及开放的国家、地区等；

区域4：填写相关的联系信息，如开发者的个人网站，Email以及电话等；

区域6：查看并选择一些条款等

最后是“发布”、“删除”、“保存”

### 7.3 升级应用的过程

如果你需要升级一个已经发布的应用，其操作也非常简单，在已经发布的应用列表中单击需要更新的应用，将展开其对应的编辑页面，如下：

IceskYsl@1sters! [Edit profile »](#)  
iceskysl@gmail.com

#### Upload an Application

**Upload assets**

Application .apk file com.eoemobile.app\_install(33k) [Upload Upgrade]

eoeAppInstaller

Version: 1.5.1

Localized to: , Chinese, English, French

This apk requests 0 permissions that users will be warned about

**Listing details**

Language | English (en\_US) |

[add language](#)

Title (en\_US) eoeAppInstaller(安装器)

Description (en\_US)  
-add chinese UI message..  
-updated to version 1.5 SDK.  
eoe App安装器 (Installer)可以收集你SD上保存的全部.apk文件,并可以直接安装该app或者删除这个Apk文件。  
-我们正在添加从互联网上直接下载、安装Apk文件的功能，敬请关注。  
(List all your .apk files on SD card,then you can install it to your phone)  
eoe mobile installer install china sd apk iceskysl 1sters  
288 characters (325 max)

Application Type Applications

Category Tools

Price Free

在上图中，其列出已经发布版本的信息，如果你需要升级，则可以单击“Upload Upgrade”，选择新版的APK包上传即可，同样，你可以修改其对应的介绍信息、分类等等。

### 7.4 其他问题

在我们使用过程中，发现一些其他的问题，分享如下：

- 1) 统计数据并不是实时的，一般上是一天定时更新两次；
- 2) 升级应用的时候需要注意特别注意签名的时候必须和以前版本一致，否则用户下载升级的时候，会报“签名不正确之类的提示”；
- 3) 升级应用的时候，升级前后应用的包名不能修改，必须保持一致，否则无法上传；
- 4) 发布或者升级的时候，所上传的APK包必须是经过正确签名的。



## 8. Android Market开发者发布协议

(Android Market Developer Distribution Agreement)

翻译：zhoubo526

原文地址：<http://www.android.com/us/developer-distribution-agreement.html>

(由于英语水平有限，本文仅供了解和参考，无法律效力)

约定

**Google**：谷歌，在特拉华州注册,主要营业地点在1600年剧场百汇，在Mountain View，CA 94043，美国。

**Device**：任何可以进入市场的移动设备。

**Products**：按照android SDK在设备上开发的软件，内容以及数字化资料，并且通过Market来分配销售

**Market**：android Market网站由谷歌维护和运营，开发者可以在这个平台上把产品分配到用户支持的设备上

**Developer or You**：任何在Market上注册并被批准的公司和个人，他们可以按照本协议发布产品

**Developer Account**：开发者用此账号可以在Market上发布产品

**Payment Processor(s)**：是谷歌给被授权的一方提供的支付处理服务，开发者可以用这个支付账号去收取用户购买产品所付的款

**Payment Account**：支付处理服务给开发者提供的金融帐户，它负责收集开发者在Market上销售产品所得的付款。开发者的申请必须被批准才能得到账号，而开发者必须保持账号有良好的信誉

移动网络运营商谁有权获得分配费的产品，出售给用户的设备在其网络上。

授权运营商：移动网络运营商有权获得产品的服务费，如果使用此产品的用户设备在其网络上。

### 8.1 导言

**1.1 Market**是一个公开的网站，开发者可以发布产品在Market上。为了能让产品成功发布到Market上，就必须获得并保持一个有效的开发人员帐户。

**1.2** 如果您要收取销售产品所得的费用，您还必须获得并保持一个有效付款帐户，并被支付处理服务授权

### 8.2 接受本协定

**2.1 android Market 开发分销协议**（“协议”）构成了一个具有法律约束力的合同，你和谷歌之间关于您使用的Market分销产品。为了利用Market分销产品，您必须首先同意本协议的情况下点击接受此选项提供给您。您不得分发产品投放市场，如果您不接受本协议。

**2.2** You may not use the Market to distribute Products and may not accept the Agreement unless you are verified as a Developer in good standing. This Agreement will automatically terminate if you are (a) not a Developer in good standing, or (b) a person or entity barred from using Android software under the laws of the United States or other countries including the country in which you are resident or from which you use the Android software.

**2.3** 如果您同意接受本协定在您的雇主或其他实体上生效，您代表并保证您的雇主或实体对本协定拥有全部法定权力。如果您不具备必要的权力，您可能无法代表您的雇主或其他实体接受此协议或利用Market。

## 8.3 价格和付款。

**3.1** 本协议涵盖了免费和收费的产品，为了收取你产品的费用，你必须有一个支付帐户，而且它在支付处理服务的单独协议中是生效的。如果在注册**Market**之前，您已经有了支付帐户，本协议的条款将取代您的支付帐户条款及条件，使用此协议在**Market**上销售产品。

您可以在支付服务容许的范围内为您的产品设置价格，**Market**显示给用户产品的价格，这个价格是用户的本地货币价格，但它不保证汇率和货币转换的准确性

**3.2** 您设定的价格决定你将收到的付款，交易费用，如以下定义，将收取的结算费用是的销售价格扣除分摊给平台服务费用后的部分，如果交易成功的话，经核实后。其余的（销售价减去交易费用）将汇给你。**"Transaction Fee"** 在<http://market.android.com/support/bin/answer.py?answer=112622>里所列

谷歌可不定期的进行修订，支付服务系统去收集产品销往地的应税和适用的税率，并由开发商负责确认，开发商负责联系当地税收管辖局去进行税务的减免。

**3.3** 您也可以选择免费分发产品。如果产品是免费的，您不会被收取的手续费。您以后可能不会从用户的副本产品中获得费用，这些用户最初被允许免费下载，这里并不是为了防止免费的版本去升级到收费的版本：产品经常以试用的方式被促销。然而，在免费试用到期之后，如果您想要收取费，您必须通过在**Market**上的支付系统来收取完整版本所得的费用。在这个协议里，**"free"**方式是指一种不收取费用的免费使用方式，所有在**Market**上发布产品所得费用，都必须通过**Market**的支付系统来收取

**3.4** 特别退款要求。关于标准的退款条件是附带在你发行到**Market**的产品上，除了以下的几种情况。可以让买家预览的**Products**（例如**ringtones**和墙纸）：没有需要的退款是不被允许。

不可以让买家预览的**Products**（例如应用）：如果买家在48个小时内请求退款在购买以后，您批准谷歌给买家产品价格的全部退款

**3.5** 去支持你的产品，你要单独负责维护产品和解决用户使用的抱怨，您的联络信息将被显示在每个应用程序的详细信息页，此目的是为了对产品提供技术支持

如果疏忽为你的产品提供支持，或许会导致产品低于规定的价值，降低关注度和销售额，甚至引发纠纷，支付服务系统收到结算纠纷后，假如产品的销售额小于**\$10**，可能会被退回给开发商，如果不属于上述情况，支付系统除了扣除服务费用，还会按照支付服务的政策标准扣除至少**\$10**

**3.6** 再安装。用户可以无限的重新安装发布在**Market**的产品

## 8.4 对于你 **Market**的用处

**4.1** 除了在下面第5部分 您授予的执照权利，谷歌同意没有获得权利去获得任何存在于这些应用中的知识产权

**4.2** 您同意为在(a)这个协议和(b)相关的法律、章程和普遍接受的惯例下使用**Market**(来自美国或其他相关国家关于数据或软件出口的任何法律)

**4.3** 您同意，如果您使用**Market**来发布产品，您将保护用户隐私和法定权利。如果用户提供您访问产品的用户名、密码，或者其他注册信息或个人信息，您必须使用户知道这些信息对您的产品将是有用的，并且您必须为那些用户提供法律上充分保密性通知和防护，进一步讲，您的产品也许会用户提供的某种功能可能必须要那么做，如果个人您的产品的商店或高度机密信息由用户提供了，那么它必须安全的和仅做它需要的，但是，如果用户选择了单独的协议以允许您的产品存储或使用个人或高度机密信息，去做与您的产品有关的事情(包括但不限于其他产品或应用)，那么单独的协议将指导您使用这些信息

**4.4 禁止行为。**您同意不会从事这些与**Market**有关的活动, 它包括发展或经销的产品, 干扰, 破坏, 损坏, 或在未经授权的方式访问设备, 服务器, 网络或其他财产或服务的任何第三方, 但是**android**的用户, 谷歌, 和任何移动网络运营商 并不在此限制范围, 您不得使用来自**Market**的客户信息, 在**Market**以外的平台发布产品

**4.5 非竞争。**您不得在**Market**提供任何产品, 其主要目的是促进在**Market**以外产品的销售

**4.6 您同意自行负责**(和谷歌没有责任, 对您或任何第三方的)在**Market**上销售的任何产品造成的后果(包括任何损失及谷歌也可能遭受的损失)。

**4.7 您同意自行负责**(和谷歌没有责任, 对您或任何第三方)违反您的本协定所规定的义务, 任何适用的合同或第三方的服务条款, 或任何适用法律或法规, 和所造成的后果(包括任何损失或损害谷歌或任何第三方可能遭受)任何此类违反

**4.8 市场**将可让您保护您的产品, 使用户可能无法共享产品与其他用户或设备

**4.9 产品评级。****Market**将允许用户对产品估计。只有在下载此产品的用户才能给他评分, 产品评级将被用来确定产品在**Market**上的排名位置, 较高的评价产品通常给予较高的排名, 但谷歌有权利根据自己的判断来改变排名, **Market**会分配给没有用户评分的产品一个综合评分, 开发者的综合得分将是一个代表您过去的产品质量的记录, 它是谷歌唯一的确定方法。如果没有新产品的开发记录, 谷歌将使用或发布业绩计量, 如卸载和/或退款率, 以肯定或删除没有达到标准的产品。

您的产品可能会受到用户评分来, 你们也许会不同意。您可以与谷歌联系, 如果您有任何问题或疑虑, 例如评级<http://market.android.com/support>。

**4.10 营销您的产品。**你将负责上传您的产品到**Market**, 提供所需的产品信息给用户, 并准确说明的产品功能和必需的安全权限以及用户设备, 产品上传不正确将不会被在**Market**上发布。

**4.11 受限制的内容。**任何产品中发布到市场必须遵守为开发商定制内容政策, 它的位于<http://www.android.com/market/terms/developer-content-policy.html>。

## 8.5 授权资助

**5.1 您给予谷歌一非独占, 全球和免费许可:**拷贝, 执行, 显示, 并使用产品的行政和示范目的来联系运作和销售的市场, 使用这些产品进行改进在**android**的平台上(这句话不知道怎么翻译才能通顺 就直译了)

**5.2 您授予谷歌非独家的和免版税许可的,**能够根据发布选项选中的选项来分发产品, 在你上传产品的**market**网页上

**5.3 谷歌可能会使用顾问**在与其他承包商之间履行义务和行使本协议下的权利, 承包商将同谷歌一样履行与自己相关的义务和行使本协议下的权利, 在终止本协议之后, 谷歌将不会发布您的产品, 但可保留与使用复制的产品来支持**Market**和**android**平台

**5.4 您授予给用户的非独家的, 全球和永久使用权,**来执行, 显示, 并使用该产品在设备上。如果您选择, 您可能会包括一个单独的最终用户许可协议( **EULA** ) 在产品中, 它将主导用户的权利去使用产品, 以代替以前的协议。

## 8.6 品牌特征和宣传

6.1 “品牌特征”是指商品名称, 商标, 服务标记, 徽标, 域名, 以及其他方面有鲜明的品牌特征, 如在这样的部分拥有(或授权)

6.2 双方都应该拥有所有权利, 包刮但不限于所有的知识产权, 与其品牌特征相关的部分, 除有限的范围本协议明确规定, 任何一方的赠款, 也不得另一方获取, 任何权利, 所有权或利益(包括但不限于任何暗示的授权), 或任何品牌特征的另一方, 符合规定和条件的本协议, 开发者授予以谷歌和其子公司有限的非独占的许可, 在协议期限内。开发在**Market**上履行其规定的义务, 并体现出他的品牌特征, 在本协议中的任何给开发商无权使用任何谷歌的商标名称, 商标, 服务标记, 徽标, 域名或其他鲜明的品牌特征

6.3 宣传。谷歌和其附属公司可能宣传包括开发者品牌的演示文稿, 营销材料, 新闻稿和客户名单(但不包括客户名单张贴在谷歌网站)为目的的营销**android Market**。谷歌授予开发者有限的, 非排他性的, 全球性的免费许可来使用**android**的品牌特征在本协议期限内, 仅用于市场营销的目的, 并且**android**按照品牌指南来做<http://www.android.com/branding.html>).

## 8.7 产品 Takedowns

7.1 您**Takedowns**。您可以删除任何通过**Market**发布的产品, 但是你必须遵守本协议和付款处理的条款, 包括无限制退款, 在以下情况你不能删除产品

- (a)影响到曾购买此产品用户的许可使用权
- (b)会连带删除用户曾经购买或下载的程序
- (c)影响到您对产品提供技术支持的服务, 这些产品此前用户已购买或下载。

7.2 谷歌**Takedowns**, 谷歌没有打算, 也不会去监视你产品的内容, 但是如果谷歌对你的产品进行下架可能是因为以下一些原因

- (a)侵犯了知识产权或其他第三方权利
  - (b)违反任何适用法律或受禁制令;
  - (c)是色情, 猥亵或其他违反谷歌的托管政策或其他方面的服务协议
  - (d)不正当的发布
  - (e)可能让谷歌造成赔偿损失或侵权行为
  - (f)被谷歌认为是一种病毒, 恶意软件, 间谍软件, 或者对谷歌和其他运营商产生不利影响, (g)开发者违反本协议的条款或市场政策的内容
  - (h)展示的产品对谷歌整体的服务产生影响(如 客户无法访问这些内容)
- 谷歌有权对这些产品进行下架和禁止开发者在**Market**上的不适当行为。

谷歌与设备制造商和运营商签订了分销协议, 将**Market**的客户端安装到设备上, 如果你的产品被删除了, 可能是因为它有某些漏洞或恶意行为, 谷歌在这方有以下权力:

- (a)不容许你再销售带有条款所列缺陷和恶意行为的产品
- (b)将收回你销售带有缺陷的产品所得款项
- (c)向你收取额外的费用, 这些费用是指去从**Market**上, 硬件设备上和网络上删除有漏洞产品所产生的费用

## 8.8 您的开发证书

8.1 您同意负责谷歌给你颁发的开发者证书的保密性, 你将全权负责所有的应用程序下制定的开发证书, 谷歌可能会有有限的派发一些开发者的账号给你, 或你的公司

## 8.9 隐私和信息

9.1 为了不断创新和完善市场, 谷歌可能会从市场和设备收集一些使用统计资料, 包括如何使用**Market**和设备的信息, 但不这些。

9.2 对所收集的数据进行审查和统计, 以此来改进**Market**的服务, 并坚持根据谷歌的隐私政策。为了确保改进产品, 对数据的汇总, 可向你提出书面的申请

## 8.10 终止本协定

10.1 这个协议在你和谷歌终止之前, 将一直生效

10.2 如果想终止本协议, 你必须提前**30**以书面形式通知谷歌(除非本协议因为第**14.1**条终止), 并停止用与之有关的开发证书

10.3 如果出现以下原因, 谷歌可以随时终止协议

- (a) 您违反本协议的任何规定
- (b) 因为法律原因谷歌必须这么做
- (c) 你的授权已过期
- (d) 谷歌决定不在提供**Market**平台。

## 8.11 免责声明

11.1 你明确并同意承担使用**Market**所带来的风险, 而且**Market**仅提供现有的平台, 不做任何保证。

11.2 你在**Market**上下载的工具和软件所产生的风险都有你自己去判断和负责, 这些风险包刮损坏您的计算机系统或其他设备或丢失的数据

11.3 **GOOGLE** 进一步明确声明, 不提供任何明示或暗示的担保和条件, 包括但不限于为适销性、特定目的的适用性以及不侵权的暗示保证和条件。

## 8.12 责任范围

12.1 您明确理解并同意 **GOOGLE**、其分支机构和附属机构以及其许可方不应因以下原因而对您负责：

(A) 您导致的任何直接、间接、偶然、继发性或惩罚性损失, 不管这是由什么原因导致的也不管是否属于责任范围内。这应当包括但不限于任何利润损失（无论是直接还是间接导致的）、任何商誉的损害、遭受的任何数据丢失、购买替代商品或服务的费用或其他无形损失；

## 8.13 赔偿

13.1 在法律容许的最大范围内, 如果出现以下情况, 你同意去维护和保障谷歌免受起诉和索赔。

(a) 你在违反本协议的情况下使用**Market** (b) 你的产品侵犯版权, 商标, 专利及知识产权或或透露保密性的资料

## 8.14 对协议的更改

14.1 谷歌可能随时变更本协议, 并通过电子邮件告诉开发者所做的修改, 谷歌还将其修改的部分发布到Market网站上, 这些修改将会被视为有效的和开发者所接受的, (a)对将刚注册成的开发商的人立被刻通知(b)如果已经成为开发商的人, 修订后的协议, 被开发商接纳后才生效(除非法律要求对所做的改变立即生效), 开发商必须表明进入Market的产品 都接受了修改后的协议, 如果开发商不同意修改后的协议, 三十后谷歌将暂停开发商在Market上的产品销售, 直到开发商同意为止, 如果90天后开发商仍没有同意修改后的协定, 你只能终止与谷歌合作。

## 8.15 常规合法条款

15.1 本协议构成了整个法律协议您与谷歌和规范的使用Market, 并完全取代了之前你和谷歌签订有关Market的所有协议。

15.2 您同意如果 Google 不使用或执行“条款”中包含的任何合法权利或补救措施(或 Google 根据任何适用法律享有的权益), 则不能视为正式放弃 Google 的权利并且这些权利或补救措施仍将适用于 Google。

15.3 如果任何(对此事宜有司法管辖权的)法庭规定这些“条款”的任何规定无效, 那么无效规定将从“条款”中删除, 而不会影响剩余的“条款”。“条款”中的剩余规定将继续有效并可执行。

15.4 您承认并同意作为 Google 子公司的公司集团的每个成员都应是“条款”的第三方受益人, 并且此类其他公司应有权直接执行并遵守对其有利(或其可从中获利)的“条款”的任何规定。除此之外, 其他任何个人或公司都不应是“条款”的第三方受益人

15.5 出口限制。产品发布到Market可能会受到美国出口法律和法规。您必须遵守所有国内和国际适用于您的发布和使用产品的法律和法规

15.6 您或谷歌的事先书面批准的另一方在本协议中对授予的权利不得转让或转移。在本协议项下无论是你还是谷歌事先书面批准的另一方都不得委托其责任或义务。

15.7 “条款”以及条款规定的您与 Google 之间的关系均应受加利福尼亚州法律的管制, 无论“条款”是否与法律规定存在冲突。您和 Google 均同意遵守位于加利福尼亚圣塔克拉拉的法院的专属管辖, 以解决“条款”引起的任何法律问题。尽管如此, 您同意仍允许 Google 就任何管辖权申请补救措施(或同等类型的紧急法律救治)

15.8 本协议职责中的义务第5, 6, 7, 11, 12, 13和15将到期或终止本Agreement或本协议规定的义务, 必须事先书面批准的其他当事方

## 9. 其他

### BUG提交

如果你发现文档中翻译不妥的地方，请到如下地址反馈，我们会定期更新、发布更新后的版本  
<http://www.eoeandroid.com/viewthread.php?tid=625&extra=>

### 参加翻译

如果你有兴趣参加我们的协同翻译，请参考如下链接  
<http://www.eoeandroid.com/forumdisplay.php?fid=39>

### 关于eoeAndroid

当前，3G商业，传统互联网与移动互联网也呈现出全业务发展的融合趋势，电信与互联网行业已经踏入继单机计算时代、传统互联网时代之后的第三个纪元。

由于看好移动互联网和Android手机平台的商业前景，同时也拥有专业而独特的产品、技术服务能力，我们聚集了一群热爱Android的技术英才，组建了eoeMobile团队。

eoeMobile是一支专注于Android平台应用开发、产品运营和相关商业与技术服务的团队，立志于建立中国最大的Android应用开发专业社区[eoeAndroid.com](http://www.eoeandroid.com)，想为Android在中国的发展尽自己的微薄之力。