

Android 1.5 SDK, Release 1 介绍

(中文版)

发布版本: Ver 1.0.0(build 2009.05.06)

© Copyright 2009 eoeAndroid.com. All Rights Reserved.

序

Android飞速发展, 1.1版本的SDK发布没多少时间, 1.5版的SDK又一次正式发布了, [eoeAndroid](http://www.eoeandroid.com)社区组织并邀请一群Android爱好者一起, 将其翻译成中文版本。

望此文档能给更多人带来有用的信息, 如果您发现有不妥的地方, 欢迎指出, 同时也希望有时间、有激情、愿意贡献的朋友一起加入翻译团队, 后续会不断组织、翻译其他的文章、文档和书籍!

资源

活动发起地址:

<http://www.eoeandroid.com/viewthread.php?tid=129>

原文地址:

http://developer.android.com/sdk/1.5_r1/index.html

特别感谢eoeAndroid社区中的下列几位, 他们为翻译投入了大量的时间和心血, 大家才可以在最短的时间里看到最新的知识。

翻译人员:

- lceskYsl(<http://iceskysl.1sters.com>)
- haiyangjy(<http://www.haiyangjy.com>)
- 404(<http://l404.blogspot.com>)
- ddna

eoeAndroid社区原创中国大陆第一本**Android**开发教程

详情可见社区 <http://www.eoeandroid.com/viewthread.php?tid=152>

国内第一本大陆原创图书 《Google Android开发入门与实战》

人民邮电出版社 国内eoeMobile开发团队全力打造的第一本著作

内容特点(包括但不限于):

- * 国内第一本原创Android图书
- * 完全基于Android最新的SDK1.5
- * 全书除了大量小型案例之外还包含了5个Android平台下的完整商业实例及源码分析, 分别是RSS阅读器、基于GoogleMap的个人GPS、豆瓣客户端、在线音乐播放器、手机信息助手
- * 随书附赠的光盘中包含300分钟的详细教学视频以及Android开发必备的开发资源
- * 读者对于此书内容的疑问可以访问<http://www.eoeandroid.com>社区, 作者团队将会及时解答

目录

Android 1.5 SDK, Release 1 介绍	1
1. 下载Android 1.5 SDK, Release 1	4
1.1 SDK内容.....	4
1.1.1 开发者工具.....	4
1.1.2 Android 平台	4
1.1.3 SDK 插件	5
1.1.4 示例代码和应用	5
1.1.5 文档	5
2. 安装 Android SDK.....	6
2.1 准备安装.....	6
2.2 安装 SDK.....	6
2.3 在Eclipse上安装ADT插件.....	7
2.4 ADT安装之疑难解答.....	8
2.5 其他安装错误.....	8
2.5.1 Linux 用户.....	9
2.6 接下来	9
2.7 安装说明.....	9
2.7.1 Ubuntu Linux 用户请注意	9
2.7.2 使用其他 Linux 发行版的用户请注意	10
3. 升级SDK	11
3.1 安装SDK.....	11
3.2 更新Eclipse上的ADT插件.....	12
3.3 卸载旧版ADT插件	12
3.4 安装0.9 版本的ADT插件.....	12
3.5 更新Eclipse上的SDK设置.....	13
3.6 更新您的项目.....	13
3.7 Eclipse用户	13
3.8 Ant用户.....	14
3.8.1 build.xml已改变.....	14
3.9 迁移应用程序.....	14
3.10 让应用程序保持最新	15
4. 系统要求	17
4.1 支持的操作系统.....	17
4.2 支持的开发环境.....	17
5. Android 1.5 SDK 发布记录	17
5.1 发布概览.....	18
5.2 安装和升级记录.....	18
5.3 SDK 扩展.....	18
5.4 Android虚拟设备(AVDs).....	19
5.5 其他的一些记录:	19
5.5.1 地图外部库:	19
5.5.2 Window中的USB驱动.....	19
5.5.3 解决的问题和改变.....	19
6. Android 1.5 版本记录.....	21
6.1 概览	21
6.2 内置的程序	22
6.3 本地化	22
6.4 新特性	23
5. API 的变动.....	23
5.1 概览	23
6. API 改变细节.....	24

6.1新增加的包.....	24
6.2改变的包.....	24
7. Android1.5平台的亮点	26
7.1 用户UI界面的改善	26
7.2 性能的改善	26
7.3 新特性	26
7.4 新的API和Manifest元素.....	0
8. Android 1.1 版本说明	0
8.1 概述	28
8.2 外部库	29
8.3 设备兼容性	29
8.4 内置的应用	29
8.5 UI 本地化.....	29
8.6 修正的问题	30
8.7 新特性	30
8.8 API 改动	30
8.8.1API更改细目:	30
9. 其他.....	32

1. 下载Android 1.5 SDK, Release 1

2009年4月

翻译者: lceskYsl

原文地址:

http://developer.android.com/sdk/1.5_r1/index.html

在下载前, 请仔细查看“系统要求”文档, 当你确定需要下载的时候, 还需要你查看并同意“Terms和Conditions”。

平台	下载包	大小	MD5值
Windows	android-sdk-windows-1.5_r1.zip	176263368 字节	42be980eb2d3efaced01ea6c32c0045f
Mac OS X (intel)	android-sdk-mac_x86-1.5_r1.zip	167848675 字节	5b2a8d9f096032db4a75bfa0d689a51b
Linux (i386)	android-sdk-linux_x86-1.5_r1.zip	162938845 字节	2addfd315da0ad8b5bde6b09d5ff3b06

欲了解更多信息, 请查阅“发布说明”。

1.1 SDK内容

1.1.1 开发者工具

SDK包含了开发、调试以及UI设计的一系列工具, 你可以在“开发向导”中查看各个工具的用途, 也可以打开你的 `<sdk>/tools/` 目录中它们。

在这个发布版的SDK中, 和相对以前版本的SDK有了很多更新, 其中包括一些工具需要的项目目录结构有所不同。如果使用新工具, 你需要将以往的项目迁移到新的开发环境中。你可以阅读“升级SDK”文档获取更多的信息。

如果你想了解新工具的特性, 请查阅“SDK发布说明”

1.1.2 Android 平台

本SDK包含了多个Android版的开发平台, 针对每个版本都提供了一套Android类库和系统镜像。下面表格列举了这个版本的SDK中包含的平台版本, 如果你需要了解每个版本平台的详细信息(比如特性, 包含的应用, 国际化, API更新等), 请参考其版本发布说明。

平台	API等级	发布说明	描述
Android 1.5	3	发布说明	包含一个标准的1.5版本的类库和系统镜像, 不包含扩展类(例如Maps类)

Android 1.1	2	发布说明	包含一个可用的1.1版本的类库和系统镜像，同时包含一些扩展类库。
-------------	---	------	----------------------------------

1.1.3 SDK 扩展

SDK扩展为开发环境提供了Android扩展类或者是定制的系统镜像。这个SDK中包含的插件列举如下，其需要的api等级也有所提及。

插件	API等级	描述
Google Apis	3	包含 <code>com.google.android.maps</code> 扩展类、系统镜像、 Geocoder 服务、文档和一些例子。

1.1.4 示例代码和应用

你可以在“开发向导”中查看示例代码和其说明，并可以在SDK包的<sdk>/platforms/android-1.5/samples/ 目录下找到这些代码。需要注意的是新版的SDK支持多版本平台模式，你可以针对每个平台编译一套代码。

1.1.5 文档

本SDK包包含了全部的离线文档，你可以在浏览器中打开 <sdk>/documentation.html文件进行查看。如果你是在诸如Eclipse的IDE中开发的话，你将可以在IDE中直接查看文档。

你还可以随时在Android开发者社区中产最新的文档，其地址如下：

<http://developer.android.com/>

2. 安装 Android SDK

翻译者：404

原文地址：

http://developer.android.com/sdk/1.5_r1/installing.html

这篇文档向开发人员描述了如何安装 **Android SDK** 及设置开发环境。如果您尚未下载**Android**开发工具包（**Android SDK**），请转到[下载](#)页面。当您完成下载后，请继续往下阅读。

要是您在安装过程中遇到什么问题，可以查阅页尾部分的[安装说明](#)。

升级旧版SDK

若您已经使用早期的**SDK**版本开发过应用程序，请参考 [升级SDK](#) 这章。

2.1 准备安装

在开始安装之前，请花些时间检查您的开发机器以确认符合[系统要求](#)。

要在**Eclipse** 上使用 **Android**开发工具插件（**ADT**）做开发（推荐这样做若您是一位**Android**新手），请确保您在计算机已经安装了适当版本的**Eclipse**（3.3版或以上）。如果需要安装**Eclipse**，可以访问如下网址来下载：

<http://www.eclipse.org/downloads/>

推荐下载 **Eclipse** 的 Java 或 RCP 版。

2.2 安装 SDK

当**SDK**下载完毕后，请将此.zip归档文件解压到合适的目录中。如果未指定存放目录，**SDK**文件将会默认地被解压到名为android_sdk_<platform>_<release>目录中。在**SDK**解压后的存放目录里，会包含文档的本地拷贝（您可以直接在浏览器中浏览documentation.html）和一些子目录诸如 tools/, add-ons/, platforms/ 等等。在 platforms/ 的每一个子目录中，您都可以找到 samples/目录，里面包含一些针对其版本的样例程序。

可选地，您也许想把**SDK**主要的 tools 目录添加到系统路径（system PATH）中。tools/ 位于 **SDK**的根目录下，将 tools/ 注入系统路径后，可以在不用切换到完整tools路径的情况下运行 **Android Debug Bridge (adb)** 和一些其他命令行工具。

- 在 Linux 上，编辑 ~/.bash_profile 或 ~/.bashrc 文件，找到设置 PATH 环境变量的那行并将完整的tools/路径写入其中。如果没有找到，也可以添加如下一行代码来实现：

```
export PATH=${PATH}:<your_sdk_dir>/tools
```

- 在 Mac 上, 可以在用户主目录中编辑 (或创建) .bash_profile 文件进行如上Linux环境中一样的操作。
- 在 Windows 上, 右击“我的电脑”, 然后选择“熟悉”, 在“高级”选项卡里边, 单击“环境变量”按钮, 此时会弹出一个对话框, 在里面找到系统变量Path并双击, 然后将完整的tools/路径写入其中点击确认即可。

注意, 如果您以后升级SDK而tools/路径有所变更的时候, 记得同时更新您在环境变量中所做的设置, 到时可能需要指向新的路径。

如果您将使用Eclipse IDE作为您的开发环境, 下一节会描述如何安装Android开发工具插件 (ADT) 及设置Eclipse。若您不使用Eclipse, 同样可以选择您喜爱的IDE来开发Android应用, 以及使用SDK中的工具来编译, 调试和部署应用程序。(跳到 [接下来](#) 这节)

2.3在Eclipse上安装ADT插件

Android 为Eclipse 定制了一个插件, 即Android Development Tools (ADT), 目的是为您提供一个强大的综合环境用于开发Android应用程序。ADT扩展了Eclipse的功能可以让您快速地建立Android项目, 创建应用程序界面, 在基于Android框架API的基础上添加组件, 以及用SDK工具集调试您的应用程序, 甚至导出签名 (或未签名) 的APKs以便发行您的应用程序。

通常, 使用Eclipse搭配ADT做Android应用开发是一种高度推荐的做法, 同时也是起步的最快途径。(如果您更喜欢用Eclipse之外的IDE工作, 那就无需安装Eclipse或者ADT, 可以直接使用SDK工具集构建和调试您的应用程序。)

当您参照[准备安装](#)一节的内容安装了Eclipse, 下载ADT后可以参照以下步骤在您自己的Eclipse中进行安装。

Eclipse 3.3 (Europa)

1. 启动 Eclipse, 在菜单栏里选择 **Help > Software Updates > Find and Install...**
2. 在弹出的对话框中, 选择 **Search for new features to install** 并点击 **Next**.
3. 单击 **New Remote Site**.
4. 在出现的对话框中, 为远程站点输入一个名称 (例如 "Android Plugin") 接着输入站点的 URL : `https://dl-ssl.google.com/android/eclipse/`
如果无法获取插件, 尝试在 URL 里使用 "http" 代替 "https" (使用https是出于安全考虑)。
点击 **OK**.
5. 此时, 您应该能够看到刚才添加的URL已经搜索列表了 (且已勾选), 点击 **Finish**.
6. 在随后用于呈现搜索结果的对话框里边, 选中"Android Plugin"。这将会选择其下的附属工具: "Android DDMS" 和 "Android Development Tools"。点击 **Next**.

Eclipse 3.4 (Ganymede)

1. 启动 Eclipse, 在菜单栏里选择 **Help > Software Updates...**
2. 在弹出的对话框中, 点击 **Available Software** 选项。
3. 点击 **Add Site...**
4. 输入网址: `https://dl-ssl.google.com/android/eclipse/`
如果无法获取插件, 尝试在 URL 里使用 "http" 代替 "https" (使用https是出于安全考虑)。
点击 **OK**.
5. 返回到可用软件窗口, 点击展开URL前面的节点, 您应该能看到"Developer Tools"项, 勾选后并点击 **Install...**
6. 在随后出现的安装窗口, "Android DDMS" 和 "Android Development Tools" 项必须选中, 接着点击 **Next**.
7. 阅读并接受协议条款并点击 **Finish**.
8. 重启 Eclipse.

7. 阅读并接受协议条款并点击 **Next**.
8. 点击位于安装窗口下面的 **Finish**.
9. ADT 插件没有数字签名，总之点击 **Install All** 进行安装即可。
10. 重启 Eclipse。

此时您需要修改Eclipse配置来指明Android SDK目录：

1. 在菜单栏中选择 **Window > Preferences...** 来打开设置面板（Mac: **Eclipse > Preferences**）。
2. 从左面板选中**Android**项。
3. 在主面板指明SDK位置的地方（*SDK Location*），点击 **Browse...** 然后将路径指向SDK 存放目录。
4. 点击 **Apply**, 接着 **OK**.

安装完毕！如果您未遇到任何问题，那么您可以开始开发Android应用程序了。可以参考 [接下来](#) 这节内容了解关于起步的一些建议。

2.4ADT安装之疑难解答

如果您在下载ADT插件中遇到问题，这里给出一些建议供参考：

- 假如Eclipse不能从远程站点获取ADT插件，尝试使用http而不是https开头的远程网址。也就是说，采用如下的远程网址：

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

- 如果您的网络隔着防火墙（例如企业防火墙），请确保您已经在Eclipse中正确地设置了代理。在 Eclipse 3.3/3.4 中，可以在Eclipse中选择 **Window (on Mac, Eclipse) > Preferences > General > Network Connections** 来设置代理。

如果您还是不能通过Eclipse从远程站点获取ADT插件，您可以单独下载 ADT zip 文件到本地目录并按如下步骤进行手工安装：

1. [下载ADT zip 文件](#)（无需解压）。
2. 参照以上Eclipse安装ADT中的前两个步骤（即步骤1和2）。
3. 在 Eclipse 3.3 中，点击 **New Archive Site...** 在 Eclipse 3.4 中，点击 **Add Site...**, 然后 **Archive...**
4. 浏览并选择已经下载到本地的zip归档文件。
5. 参照以上Eclipse安装ADT插件过程的第5步及其后面的步骤继续安装。

若要更新通过zip归档文件安装的ADT插件，您必须再一次参照以上流程而不是遵循默认的更新向导。

2.5其他安装错误

注意，ADT的一部分功能依赖一些可选的Eclipse组件（比如WST）。如果您在安装ADT时遇到错误，可能是因为您的Eclipse安装包中没有包含这些组件。要了解有关快速安装Eclipse组件的相关信息，详见这篇[解疑难问题的主题 ADT Installation Error: "requires plug-in org.eclipse.wst.sse.ui"](#)。

2.5.1 Linux 用户

如果您在Eclipse上安装ADT插件的过程中遇到如下错误：

An error occurred during provisioning.
Cannot connect to keystore.
JKS

...那是因为您的开发机器上缺少合适的Java VM环境；可以通过安装Sun Java 6解决此问题，之后您需要重新安装ADT插件。

2.6 接下来

当您完成安装后，您就可以开始开发应用程序了。这里给出一些方法供参考：

学习Android

- 可以看看 [开发指南](#) 了解相关信息
- 以及了解 [Android 是什么?](#)
- 要了解 Android 框架及其应用程序是如何在之上运行的，详见 [Application Fundamentals](#)
- 了解Android框架的API说明，详见 [Reference](#)。

SDK 指引

- 这里为您提供了一份 [开发工具](#) 概览
- 了解如何使用 [Eclipse/ADT](#) 或 [其他 IDEs](#)进行开发

一些代码

- 创建一个[Hello World 程序](#)（强烈推荐，尤其是Eclipse用户）
- 参考 [记事本教程](#) 创建一个完整的 Android 应用
- 从`<sdk>/platforms/<platform>/samples` 样例程序中生成一个项目，然后试着在开发环境中编译并运行

访问Android 开发组

- 去 [社区](#) 看看了解开发组信息。也许您特别想看看[Android 开发组](#)来了解下Android开发社区的情况。

2.7 安装说明

2.7.1Ubuntu Linux 用户请注意

- 如果您在开发机器上安装和设置Java需要帮助，您可能在查找如下这些有用的资源：

<https://help.ubuntu.com/community/Java>
<https://help.ubuntu.com/community/JavaInstallation>

- 在安装Android SDK 和 ADT Plugin之前，在此列出一些Java和Eclipse的安装步骤。

1. 如果您在开发机器上运行的是64位发行版，需要通过apt-get命令安装 ia32-libs 软件包：
apt-get install ia32-libs
2. 然后，安装Java:
apt-get install sun-java6-bin
3. Ubuntu 的包管理器暂不能下载 Eclipse 3.3，我们推荐您从官网eclipse.org (<http://www.eclipse.org/downloads/>) 下载。推荐使用 Eclipse 的 Java 或 RCP 版。
4. 参照之前在Eclipse上安装ADT的步骤安装 SDK 和 ADT 插件。

2.7.2使用其他 Linux 发行版的用户请注意

- 如果您在开发机器上已经安装了JDK，请检查下JDK的版本以确保符合[系统要求](#)。要特别注意的是，一些Linux发行版可能已经包含了JDK 1.4或Java GNU编译器，而这两项都不适合做Android的应用开发。

3. 升级SDK

翻译者：404

原文地址：

http://developer.android.com/sdk/1.5_r1/upgrading.html

升级概览

Android 1.5 SDK使用新的项目结构和新版ADT插件。

将现有的项目移植到新版SDK中，您必须针对您的开发环境做一些必要的改动。

新版ADT插件(0.9版)与在以往的SDKs中创建的项目不会冲突。

在安装ADT 0.9之前，您需要卸载旧版的ADT插件。

本章目录

[安装SDK](#)

[更新Eclipse上的ADT插件](#)

[更新您的项目](#)

[Eclipse用户](#)

[Ant用户](#)

[迁移应用程序](#)

[让应用程序保持最新](#)

迁移参考

[Android 1.5 API的差异](#)

[让您的应用程序保持最新»](#)

[Android 1.5中UI框架的改变»](#)

这篇文档介绍了如何将您的开发环境和现有的Android应用程序从Android 1.0或1.1版的SDK迁移至Android 1.5 SDK。如果您要从1.0之前的SDK中迁移应用程序，请同时查阅Android 1.0 SDK的升级文档。

升级到新版SDK有几个主要原因，比如新的SDK工具可以提高您的开发效率，而且新的APIs允许您扩充现有应用程序的功能。然而，即便您或您的应用程序无需这些增强功能，也很有必要升级以确保您的应用程序能够顺利地运行在Android 1.5平台上。

Android 1.5平台将会很快地投入到世界各地的一些设备中。如果您已经向公众发布过应用程序，您应该尽快地在最新版本的平台为您的应用程序进行前向兼容性（forward-compatibility）测试。尽管您不太可能在测试中遭遇断点，但是为了最佳的用户体验，您应该谨慎一点。所以，请安装新版Android SDK并且在Android 1.5上测试您的应用程序。

欲了解更多有关新版SDK的功能和系统变动，详见：[Android 1.5 Version Notes](#)。

3.1 安装SDK

如果您尚未下载SDK，可以 [从这里下载](#) 并解压到合适可用的目录中。

开始之前: 如果您以前设置过PATH环境变量来指向SDK的tools目录, 那么您需要更新来指向新的SDK。比如, 在一个.bashrc或.bash_profile文件中:

```
export PATH=$PATH:<your_sdk_dir>/tools
```

如果您不使用Eclipse做开发, 跳到 [更新您的项目](#)。

3.2更新Eclipse上的ADT插件

如果您已经安装过ADT-0.9_pre 和早期的1.5 SDK, 请注意新版本已经另有改动, 所以, 请您继续参考这篇指南并更新到最终的ADT 0.9 版。

Android 1.5 SDK 需要新版ADT插件(目前最新版本是0.9), 因为自从Android 1.1 起部分结构已经发生了变更, Android 1.5 SDK 不能工作在ADT 0.8(或更旧的)版本上且之前已安装的SDK也不能作用于ADT 0.9。然而, Android 1.5 SDK 包含一个 Android 1.1 SDK 镜像可以让您在此镜像上使用ADT 0.9 创建项目。

欲了解更多关于使用不同的系统镜像(比如Android 1.1)来运行此SDK, 根据您的开发环境参考[使用Eclipse和ADT进行开发](#)或在[其他IDEs中进行开发](#)这两篇文档。

为了升级您的Eclipse IDE 来使用 0.9 版本的ADT, 参照以下各版本的升级步骤。

3.3 卸载旧版ADT插件

您必须卸载旧有的ADT插件(0.8或更早版本), 如果不卸载掉, 在安装新版ADT时, 它将会与Android Editors发生冲突。(如果您已经安装过ADT-0.9_pre 和早期的1.5 SDK, 可以跳过此卸载流程前往阅读[安装 0.9 版的 ADT 插件](#)这一节的内容)。

Eclipse 3.3 (Europa)

1. 选择 **Help > Software Updates > Manage Configuration**.
2. 展开左侧面板的列表以显示已安装的工具。
3. 右键 "Android Editors" 并点击 **Uninstall**. 点击 **OK** 确认.
4. 重启 Eclipse.

(不要卸载 "Android Development Tools".)

Eclipse 3.4 (Ganymede)

1. 选择 **Help > Software Updates**.
2. 选中 **Installed Software** 项.
3. 选择 "Android Editors". 并点击 **Uninstall**.
4. 在接下来的窗口中, 确保 "Android Editors" 已被选中, 然后点击 **Finish** 进行卸载.
5. 重启 Eclipse.

(不要卸载 "Android Development Tools".)

3.4 安装0.9 版本的ADT插件

只有在 [卸载旧版ADT 插件](#) 后才可以安装新的ADT插件。

Eclipse 3.3 (Europa)

Eclipse 3.4 (Ganymede)

1. 选择 **Help > Software Updates > Find and Install**.
 2. 选择 **Search for new features to install**.
 3. 选中Android 插件条目, 然后点击 **Finish**.
 4. (插件的原始条目应该还在, 否则, 参照 [安装 ADT 插件 指南](#).)
 5. 然后, 展开 Android 插件条目并确保 "Developer Tools" 已勾选, 接着点击 **Next**. (这将包含 "Android DDMS" 和 "Android Development Tools".)
 6. 阅读并接受协议条款, 然后点击 **Next**.
 7. 在接下来的窗口中, 点击 **Finish** 开始安装.
 8. ADT 插件没有数字签名, 总之点击 **Install All** 接受安装即可。
 9. 重启 Eclipse.
1. 选择 **Help > Software Updates**.
 2. 选中 **Available Software** 项.
 3. 展开 Android 插件项 (可能是作为URL列出), 选中 "Developer Tools" 项, 然后点击 **Install**.
 4. 在接下来的窗口中, "Android DDMS" and "Android Development Tools" 应该都被选中. 点击 **Finish**.
 5. 重启 Eclipse.

如果遇到任何问题, 确保您的ADT已经完全卸载并参考 [在 Eclipse 上安装 ADT 插件](#)

3.5 更新Eclipse上的SDK设置

最后需要更新您的Eclipse设置来指向新的SDK目录:

1. 选择 **Window > Preferences** 打开设置面板 (Mac: **Eclipse > Preferences**).
2. 从左侧面板中选择 **Android**.
3. 在主面板指明SDK位置的地方 (SDK Location), 点击**Browse...** 然后将路径指向SDK 存放目录.
4. 点击 **Apply**, 然后 **OK**.

3.6 更新您的项目

此时, 您需要更新在以前版本的Android SDK 上开发过的所有Android项目。

3.7 Eclipse用户

如果您使用 Eclipse 开发应用程序, 参考以下流程逐个更新项目:

1. 选择单个项目(在 Package Explorer 中), 右键点击 **Properties**。
2. 在属性面板中, 打开Android 面板并选择一个"build target" 以备编译。SDK提供了Android 1.1 和 Android 1.5 这两个平台可用于选择。若您是初次更新项目到新版的SDK, 我们推荐您选择Android 1.1 版的类库平台, 接着点击 **Apply**, 然后按下 **OK**。

新插件在您的项目中创建了一个 **gen/** 文件夹, 里面包含 **R.java** 以及自动生成的 **AIDL java** 文件。如果您遇到类似 **The type R is already defined** 这样的错误, 那么您可能需要删除 **src/** 文件夹中老旧的 **R.java** 或以前自动生成的**AIDL java** 文件。(这不包括您手动创建的那些**AIDL java** 文件)

请注意，在Android 1.5 SDK中用Android模拟器运行应用程序之前有一个新流程，具体地讲，您必须在启动一个模拟器实例之前创建一个Android Virtual Device (AVD)。若您试图在新版SDK上运行应用程序，请继续参考如下 [迁移应用程序](#) 这节内容。

3.8 Ant用户

如果您使用Ant工具建立项目（而非Eclipse），注意在新版SDK tools中已发生如下变更。

3.8.1 build.xml已改变

您必须重新创建 build.xml文件。

如果您曾自行修改过build.xml，最好先备份一下：

```
$ cd my-project
$ cp build.xml build.xml.old
```

此时，您可以使用新的android 工具 (位于 *your_sdk/tools/* 目录中) 来创建一份关联特定的平台的新build.xml文件：

```
$ android update project --path /path/to/my-project --target 1
```

参数"target"的值指定了一个特定版本的Android类库（包括一些扩展，比如Google APIs），您可以基于此创建项目。可以使用android list targets命令查看可用的平台版本（及对应的整数ID）。若您是初次更新项目到新版的SDK，我们推荐您选择target ("1")，即基于Android 1.1 版的类库平台进行更新。

在您初次创建项目时，将会新建一个gen/文件夹，里面包含 R.java 和一些系统自动生成的AIDL java 文件。您必须删除src/ 文件夹中老旧的 R.java 或以前自动生成的AIDL java 文件。(这不包括您手动创建的那些AIDL java 文件)

注意: 原"activitycreator" 已经被新工具 "android" 取代。欲了解更多使用android 工具创建新项目的信息，详见这份 [Developing In Other IDEs](#) 文档。

另外，在Android 1.5 SDK中用Android模拟器运行应用程序之前添加了一个新步骤，具体地讲，您必须在启动一个模拟器实例之前创建一个Android Virtual Device (AVD)。若您试图在新版SDK上运行应用程序，请继续参考如下 [迁移应用程序](#) 这节内容。

3.9 迁移应用程序

当您更新完项目后，也许已经等不及想在基于Android 1.5 系统镜像的模拟器上运行您的应用程序。当您真的在Android 1.5 中运行这些应用程序的时候，有可能（不过，不太确定）会遇到一些问题。系统改动可能会影响您应用程序的运行，且不管您相信与否，都很有必要在Android 1.5上为您的应用程序进行前向兼容性（forward-compatibility）测试。

为了进行前向兼容性（**forward-compatibility**）测试，您可以在基于Android 1.5 系统镜像的模拟器中简单地试运行(**as-is**)现有的应用程序。以下步骤将会指导您按流程在模拟器中运行已有的应用程序。所以请您在开始之前仔细地阅读以下指南。

在Android 1.5模拟器上测试您的应用程序：

1. 更新您的项目 (参考以上章节，您必须已经完成此操作)。
2. 试着在基于Android 1.5 系统映像的模拟器上运行您现有的项目。
在前面，我们提及了如何 [更新您的项目](#)，您应该已经选择Android 1.1 系统映像以便编译您的应用程序，所以，在您应用程序的代码中应该不会出现新的错误。
Eclipse 用户：参考 [Eclipse guide to Running Your Application](#).
Ant 用户：参考 [Ant guide to Running Your Application](#)
在您运行应用程序期间，为AVD选择一个包含Android 1.5平台的"deployment target"。如果您的应用程序用到Google Maps APIs (比如MapView)，请一定要选择包含了Google APIs的目标项。
完成以上步骤后，回到此处继续往下阅读。
3. 当您的应用程序运行在模拟器上的时候，请为您的应用程序执行所有常规测试以确保功能不出问题（即使是山水风景和图像的方位）。

机会有的是，只要您的应用程序在Android 1.5上运行良好，一些新设备将能够很顺利地安装并运行您的应用程序，且这些用户中当他们更新后也将能一如继往地使用你的应用程序。然而，假使它不能如您期望的方式工作；那么，您或许要复查下您的项目抑或是做一些必要的代码改动。

您可以在Eclipse中打开项目来查看由于API的改动所引起的代码破损情况，尝试更换"build target"至Android 1.5平台，看看代码中出错的确切位置。

3.10 让应用程序保持最新

此SDK发行版中增加了数个API，但是只有少许API作了一些实际改动。仅一组(相对闲置的)元素被移除且取消了个别过时的，所以您基于Android 1.1系统内库所开发的应用程序应该能良好地工作。然而，如果您的应用程序运行在Android 1.5上且符合以下任何一种情况，那么有可能会出现问题。

- 使用内部APIs。需要说明的是，这些APIs并未得的正式地支持且并未写入参考文献。任何非正式的APIs总是比较容易发生变更（这也是将它们列为非正式的原因）而事实上确实有一些已经改过了。
- 直接操作系统设置。在过去，有些设置（例如,GPS, 移动业务, 蓝牙等等）是对应用程序可写的；但当如今变更后，使得它们只能被系统设置的用户所修改。您可以查阅 [Settings.Secure](#)这篇文档了解哪些设置是受安全保护的且不能直接地被您的应用程序所修改。
- 过度地滥用视图层（比如深度超过10个层级或总数超过30）。处理如此复杂的视图层一直都很麻烦，但是Android 1.5却更容易由此引发问题从而可能导致您的应用程序崩溃。
- 过分依赖硬件。随着软键盘被支持，考虑到并非所有的硬件设备都将具备完整的键盘按键。当您的应用程序要侦听特定的击键行为，在没有键盘可用的情况下，您的应用程序对此将束手无策。
- 基于重力感应加速器（或其他感应器）来改变原本的视图朝向。一些运行Android 1.5的设备将能自动旋转方向（所有设备都能可选地开启自动旋转），如果您的应用程序试图旋转朝向，可能会发生不可预知的行为。另外，如果您的应用程序使用重力感应加速器来侦测摇动并且您不期望方向能够旋转，那么您应该锁定当前朝向并参考[android:screenOrientation](#)。

关于以上问题，可以阅读我们博客上的这篇文章[Future-Proofing Your Apps](#)了解更多信息。

欲了解关于Android 1.5 的其他改动，参考以下文档：

- [Android 1.5 API Differences](#)
- [Android 1.5 Version Notes](#)
- [UI framework changes in Android 1.5 »](#)

如果您在更新代码中遇到其他问题，可以访问[Android Developers Group](#)向其他Android开发者寻求帮助。

4. 系统要求

翻译者：404

原文地址：

http://developer.android.com/sdk/1.5_r1/requirements.html

以下各小节描述了使用Android发行版中的开发工具包（SDK）来开发Android应用程序所需要的系统环境和相关软件。

4.1 支持的操作系统

- Windows XP（32位 或 Vista（32 -或64位）
- Mac OS X 10.4.8 或以上版本（仅限X86）
- Linux操作系统（已在 Ubuntu Dapper Drake 发行版上通过测试）

4.2 支持的开发环境

- Eclipse IDE
 - o [Eclipse 3.3](#)（Europa）, [3.4](#)（Ganymede）
 - § Eclipse的[JDT](#) 插件（包含在大多数 Eclipse IDE 软件包中）
 - § [WST](#)（可选, 但是需要 Android 编辑器功能； 包含在[大多数 Eclipse IDE 包](#) 中）
 - o [JDK 5 或 JDK 6](#)（仅JRE还不够）
 - o [Android 开发工具插件](#)（可选）
 - o 不兼容Java GNU编译器（gcj）
- 其他开发环境或 IDEs
 - o [JDK 5 或 JDK 6](#)（仅JRE还不够）
 - o Linux和Mac需要 [Apache Ant 1.6.5](#) 或以上版本，Windows 则需要1.7或更高版本
 - o 不兼容Java GNU编译器（gcj）

注意：如果你已经在开发机器上安装了JDK，请检查下JDK的版本以确保符合以上要求。要特别注意，一些Linux发行版可能已经包含了JDK 1.4或Java GNU编译器，而这两项都不适合做Android开发。

5. Android 1.5SDK 发布记录

翻译者：海阳|haiyangjy

原文地址：

<http://developer.android.com/sdk/RELEASENOTES.html>

新的SDK发布不但对开发工具进行了升级，而且对Android系统也做了升级。用新的SDK可以开发出适配不同不同设备配置的应用。

5.1 发布概览

新的SDK包含了一些新的特性，改变的一些亮点包括：

- 包含Android1.1和Android1.5两个版本的Android开发平台。开发工具也进行了升级，这样你可以同时部署不同版本的应用。
- 引入了Android虚拟设备(AVD),通过配置AVD，模拟器可以更好的模拟设备。每一个AVD拥有自己的存储区域，这样可以同时启动多个模拟器并行工作。
- SDK扩展。你可以扩展SDK，并且可以访问Android的内部库。另外模拟器也可以运行特制作的固件。
- 新的eclipseADT开发插件(0.9版本)提供了新的新建向导，这样你在新建项目的时候就可以选择特定的Android配置。也可以通过这个插件生成XML的资源（包括layout，animations,和menu），生成可选的布局，可以导出并签名即将发布的程序。
- 改善了ADT对JUnit的支持。
- 性能分析变的更加容易
- 程序本地化的管理功能更加容易。在你构建你的应用的应用的时候你可以包含或者排除本地化资源。
- 引入一个名为“android”的工具来代替以前的activitycreator 脚本。

5.2 安装和升级记录

如果你一直使用Android1.1SDK进行应用的开发，那么如果要迁移到新的SDK上的话，你需要对你的开发环境做些改变。我们提供了工具和文档来帮助你。假如你的应用没有用到Android的内部数据结构或者API，那么你并不不需要改变目前的源代码。

为了确保你当前的程序能够在最新的Android平台上正常运行，我们强烈鼓励你将应用迁移到新的SDK上，采用的和原来API等级一样的Android库，并且在最新的平台上运行你的程序。

如果你是第一次安装Android的SDK，那么你可以参见[Installing the SDK](#)文档。

5.3 SDK 扩展

这个版本的SDK支持扩展。你可以扩展SDK，并且可以访问的到Android的内部库。另外模拟器也可以运行特制的固件。SDK扩展机制的目的是给你一种开发特定设备应用的方式，这种特定设备可以给应用提供通过访问外部库或者系统定制化所能使用的API。

从Android开发环境的角度来看，一个SDK扩展版本和包含在SDK中的Android其他平台是类似，它也包括一个外部库，一个系统的固件，个性化的模拟器皮肤和系统的其他属性。这个扩展版本和标准版本不一样的地方在于，它可以提供一些个性化的UI、资源和操作表现行为、不同的预装程序或者其他类似的修改。

当前的SDK中包含了一个SDK扩展版-Google API 扩展版。这个扩展版可以让你的程序访问 `com.google.android.maps` 外部库，这个库存在于大多数的基于Android系统的手机上。Google API 扩展版提供了 [Geocoder](#) backend service的实现。更多的信息可以看下边"Maps External Library"的部分。

5.4 Android虚拟设备(AVDs)

1.5的SDK让你有能力编译可以运行在不同目标库上的程序。这里有中版本的目标库(target)。

- 代表核心Android平台版本的目标库。
- SDK扩展版的目标库。SDK扩展版本可以让程序访问一个或者多个外部库或者定制的(但是兼容的)系统固件，这个固件可以运行在模拟器上。

一个名叫"android"的新工具用于发现可以使用的目标库(target)和AVD

如果你想了解更多的关于AVD的信息，请参看[Android Virtual Devices](#)。

5.5 其他的一些记录：

5.5.1 地图外部库：

在以前的SDK版本中，`com.google.android.maps`包包含在标准的Android库和系统固件当中。在Android 1.5 SDK中，情况有变。Android 1.5的库和系统固件没有包含地图的外部库。但是地图的外部库可以在Google SDK 扩展版本中获得。下载地址为：<http://code.google.com/android/add-ons/google-apis>

关于如何注册Maps API Key的信息，可以参见：[Obtaining a Maps API Key](#).

5.5.2 Window中的USB驱动

如果你使用的是Windows，并且在基于Android操作系统的设备（比如G1）里边开发或者测试，你需要安装USB驱动。为了让你使用方便，Windows版本的SDK包含了USB的驱动。

- 32位支持XP和Vista的驱动
- 64位支持Vista的驱动

USB的驱动文件放在 `<SDK>/usb_driver` 目录下。细节和安装指南请参见[Setting Up a Device for Development](#).

5.5.3 解决的问题和改变

媒体

- 更新了 [android.media.SoundPool](#) 的文档。
- [WebView](#) 对象不再自动保存缩略图。只有当你手动调用 [capturePicture\(\)](#) 的时候才可以。

已知的一些问题：

在模拟器中的感应器问题

- 如果你的程序用到了**Sensor API**，那么程序在基于**Android 1.5**固件的模拟器中是有问题的。你的程序可能会产生出**ANR**消息，或者直接会崩溃掉。这个问题已经开始调查了。
- 其他的
- 我们很抱歉告知在**Android 1.5**中我们将不再支持**Zilog Z80**处理器结构。

6 Android 1.5 版本记录

翻译者：海阳|haiyangjy

原文地址：

<http://developer.android.com/sdk/android-1.5.html>

这份文档主要描述了Android 1.5版本变化的一些记录。主要包括以下几个方面的内容：

- 概览
- 外部库
- 设备兼容性
- 内置的应用
- UI 的本地化
- 新特性
- API的改变

6.1 概览

Android 1.5 提供发布了更新的框架版本号。就像以前的版本号一样，这次1.5版本的API的版本号是：3（整数字符），这个字符保存在系统内部。我们叫这个字符为API等级，这种机制的好处是我们可以安装程序前系统可以正确的判断出当前即将被安装的程序是否和系统所兼容。

应用可以在manifest 文件当中来指定一个API等级，这样就可以告诉提供可以运行此程序的Android系统的最低版本。为了定义最小API等级，在程序的manifest文件中引入了minSdkVersion一个新的 属性。这个属性的值是个整数，并且和API等级相对应。在安装程序前，系统会检查minSdkVersion的值，当这个值比系统中保存的API等级低或者他们相等的时候，系统才会安装此程序。

当你把你的程序迁移到新的SDK的时候，你必须在编译程序之前选择一些平台版本。通常你只需要在支持你的应用的最低版本号的平台上编译此应用。当你确定好这个最低的版本号后，你必须确保你将这个值写入到了manifest 的minSdkVersion属性中。（译者注：如果不设置这个值，无法将程序上传到market里边）。

当你编译完程序后，最好在你minSdkVersion制定的平台上测试一下你的程序。并且，为了保证向前兼容，你应该在比你的minSdkVersion指定的更高API级别的平台中运行此程序。为了在模拟器中测试不同的平台，你可以为每一个版本的平台新建一个AVU。关于AVU的知识请参照SDK中的[Android Virtual Devices](#)文档。如果你直接用手机而不是模拟器来测试的话，你需要知道当前手机上跑的Android平台的API级别是多少。（译者注：大部分G1的版本是1）

如果你的应用用到的是Android 1.5的库并且你的应用程序用到了在3级API 中的APIs，你**必须**将你的android:minSdkVersion属性设置为"3".

你可以将android:minSdkVersion属性定义在<manifest>节点下的子节点<uses-sdk>中，当你设置完后，看起来应该如下所示：

```
<manifest>
...
<uses-sdk android:minSdkVersion="3" />
...
</manifest>
```

通过以这种方式设置`android:minSdkVersion` 属性后，你可以当用户手机的Android系统和此程序兼容的情况下，用户才可以安装此程序。反过来这样可以确保你的程序能在用户的手机上运行良好。这是非常重要的，尤其是当你的程序用到了在Android1.5中改变的或者新的API。

如果你的程序用到了Android1.5中的API，但是没有如下的声明：`<uses-sdk android:minSdkVersion="3" />`，那么你的程序在基于Android1.5平台的设备上运行良好的，但是却不能运行在Android1.0的平台上。

反过来，如果你的程序没有使用在Android1.5中新的API，应用很可能在随后的平台版本上运行正常。但是如果你已经发布了应用，我们强烈建议你在Android1.5的固件和SDK上测试运行一下。如果用户将自己的固件升级到Android的一个新的版本上，上边的操作可以保证用户平滑的过渡。

最后，如果你的应用程序没有使用Android1.1版本后的新的API，那么你可以在你的应用程序中将`android:minSdkVersion`属性设置为"2"。如果你的程序没有使用Android1.0版本后的新API，那么那么你可以将上边这个属性设置为"1"。但是在你发布应用前，你必须确保在`android:minSdkVersion`属性所对应的Android库中编译你的程序。

6.2 内置的程序

系统的固件内置了一下程序：

- 闹钟(Alarm Clock)
- 浏览器(Browser)
- 计算器(Calculator)
- 摄像机(Camcorder)
- 照相机(Camera)
- 联系人(Contacts)
- 本地化程序Custom Locale (developer app)
- 开发工具Dev Tools (developer app)
- 打电话 Dialer
- 邮件Email
- 目录浏览器Gallery
- 短信息Messaging
- 音乐Music
- 设置Settings
- Spare Parts (developer app)

6.3 本地化

系统固件提供了非常多的本地语言支持。在一些情况下，甚至区域性的语言词组都是支持的。在其他的情况下，使用默认版本的语言。Android1.5固件支持的语言如下：

- Chinese, PRC (zh_CN) 中文简体
- Chinese, Taiwan (zh_TW) 中文繁体
- Czech (cs_CZ) 捷克语
- Dutch, Netherlands (nl_NL) 荷兰语
- Dutch, Belgium (nl_BE) 荷兰语 比利时
- English, US (en_US) 英语(美国)
- English, Britain (en_GB) 英语(英国)
- English, Canada(en_CA)英语(加拿大)
- English, Australia (en_AU)英语(澳大利亚)
- English, New Zealand (en_NZ)英语(新西兰)

- English, Singapore(en_SG)英语(新加坡)
- French, France (fr_FR)法语(法国)
- French, Belgium (fr_BE)法语(比利时)
- French, Canada (fr_CA)法语(加拿大)
- French, Switzerland (fr_CH)法语(瑞士)
- German, Germany (de_DE)德语(德国)
- German, Austria(de_AT)德语(澳大利亚)
- German, Switzerland (de_CH)德语(瑞士)
- German, Liechtenstein (de_LI)德语(Liechtenstein)
- Italian, Italy (it_IT)意大利语(意大利)
- Italian, Switzerland (it_CH)意大利语(瑞士)
- Japanese (ja_JP)日语
- Korean (ko_KR)韩语
- Polish (pl_PL)波兰语
- Russian (ru_RU)俄语
- Spanish (es_ES)西班牙语

6.4 新特性

系统的新特性，请参见 [Android 1.5 Platform Highlights](#) 文档。

5.API 的变动

5.1 概览

- UI框架
 - 和后台的UI线程交互变的更加方便。
 - 增加了 [SlidingDrawer](#) 小组件(widget)
 - 新增了 [HorizontalScrollView](#) 小组件(widget)
- AppWidget 框架
 - 增加了新建桌面AppWidget 的API。关于如何使用AppWidget 请参照开发指南 [AppWidgets](#) 。并且可以参照Android 开发者博客：[Introducing home screen widgets and the AppWidget framework](#) 。
 - 增加组装拥有个性化内容的[Live Folders](#) 。
- 媒体框架
 - 增加原始音频录音和回放的API
 - 可交互的MIMD回放引擎
 - 开发者可用的视频录制API（支持格式为3GP格式）
 - 视频和图片分享的Intents
 - 增加多媒体搜索的Intent
- 输入法框架
 - 输入法服务框架([Input Method Service](#))
 - 文本预测引擎(Text-prediction engine)
 - 支持可下载的IMEs
- 程序可以定义对硬件的需求
 - 应用程序可以在manifest 文件当中用<uses-configuration> 来提示Android系统为了运行程序需要什么样的硬件条件。比如，一个程序可以用这个xml元素来确定自己需要使用物

理键盘或者一个特殊的浏览设备，比如trackball。在安装程序前，Android系统会检查<uses-configuration> 元素，当硬件条件都满足的时候才会继续安装程序。

- 语言识别框架
 - 支持通过Intent使用语言识别库。
- 其他
 - **LocationManager** -应用程序可以通过Intent来或者所在地点改变的信息。
 - **WebView** -支持 Touch的start/end/move/cancel 的DOM事件。
 - 重新设计了 **Sensor Manager** 的APIs。
 - **GLSurfaceView** -方便的创建OpenGL程序的框架。
 - 新增应用更新安装成功的广播-提高App升级更加平滑的体验。

6.API 改变细节

6.1新增加的包

Added Packages

android.appwidget
android.inputmethodservice
android.speech
android.text.format
android.view.inputmethod
java.beans

6.2改变的包

Changed Packages

android
android.app
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite
android.graphics
android.graphics.drawable
android.graphics.drawable.shapes
android.hardware
android.location
android.media
android.net
android.net.wifi
android.opengl
android.os
android.preference
android.provider
android.telephony
android.telephony.gsm
android.test
android.test.mock

android.test.suitebuilder
android.text
android.text.method
android.text.style
android.util
android.view
android.view.animation
android.webkit
android.widget
dalvik.system
java.lang
java.lang.reflect
java.net
java.util
java.util.jar
[java.util.logging](#)

7. Android 1.5平台的亮点

Android 1.5平台给用户和开发者带来了一些新的特性。下边的列表我们将提纲挈领的看一下改变有哪些。

- 用户UI界面的改善
- 性能的改进
- 新的特性
- 新的API和Manifest文件中可定义的元素

7.1 用户UI界面的改善

- 整个系统来看
 - 所有的核心UI元素都进行了改善。
 - 窗口的切入和切出可以增加动画效果。(动画效果默认是关闭的)
 - 基于加速器的程序旋转。
- 更加优化的UI包括：
 - 接打电话的用户体验
 - 联系人，电话记录和收藏应用
 - 短信和彩信应用
 - 浏览器应用
 - Gmail应用
 - 日历应用
 - 邮件应用
 - 照相机和分类应用

7.2 性能的改善

- 照相机的启动速度更快，牌照的速度也更快。
- GPS定位速度更快
- 在拖动浏览器的时候页面的滑动更加平滑
- 在滑动浏览Gmail对话列表的时候速度更快

7.3 新特性

- 屏幕内的软键盘
 - 手机横向和纵向都可以使用。
 - 支持用户安装第三方的软件盘。
 - 可定制个性化词语的用户词典
- 桌面
 - Widget
 - 绑定到桌面的widget小组件包括:模拟闹钟，日历，音乐播放器，图片略景和搜索。
 - Live 文件夹
 - 照相和分类
 - 录像(MPEG-4 & 3GP 格式)

- 录像的回放。
- 蓝牙
 - 支持立体蓝牙(A2DP and AVCRP profiles)
 - 自动配对
 - 改进的蓝牙握手匹配用户体验
- 浏览器
 - 更新至最新的Webkit 浏览器和Squirrelfish Javascript 引擎
 - 支持在浏览器当中进行拷贝和粘贴操作
 - 在一个网页内部进行搜索
 - 用户可选的输入文字编码
 - UI的改变包括：
 - 统一了Go和Search的UI表现
 - 可以以Tab的形式显示书签，历史和最近常访问的网页
- 联系人
 - 可以显示收藏用户的头像。
 - 可以指定在通话记录中记录的时间戳。
 - 从通话记录可以通过一次点击就可以访问到联系人的详细信息。
- 系统
 - 新的Linux 内容 (版本为2.6.27)
 - sd卡中，文件系统的自动检测和修复
 - SIM卡程序工具包 1.0版本
- Google应用程序（包含在Android SDK中，但是并没有包含在Android1.5的系统固件中）
 - 可以在下边的程序中看到联系人的Gtalk状态，这些程序包括 联系人应用、短信应用、彩信应用、Gmail、和邮件应用程序
 - 可以对批量对邮件进行如保存，删除，打上标记的操作
 - 上传视频到Youtube
 - 上传图片到Picasa

8. Android 1.1 版本说明

翻译者：ddna

原文地址：<http://developer.android.com/sdk/android-1.1.html>

日期：2009年2月

API等级：2

本文档提供了SDK包含的Android 1.1 版本系统镜像的说明。

- 概述
- 外部库
- 设备兼容性
- 内置应用
- UI本地化
- 已解决的问题
- 新特性
- API更改

8.1 概述

本次随SDK发布的Android 1.1系统镜像是Android 1.1产品的开发用配套镜像，可以用来开发2009年2月份以后的Android手机。

Android 1.1系统镜像更新了框架API的版本。和1.0 API一样，Android 1.1 API分配了一个整型标识2在系统内部，这个标识被称为：“API 等级”，它可以使系统在安装应用之前判断这个应用是否和系统兼容。

应用程序通过添加minSdkVersion属性标识出其兼容的系统API的最低等级。这个属性和API等级相对应。安装应用之前，系统会检查minSdkVersion的值，只有在这个值小于等于系统自身的API等级时才允许安装。

如果你要使用Android 1.1系统镜像创建一个和基于1.1平台的Android设备兼容的应用，你必须设定其minSdkVersion属性值为“2”，这样你的应用就和使用Android 1.1(或者更高)系统镜像的设备保持了兼容性。

你可以在manifest文件里<manifest>的子节点<uses-sdk>中设置android:minSdkVersion属性。属性设置如下：

```
<manifest>
...
<uses-sdk android:minSdkVersion="2" />
...
</manifest>
```

通过按这种方式设定android:minSdkVersion，你可以确保只有运行在Android 1.1平台上的设备才能安装你的应用。反过来说，如果你的应用使用了Android 1.1引入的新API，这个机制保证了你的应用可以在他们的设备上正常运行。

如果你的应用使用了Android1.1引入的新API，但是没有申明`<uses-sdk android:minSdkVersion="2" />`，它能够正常运行在Android1.1设备上，但是不能够正确运行在Android1.0设备上。在后一种情况下，应用将在调用Android1.1的API时运行错误。

如果你的应用没有使用任何新的Android1.1API，你可以通过去除minSdkVersion属性或者设定其属性值为“1”来表明和Android1.0的兼容性。但是，在发布你的应用之前，必须确保使用Android1.0系统镜像(可以在Android1.0SDK得到)编译该应用以保证它可以在Android1.0设备上创建并正常运行。你需要测试应用和所依赖的API等级相对应的系统镜像的兼容性。

如果你的应用没有使用Android1.1API，同时也不需要它们，你可能会发现使用Android 1.0 SDK比 1.1 SDK更容易(因为不需要做SDK从1.0到1.1的移植和相应增加的测试)。

8.2 外部库

系统镜像包括的外部库可以通过在应用里加上`<uses-library>`来访问。

`com.google.android.maps` - 让你的应用可以访问Google Maps数据。注意，使用Google Maps数据需要一个Maps API的键值 (key)。

8.3 设备兼容性

Android1.1系统镜像已在下面列出的Android设备上测试过兼容性T-Mobile G1

8.4 内置的应用

系统镜像包含了以下几项内置的应用：

- 闹钟
- API 示例
- 浏览器
- 计算器
- 照相机
- 通讯簿
- 设备工具
- 电话
- 邮箱
- 地图(街道视图)
- 短信息
- 音乐
- 图片
- 设置

8.5 UI 本地化

系统镜像针对如下语言提供了本地化UI字符串：

- 英语(美国)
- 德语

本地化UI字符串可以匹配模拟器支持显示的区域(Locale)，这些区域可以通过设备的Settings应用设置。

8.6 修正的问题

- 闹钟提醒现在可以直接播放音频/振动，而不再通过AlarmManager。同时闹钟在它的IntentReceiver里启动播放音频/振动，而不是在Activity启动的时候。这些改变可以阻止从模式对话框中出现的告警。
- 修正了设备休眠的问题。
- 单击不再会打开内置的拨号键盘，用户现在需要触摸以拖出它。
- 修正了一个概率大约为1/25的问题：在将发送短信移至已发送文件夹时会冻结IMAP连接(至一个Gmail服务器)。
- 删除自动帐户建立的已损坏和不可测试的一些设置项。对一些保留项稍微做了修改，改进了一些特殊情况下的告警对话框。
- 将邮箱的检查时间间隔由默认的“从不”改为每15分钟一次。
- 修改了IMAP使用的密码引用问题，用户可以在密码里包含特殊的字符(如：空格)
- 修改了一些在自动和手动帐户设置时的错误。
- 改进一些连接错误的报告，方便用户更容易地诊断帐户设置错误。
- 修正了POP3帐户的新邮件通知。
- 确保能够对设定了“从不检查”帐户做合适的自动检查。
- 现在可以通过用户参数设置显示日期和时间(如：24二十四小时 AM/PM)。
- 现在可以在消息框中显示抄送。
- 改进了从POP3链接失败中恢复的问题。
- POP3解析宽泛的规则，使应用可以运行在一些特殊的email服务器上。

8.7 新特性

- 地图：添加了细节显示和评论，当用户搜索地图并点击查看具体某一处时。
- 电话：当使用扬声器通话时屏幕默认不会超时关闭。
- 电话：添加了一个“显示拨号键盘”/“隐藏拨号键盘”项在通话菜单中，可以更轻松地发现DTMF拨号键盘。
- 添加了彩信的附件保存功能。
- 添加了布局内容的滚动显示支持。

8.8 API 改动

概要

- 添加了测试系统的注释，API实际并没有改动(未测试)。
- 添加了一个方法使进程可以简单地确定其UID。
- 添加了布局内容的滚动显示支持。
- 添加了新方法确定视图的补白(padding)，这对手动实现View子类很实用。
- 添加了新的许可允许一个应用广播彩信和WAP 推送信息。
- API清理：删除了SDK自带的系统镜像的protected构造器。

8.8.1 API更改细目：

模块和特性	更改描述
-------	------

测试系统注释	添加 LargeTest 注释.
	添加 MediumTest 注释.
	添加 SmallTest 注释.
使进程更容易地得到 UID	添加 public method myUid() 至 class android.os.Process
视图补白(Padding)	添加 public method getBottomPaddingOffset() 至 class android.view.View.
	添加 public method getLeftPaddingOffset() 至 class android.view.View.
	添加 public method getRightPaddingOffset() 至 class android.view.View.
	添加 public method getTopPaddingOffset() 至 class android.view.View.
	添加 public method isPaddingOffsetRequired() 至 class android.view.View.
滚动显示支持	添加 public method setMarqueeRepeatLimit(int) class TextView
	添加 public field android.R.attr.marqueeRepeatLimit
新许可	添加 public field android.Manifest.permission.BROADCAST_SMS
	添加 public field android.Manifest.permission.BROADCAST_WAP_PUSH
API 清理	删除 protected constructor java.net.ServerSocket.ServerSocket(java.net.SocketImpl).

9. 其他

翻译人员：

- lceskYsl(<http://iceskysl.1sters.com>)
- haiyangjy(<http://www.haiyangjy.com>)
- 404(<http://l404.blogspot.com>)
- ddna

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=153>

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，想为Android在中国的发展尽自己的微薄之力。