HZ BOOKS
华章科技

拥有近10年从业经验的资深软件开发工程师和测试工程师经验结晶，系统且实用，为开发高质量Android应用提供全方位指导

从测试和调试两个维度深度探索Android系统，为Android应用的自动化测试和调试提供原理性的解决方案

移动开发

施懿民◎著

Android Application Testing and Debugging

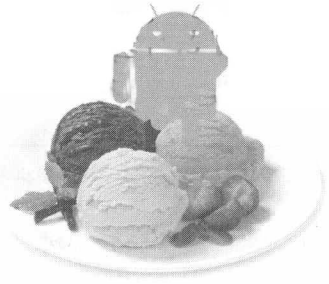
Android应用测试与调试实战

机械工业出版社
China Machine Press

Android Application Testing and Debugging

Android应用测试与调试实战

施懿民◎著



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Android 应用测试与调试实战 / 施懿民著 . —北京：机械工业出版社，2014.3
（移动开发）

ISBN 978-7-111-46018-3

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

中国版本图书馆 CIP 数据核字（2014）第 041414 号

本书是 Android 应用测试与调试领域最为系统、深入且极具实践指导意义的著作，由拥有近 10 年从业经验的资深软件开发工程师和调试技术专家撰写，旨在为广大程序员开发高质量的 Android 应用提供全方位指导。它从 Android 应用自动化测试工程师和开发工程师的需求出发，从测试和调试两个维度，针对采用 Java、HTML 5、C++&NDK 三种 Android 应用开发方式所需要的测试和调试技术、方法进行了细致而深入的讲解，为 Android 应用的自动化测试和调试提供原理性的解决方案。

全书一共 16 章，分为两大部分：第一部分为自动化测试篇（第 1~11 章），详细讲解了进行 Android 自动化测试需要掌握的各种技术、工具和方法，包括 Android 自动化测试基础、Android 应用的白盒自动化测试和黑盒自动化测试的技术和原理、Android 服务组件和内容组件的测试、HTML 5 应用和 NDK 应用的测试，以及 Android 应用的兼容性测试和持续集成自动化测试；第二部分为调试技术篇（第 12~16 章），详细讲解了 Android 应用调试所需要的各种工具的使用、操作日志的分析、内存日志的分析，以及多线程应用 HTML 5 应用和 NDK 应用的调试方法和技巧。

Android 应用测试与调试实战

施懿民 著

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：姜 影
印 刷：藁城市京瑞印刷有限公司 版 次：2014 年 4 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：27.75
书 号：ISBN 978-7-111-46018-3 定 价：79.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

前　言

为什么要写这本书

几年前看过微软工程师 Adam Nathan 写的一本书《 .NET and COM: The Complete Interoperability Guide》，对作者说过的一句话印象极其深刻：学习一门新技术的最佳方法就是写一本书。对于这个说法，笔者深以为然，在实际工作中，迫于项目交付的压力，很多时候只会使用一门技术中自己熟悉的部分，对于其他部分，甚至自己熟悉的那一部分，也只是知其然而不知其所以然，似懂非懂。而当要写一本书介绍这门技术的时候，就不得不去上下求索，这时不仅要横向了解该门技术的各个细节，还要纵向了解该门技术的发展思路以及各部分的来龙去脉，才能向读者解释它。因此可以说，写作本书的过程也是笔者系统学习 Android 操作系统的过程。

而笔者之所以从测试和调试的角度来介绍 Android 操作系统，是因为从这两个角度入手能够很好地从横向和纵向两个方向学习 Android 系统。从测试的角度来说，由于一般测试人员对 Android 应用进行集成测试和系统测试，开发人员负责单元测试工作，这就要求测试人员，特别是自动化测试人员对 Android 应用所涉及的技术有一个广度的认知，对 Android 的各种技术都要知其然。从调试的角度来说，却又可以从源码级别知各项技术的所以然。比如，要调试应用的功能错误，就必须对应用的源码有一个完整的理解，只有这样才能知道应该在什么地方设置断点，在什么时候让调试器捕捉异常；而要调试应用的内存泄露问题，就必须对 Android 系统采用的垃圾回收机制有一个通透的理解，只有这样才能从根源上发现并解决内存方面的问题。另外，反观市面上的技术书籍，介绍自动化测试和调试技术的书实在少。但在软件开发过程中，不仅测试是必不可少的环节，而且程序员实际上只花 20% 的时间写代码并完成编译过程，剩下 80% 的时间都是花在调试和修改代码上，这种情况也坚定了笔者写作本书的信心。

最后，由于笔者能力有限，同时为了及时出版本书，不得不放弃一些如多核设备上并行程序的调试、NDK 程序的验尸调试等内容，希望感兴趣的读者可以完成这方面的分享。

读者对象

- Android 自动化测试工程师
- Android 开发工程师
- 无编程基础的 Android 测试工程师

如何阅读本书

虽然本书讲解的是自动化测试和程序调试相关技术，但有些测试工程师的行业经验更丰富些，而编程基础则相对薄弱。因此本书分为两大部分：

第一部分为自动化测试篇（第 1 ～ 11 章），这一部分列举 Android 自动化测试中可以使用的几种测试技术，尽可能详细地介绍了 Android 白盒、黑盒自动化测试所用到的技术及其原理。由于 Android 应用可以使用 Java 语言配合 SDK，也可以用 HTML 5 技术，还可以用 C/C++ 语言配合 NDK 技术编写，所以这部分尽量涵盖了针对这三种技术编写的应用所采用到的测试技术。这些内容适合 Android 自动化测试工程师和对自动化测试感兴趣的手工测试工程师阅读。这一部分除了第 11 章需要有 C/C++ 编程经验之外，其他章节无需编程基础即可阅读。另外每个章节都是独立的，读者可以根据自己的实际需要分开来阅读。

第二部分为调试技术篇（第 12 ～ 16 章），第 12 章讲解的是通用的调试技术，这部分对于 Android 自动化测试工程师和 Android 开发工程师都是必要的知识，这一章节无需编程知识即可掌握。而第 13 章之后，主要涉及的是性能方面的调试技术，其中涉及一些 Android 系统内部的实现细节，这些技术更适合具备一定开发经验的自动化测试和开发工程师阅读。

勘误和支持

由于作者的水平有限，加之编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。为此，作者特意在 Google+ 上创建一个在线支持与应急方案的社区 https://plus.google.com/u/0/communities/112928495323595574856。你可以将书中的错误发布在 Bug 勘误表页面中，同时如果你遇到任何问题，也可以访问 Q&A 页面，作者将尽量在线上提供满意的解答。书中的全部源文件除可以从华章网站[⊖]下载外，还可以从这个社区和 github 上 https://github.com/shiyimin/androidtestdebug 下载，作者也会将相应的功能更新及时发布出来。如果你有更多的宝贵意见，也欢迎发送邮件至邮箱 shiyimin.aaron@gmail.com，期待能够得到你们的真挚反馈。

⊖ 参见华章网站 www.hzbook.com。——编辑注

致谢

首先要感谢支付宝的陈晔（新浪微博：@Monkey 陳曄曄），在本书写作时，他参与了大部分章节的技术审阅工作，帮助完善了书中的细节。

感谢阿里集团的梁剑钊（新浪微博：@liangjz）推荐的玄黎（新浪微博：@ 浪头）、李子乐（新浪微博：@ 子乐 _ 淘宝太禅）参与本书技术审阅工作。

感谢机械工业出版社华章公司的编辑杨福川老师，在这一年多的时间中始终支持我的写作，他的鼓励和帮助引导我顺利完成全部书稿。

谨以此书献给我最亲爱的家人，以及众多热爱 Android 测试的朋友们！

施懿民

目　录

第 1 章
Android 自动化测试初探

本章的目的是让有经验的测试开发工程师迅速上手 Android 自动化测试代码的编写流程。因为大部分 Android 程序都是基于 SDK 开发，所以本章只介绍为基于 Android SDK 的应用编写自动化测试代码，想了解测试 Android NDK 应用的方法可阅读第 9 章。

1.1 快速入门

Android 开发环境的安装，由于种种原因，变得非常难于实现，这在一定程度上给 Android 开发的初学者带来了很大的不便。为了帮助读者将主要精力放在学习 Android 自动化测试的开发技术上，而不是将时间浪费在环境的准备和安装上，本书配套资源提供了一个 VirtualBox 虚拟机，方便读者学习和尝试本书里讲解的各种技术。本书后续章节的示例代码和示例命令，如果没有特殊说明，均使用该虚拟机演示。使用该虚拟机可参照以下步骤：

1）下载并安装最新版本的 VirtualBox：https://www.virtualbox.org/wiki/Downloads。

2）打开 VirtualBox，依次单击菜单栏的“控制”和“注册”菜单项，导入“ AndroidBook.vbox”虚拟机。

3）在 VirtualBox 中选择刚刚注册的虚拟机，单击工具栏里的“启动”按钮启动它。

4）在虚拟机上安装的是编写本书时最新的 Ubuntu 12.04 操作系统，超级用户的用户名和密码均为“student”。

5）在虚拟机中已经准备好了 Eclipse、Android 开发环境和使用真机设备的相关配置。在虚拟机中单击“ Dash”图标（或者同时按下键盘的 Alt 和 F2），并输入“ gnome-terminal”打开终端程序，如图 1-1 所示。

图 1-1 在附带虚拟机中打开终端程序

6）在终端中输入下列命令启动 Eclipse：

```
$ ~/eclipse/eclipse
```

注意

在 Windows 中，如果当前登录用户的用户名是中文，当在资源管理器中双击 VirtualBox 的 .msi 安装包尝试安装时，VirtualBox 安装程序会弹出“系统找不到指定的路径”的错误消息框，如图 1-2 所示。

图 1-2 “系统找不到指定的路径”的错误消息框

这是 VirtualBox 安装程序的一个缺陷，因为 MSI 安装程序运行时会解压一些临时文件到登录用户的临时文件夹中，而 VirtualBox 安装程序对包含中文的路径支持得不是很好，就会弹出这样的对话框，所以这里建议使用英文名的管理员用户安装。

1.2 待测示例程序

本章示例所采用的待测程序是一个简单的 Android 应用，模拟数据库程序的增删改查功能。程序的主界面是一个书籍列表界面，按作者名列出了每个作者的著作书名。在列表中单击书名可以查看书籍的详细信息，在详细信息界面上单击“编辑”按钮可以编辑书籍的信息，完成后单击“保存”按钮即可保存更改并返回到列表界面。列表界面上有“删除”和“添加”按钮，向列表中添加一本新书籍的操作与“编辑”类似；在从列表中删除一本书时，需要先单击“删除”按钮，然后再单击要删除的书籍。待测示例程序的主界面如图 1-3 所示。

待测程序的源代码可以在配套资源（或虚拟机的 /home/student/samplecode 中）的“chapter1\cn.hzbook.android.test.chapter1”文件夹中找到，按照下面的步骤在 Eclipse 中导入该工程：

1）启动 Eclipse 。

2）依次单击 Eclipse 菜单栏中的“File”、“Import…”菜单项。

3）在新弹出的“Import”对话框中选择“Existing Projects into Workspace”列表项，然后单击“Next”按钮进入下一步，如图 1-4 所示。

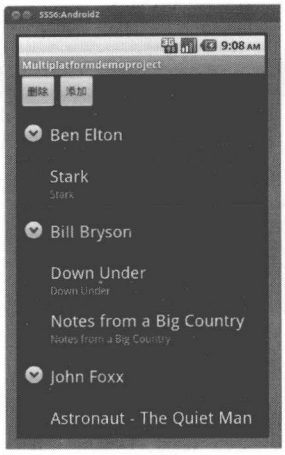


图 1-3 待测示例程序的主界面截图

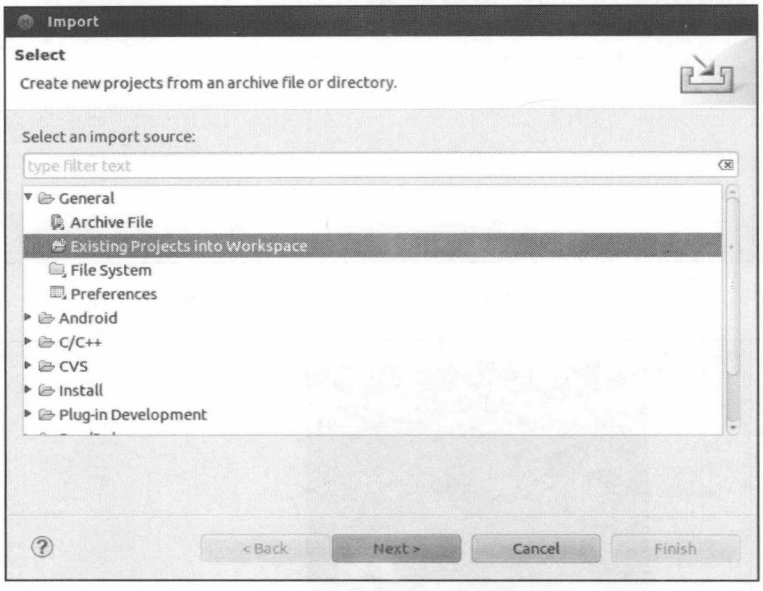


图 1-4 在 Eclipse 中导入待测示例应用工程

4）在“Import Projects”界面中勾选“Select root directory”项，并单击旁边的“Browse…”按钮，填入配套资源中待测应用源文件的根目录。完成后应该可以在“Projects”列表框中看到要导入的工程名：“cn.hzbook.android.test.chapter1”。勾选“Copy projects into workspace”复选框，以便将应用的源代码复制到本地硬盘。最后单击“Finish”完成导入操作。如图 1-5 所示。

5）完成导入后，用右键单击刚导入的工程文件，并依次选择“Run As”，“Android

Application”运行应用。如图 1-6 所示。

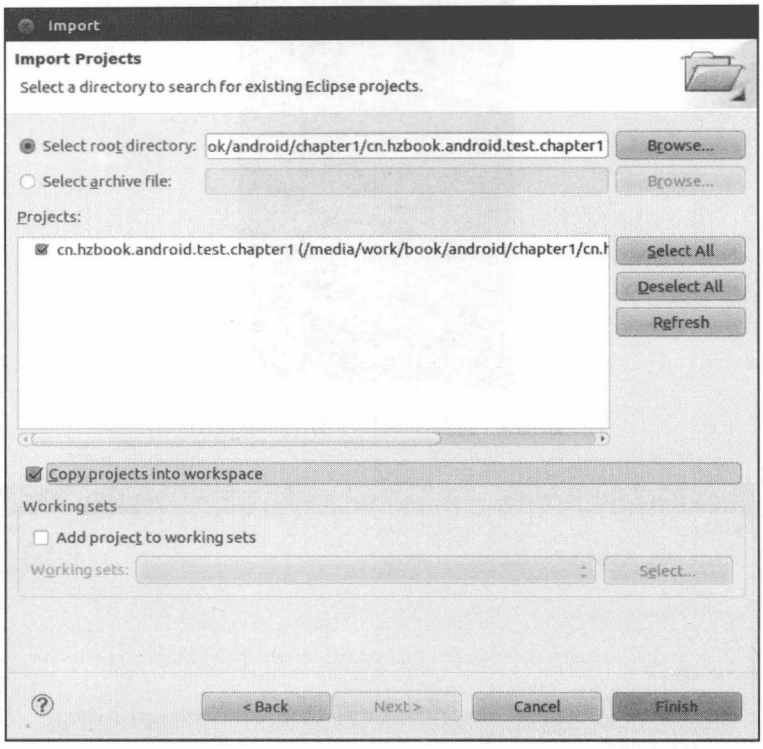


图 1-5 在 Eclipse 中导入工程向导中选择待测示例工程

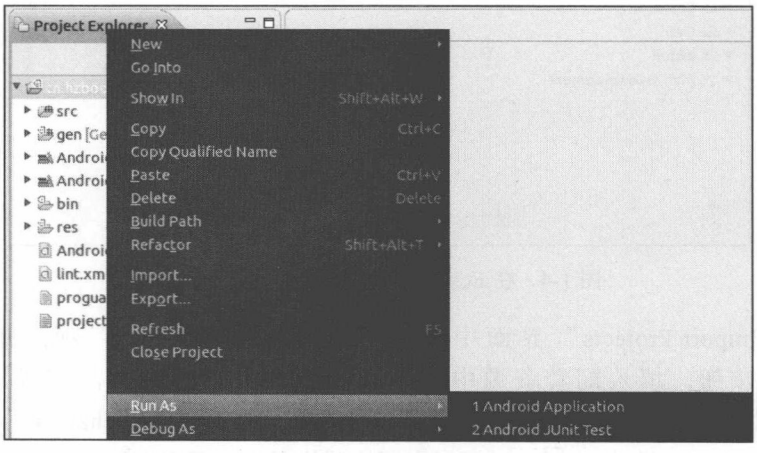


图 1-6 在 Eclipse 中运行示例待测应用

6）这时 Eclipse 会自动启动 Android 模拟器并打开应用。此时打开 Eclipse 下方的“Console”窗口并选择“Android”下拉框，应该可以看到类似图 1-7 的输出。

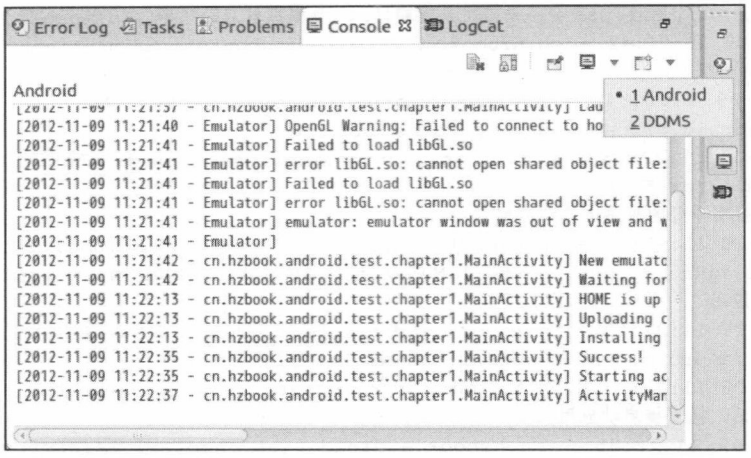


图 1-7 在 Eclipse 中启动应用的过程输出

图 1-7 的输出中详细显示了 Eclipse 从启动模拟器到运行应用的完整过程。在测试过程中，会经常用到输出内容来排查错误，下面结合注释解释输出内容：

```
    # 在这里 Eclipse 接受到启动 Android 应用的命令
    [2012-11-09 11:21:37 - cn.hzbook.android.test.chapter1.MainActivity] Android
Launch!
    #Eclipse 首先确定用来与模拟器进行通信（调试程序）的 adb 程序是否运行，如果没有运行就会启动它
    #adb 程序的用法会在后面调试章节中讲解
    [2012-11-09 11:21:37 - cn.hzbook.android.test.chapter1.MainActivity] adb is
running normally.
    # 找到应用的主 Activity 并启动它
    [2012-11-09 11:21:37 - cn.hzbook.android.test.chapter1.MainActivity] Performing
cn.hzbook.android.test.chapter1.MainActivity activity launch
    # 自动选择最合适的模拟器，这里因为示例应用最低要求 Android 2.2 版本，而系统中正好有
    # Android 2.2 版本的模拟器，所以直接做出最佳选择
    [2012-11-09 11:21:37 - cn.hzbook.android.test.chapter1.MainActivity] Automatic
Target Mode: launching new emulator with compatible AVD 'Android2'
    [2012-11-09 11:21:37 - cn.hzbook.android.test.chapter1.MainActivity] Launching
a new emulator with Virtual Device 'Android2'
    # 这些错误与 OpenGL 有关，可以忽略它们
    [2012-11-09 11:21:40 - Emulator] OpenGL Warning: Failed to connect to host.
Make sure 3D acceleration is enabled for this VM.
    [2012-11-09 11:21:41 - Emulator] Failed to load libGL.so
    [2012-11-09 11:21:41 - Emulator] error libGL.so: cannot open shared object file:
No such file or directory
    [2012-11-09 11:21:41 - Emulator] Failed to load libGL.so
    [2012-11-09 11:21:41 - Emulator] error libGL.so: cannot open shared object file:
No such file or directory
    # 模拟器会自动将自身窗口调整到屏幕的最佳位置
    [2012-11-09 11:21:41 - Emulator] emulator: emulator window was out of view and was
recentered
    [2012-11-09 11:21:41 - Emulator]
```

```
    # 在模拟器启动时，Eclipse 会不停扫描系统端口，一旦模拟器启动完毕，Eclipse 就会连接上它
    [2012-11-09 11:21:42 - cn.hzbook.android.test.chapter1.MainActivity] New emulator
found: emulator-5554
    [2012-11-09 11:21:42 - cn.hzbook.android.test.chapter1.MainActivity] Waiting
for HOME ('android.process.acore') to be launched...
    [2012-11-09 11:22:13 - cn.hzbook.android.test.chapter1.MainActivity] HOME is up
on device 'emulator-5554'
    #Eclipse 将编译好的应用上传到模拟器
    [2012-11-09 11:22:13 - cn.hzbook.android.test.chapter1.MainActivity] Uploading
    cn.hzbook.android.test.chapter1.MainActivity.apk onto device 'emulator-5554'
    # 接着安装应用
    [2012-11-09 11:22:13 - cn.hzbook.android.test.chapter1.MainActivity] Installing
    cn.hzbook.android.test.chapter1.MainActivity.apk...
    [2012-11-09 11:22:35 - cn.hzbook.android.test.chapter1.MainActivity] Success!
    # 成功安装后，就直接启动应用
    [2012-11-09 11:22:35 - cn.hzbook.android.test.chapter1.MainActivity] Starting
activity cn.hzbook.android.test.chapter1.MainActivity on device emulator-5554
    [2012-11-09 11:22:37 - cn.hzbook.android.test.chapter1.MainActivity] ActivityManager:
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.
LAUNCHER] cmp=cn.hzbook.android.test.chapter1/.MainActivity }
```

注意

模拟器启动后，默认处于锁屏状态，需要手动解锁，解锁后就会看到应用已经启动。自动解锁的方式在 10.3 节的代码清单 10-10 中说明。

1.3 第一个 Android 应用测试工程

本章先建立一个简单的 Android 自动化单元测试工程来演示 Android 自动化测试的流程。在 Android 系统中，Android 自动化单元测试也是一个 Android 应用工程，它跟普通 Android 应用工程不同的地方是启动的方式不一样，这在本书第 3 章会讲到。

Android 的 Eclipse ADT 插件提供了 Android 自动化单元测试的模板，方便我们创建自动化测试项目，这里新建一个测试工程：

1）启动 Eclipse，这次可以看到之前在工作空间已被导入的 Android 工程。

2）依次单击 Eclipse 菜单栏里的“File”、“New”、“Project…”菜单项，如图 1-8 所示。

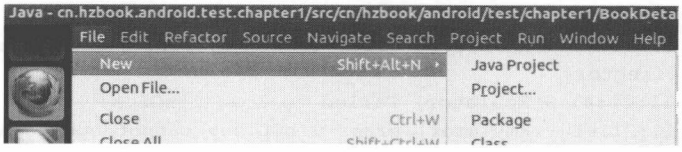


图 1-8 在 Eclipse 中新建工程（1）

3）在弹出的“New Project”对话框中，展开“Android”列表项，并选择“Android Test Project”项来指明要创建一个 Android 自动化测试工程，然后单击“Next”按钮，如