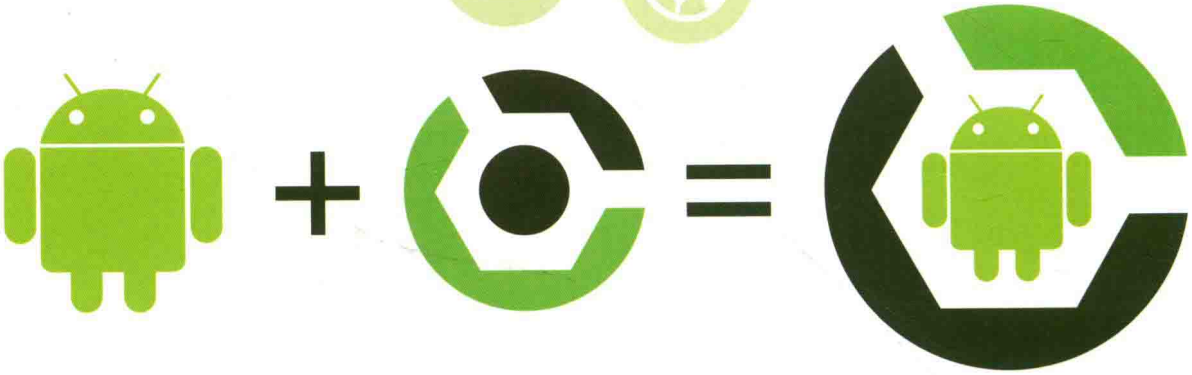
异步图书
www.epubit.com.cn

一本关于 Android Gradle 的权威书籍
基于新的 Android Gradle

Android Gradle 权威指南

飞雪无情◎编著

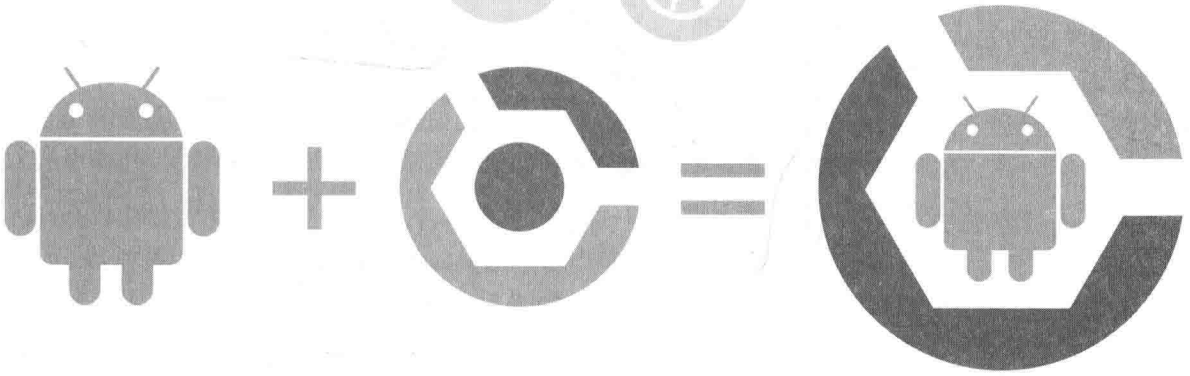


中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

一本关于 Android Gradle 的权威书籍
基于新的 Android Gradle

Android Gradle 权威指南

飞雪无情◎编著



人民邮电出版社
北京

图书在版编目（CIP）数据

Android Gradle权威指南 / 飞雪无情编著. -- 北京：人民邮电出版社，2017.9
ISBN 978-7-115-46123-0

Ⅰ. ①A… Ⅱ. ①飞… Ⅲ. ①移动终端－应用程序－程序设计－指南 Ⅳ. ①TN929.53-62

中国版本图书馆CIP数据核字(2017)第165955号

内 容 提 要

本书全面讲解了 Android 下 Gradle 的详细用法，并结合实例，让读者达到学以致用的目的。本书主要内容如下：

第 1 章 Gradle 入门，讲解了配置 Gradle 环境、Gradle Wrapper、Gradle 命令行；第 2 章 Groovy 基础，讲解了字符串、闭包等；第 3 章讲解了 Gradle 构建脚本基础；第 4 章为 Gradle 任务；第 5 章 Gradle 插件；第 6 章 Java Gradle 插件；第 7 章 Android Gradle 插件；第 8 章自定义 Android Gradle 工程；第 9 章 Android Gradle 高级自定义；第 10 章 Android Gradle 多项目构建；第 11 章 Android Gradle 多渠道构建；第 12 章 Android Gradle 测试；第 13 章 Android Gradle NDK 支持；第 14 章 Android Gradle 持续集成等核心开发知识。

本书讲解通俗易懂，适合 Android 程序员阅读，也适合作为大专院校相关专业师生的学习用书和培训学校的教材。

◆ 编　　著　飞雪无情
责任编辑　张　涛
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：15
字数：316 千字　　2017 年 9 月第 1 版
印数：1－2 000 册　　2017 年 9 月北京第 1 次印刷

定价：59.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前　言

背景

我 2010 年开始从事 Android 开发，是接触 Android 最早的那一批程序员，到现在也算是一个老兵了。最近几年 Android 很火，2013 年，Android 团队开始做 Android Studio 这个 IDE，想替换掉 ADT。Android Studio 是基于 Idea 开发的，它比 Eclipse 好用很多，而且又配合 Gradle 这个强大的构建工具，灵活，多工程构建方便，和 Maven 完美结合，比基于 Eclipse 的 ADT 强太多了，所以我就一直在关注 Android Gradle 的开发。

2014 年年底，Android Studio 发布 1.0 正式版，我就带领公司的整个 Android 开发团队，逐步完成了从 Eclipse ADT 到 Android Studio 的迁移。整个迁移过程中，遇到了很多问题，都慢慢地逐一解决，然后有时间的时候我就把遇到的这些问题总结到博客上，后来就接触到了很多用 Android Studio 开发的朋友，都是从 ADT 转过来的。从交流中我发现，大家对 Android Gradle 这种构建方式并不是很了解，都是很简单地会使用，如果真要遇到了什么问题，自己还是没有解决问题的能力，这主要是因为他们对 Gradle 这个构建工具以及 Android Gradle 这个构建插件不熟悉。

起因

仔细想想，其实不熟悉也属于正常，因为做 Android 研发的程序员，Android 的很多特性需要学习，除此之外，还要学习 SQLite 数据库、设计模式、业务等，哪有精力再去学 Android Gradle 构建呢，只是想：网上有现成的程序，抄过来，能运行通过不就好了吗！但是我要说，这是不对的。当问题你能解决而别人不能的时候，就是你“鹤立鸡群”的开始！

我以前做过 J2EE，学过 Groovy，接触过 Gradle，所以这是我的优势；在 ADT 迁移到 Android Studio 的时候又遇到了 Android Gradle 的很多“坑”，并且解决了，积累了经验，那么我可以把这些记录下来，让后来的所有人都可以查阅，以帮助大家解决项目中遇到的问题。一开始是记录在我的博客上，想到什么就写什么，但是发现没有系统性，很琐碎，不便于系统学习和了解 Android Gradle，所以就有了写书的想法，想通过由浅入深的介绍，融合我在项目中积累的经验，帮助大家更好地了解 Android Gradle，提高工作效率。

关于本书

本书是一本由浅入深讲解 Android Gradle 的书，本书将对 Gradle 基础、Groovy 基础、Gradle 插件、Android Gradle 构建、基于 Android Gradle 的单元测试和持续集成等做循序渐进的讲解，并且在讲解的过程中融入我在项目中遇到的问题、解决问题的思路以及方法。通过本书，你可以入门，并且深入了解 Gradle 以及 Android Gradle 构建，并以此为基础，深入 Android Gradle 相关知识点和使用技巧，让你在工作中事半功倍。

写作路线

本书分为 14 章，第 1～5 章介绍 Gradle、Groovy、Gradle Task、Gradle 插件等相关知识；第 6～10 章介绍 Android Gradle 的入门、构建、发布等相关知识；第 11～14 章则介绍基于 Android Gradle 的高级功能、单元测试以及持续集成等。

该路线也基本上是我在学习和研究的过程中总结规划的一条路线。因为我在学习的过程中，有的时候看到程序中的一种写法或者一个表示方式，不知道为什么这样用，当时只能记住，等到后面看到相关的介绍或者说明的时候才恍然大悟。但是这种方式不好，不易于理解和掌握 Android Gradle，所以贯穿本书的一点就是，所有知识都是先介绍、讲解说明，然后才会讲使用，让大家知其然并知其所以然。

代码约定以及下载

本书所有的 Gradle 脚本和 Android 代码都会遵循 Gradle 和 Android 规范，并且托管在 Github 上，以章节分目录和模块，便于查找。代码会随着书的更新而更新，下载地址如下：

Github 地址 https://github.com/rujews/android-gradle-book-code，大家可以 star 或者 fork。

代码开发环境

我一直提倡开发环境一定要是 Linux，不要使用 Windows，特别是做 Java、Android 开发，比如整个 Android 的源代码就不能在 Windows 系统下构建。Android 也是基于 Linux 平台的，只有使用它，才能更好地了解它，进而对你理解整个 Android 等有很大的帮助，而且在 Linux 系统下开发，其中提供的各种脚本、工具也能让你事半功倍。我从大学开始接触 Ubuntu 8.04，之后一直用它，2011 年年底开始带领公司整个 Android 团队把操作系统换成 Ubuntu，并一直使用到现在。

我在写作的过程中使用到的操作系统、SDK 工具包、IDE 等如下。

（1）操作系统：Ubuntu 16.04 发行版。

此为试读，需要完整PDF请访问：www.ertongbook.com

（2）JDK：OpenJDK 1.8.0。

（3）Gradle：Gradle 2.14.1 All 版。

（4）IDE：Android Studio 2.2.3。

（5）Android Plugin:Android Gradle 2.2.3。

（6）Android：API 23。

以上开发环境可能会在我写作的过程中更新，因为这些工具可能会有新版发布，所以我尽可能用最新版，尤其是 Android Gradle 这个插件的版本，这样我就能及时介绍它新发布的功能特性。

联系作者

遵循着为读者负责及用心写好书的原则，我会尽可能把 Android Gradle 的知识和我在项目中的经验写出来。本书可能会有不对的地方，希望大家能留言指正。如果觉得写得好，请不要吝啬给五星好评哦，你的鼓励就是对我最大的支持。下面是我的联系方式。

博客：http://www.flysnow.org/。

微信公众号：flysnow_org。

作者

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第1章 Gradle入门

Gradle是一款非常优秀的构建系统工具，它的DSL基于Groovy实现，可以让你很方便地通过代码控制这些DSL来达到你构建的目的。Gradle构建的大部分功能都是通过插件的方式来实现，所以非常灵活方便，如果内置插件不能满足你的需求你可以自定义自己的插件。

本章我们就介绍Gradle的入门知识，在介绍之前，我们先假定读者已经具备以下知识。

（1）了解并且会使用Java，精通最好。

（2）会独立搭建Java开发环境。

（3）最好会使用Linux操作系统，比如Ubuntu。

为什么会有这样的假定呢？因为这本书是介绍Android Gradle开发构建的书，所以不会讲Java的基本知识。希望读者会用Linux操作系统的原因，是因为本书的所有脚本、代码、IDE等都是基于Ubuntu完成的，当然比如涉及Gradle安装还会介绍一下Windows的安装步骤，但是不会太多涉及Windows的东西，所以还是希望读者在阅读本书前已经掌握了这些知识。

1.1 配置Gradle环境

安装之前确保已经安装配置好Java环境，要求JDK 6以上，并且在环境变量里配置了JAVA_HOME，查看Java版本可以在终端输入如下命令：

```
java -version
```

我这里使用的是open jdk 1.8.0_91：

```
➜  ~ java -version
openjdk version "1.8.0_91"
OpenJDK Runtime Environment (build 1.8.0_91-8u91-b14-3ubuntu1~16.04.1-b14)
OpenJDK 64-Bit Server VM (build 25.91-b14, mixed mode)
```

1.1.1　Linux 下搭建 Gradle 构建环境

这里以 Ubuntu 16.04 发行版为例介绍如何在 Linux 下搭建 Gradle 构建环境，其他诸如 CentOS 大同小异，参考一下就可以了。

我们这里以 Gradle 2.14.1 版本为准进行介绍。先到 Gradle 官网 https://gradle.org/ 下载好 Gradle SDK，直接下载地址为 https://downloads.gradle.org/distributions/gradle-2.14.1-all.zip。我们下载的是 all 版本，也就是说，里面包含了 Gradle SDK 所有相关的内容，包括源代码、文档、示例等。如果因为网络问题下载不了，可以使用镜像下载，镜像首页为 http://mirrors.flysnow.org/，该 Gradle 版本下载地址为 http://mirrors.flysnow.org/gradle/gradle-2.14.1-all.zip，下载之后进行解压，我们可以得到如下目录清单。

① docs：API、DSL、指南等文档。

② getting-started.html：入门链接。

③ init.d：gradle 的初始化脚本目录。

④ lib：相关库。

⑤ LICENSE。

⑥ media：一些 icon 资源。

⑦ NOTICE。

⑧ samples：示例。

⑨ src：源文件。

要运行 Gradle，必须把 GRADLE_HOME/bin 目录添加到你的环境变量 PATH 的路径里才可以。在 Linux 下，如果你只想为当前登录的用户配置可以运行 Gradle，那么可以编辑～/.bashrc 文件添加以下内容：

```
#这里是作者的 Gradle 目录，要换成你自己的
GRADLE_HOME=/home/flysnow/frame/gradle
```

```
PATH=${PATH}:${GRADLE_HOME}/bin
Export GRADLE_HOME PATH
```

上面 GRADLE_HOME 是我的 Gradle 解压后的目录，这里要换成你自己的。以上添加后保存，然后在终端输入 source ~/.bashrc，回车执行让刚刚的配置生效。

如果你想让所有用户都可以使用 Gradle，那么你就需要在/etc/profile 中添加以上内容，在这里添加后，对所有用户都生效，这种方式的添加，必须要重启计算机才可以。

好了，现在我们已经配置好了，要验证我们的配置是否正确，是否可以运行 Gradle，我们只需要打开终端，输入 gradle -v 命令查看即可。如果能正确显示 Gradle 版本号、Groovy 版本号、JVM 等相关信息，那么说明你已经配置成功了。这里以验证我的配置为例：

```
➜  ~ gradle -v
------------------------------------------------------------
Gradle 2.14.1
------------------------------------------------------------

Build time:   2016-10-20 03:46:36 UTC
Build number: none
Revision:     b463d7980c40d44c4657dc80025275b84a29e31f

Groovy:       2.4.4
Ant:          Apache Ant(TM) version 1.9.3 compiled on December 23 2013
JVM:          1.8.0_91 (Oracle Corporation 25.91-b14)
OS:           Linux 4.4.0-38-generic amd64
```

1.1.2 Windows 下搭建 Gradle 构建环境

Windows 下搭建 Gradle 环境和 Linux 非常相似，只不过方式不同。我们通过右击我的电脑，打开属性面板，然后找到环境变量配置项，添加 GRADLE_HOME 环境变量，然后把 GRADLE_HOME\bin 添加到 PATH 系统变量里保存即可。完成后打开 CMD，运行 gradle -v 来进行验证，整体效果和 Linux 差不多，这里就不再一一详述。

1.2 Gradle 版 Hello World

环境搭建好了，那么我们就开始写一个 Hello World 版的 Gradle 脚本。

新建好一个目录，我这里是 android-gradle-book-code，然后在该目录下创建一个名为 build.gradle 的文件，打开编辑该文件，输入以下内容：

```
task hello{
    doLast{
        println'Hello World!'
    }
}
```

打开终端，然后移动到 android-gradle-book-code 下，使用 gradle -q hello 命令来执行构建脚本：

```
$ gradle -q hello
Hello World!
```

好了，如愿以偿地打印出来我们想要的结果，下面我们一步步分析结果产生的步骤和原因。build.gradle 是 Gradle 默认的构建脚本文件，执行 Gradle 命令的时候，会默认加载当前目录下的 build.gradle 脚本文件。熟悉 Ant 的读者感觉和 build.xml 差不多，当然你也可以通过 -b 参数指定想要加载执行的文件。

这个构建脚本定义一个任务（Task），任务名字叫 hello，并且给任务 hello 添加了一个动作，官方名字是 Action，阅读 Gradle 源代码你会到处见到它，其实它就是一段 Groovy 语言实现的闭包。在这里我觉得叫业务代码逻辑或者回调实现更贴切一些，因为 doLast 就意味着在 Task 执行完毕之后要回调 doLast 的这部分闭包的代码实现。

熟悉 Ant 的读者，会觉得任务（Task）和 Ant 里的 Target（目标）非常相似。其实没错，现在可以认为它们基本上相同。

再看 gradle -q hello 这段运行命令，意思是要执行 build.gradle 脚本中定义的名为 hello 的 Task，-q 参数用于控制 gradle 输出的日志级别，以及哪些日志可以输出被看到。

看到 println 'Hello World!'了吗，它会输出 Hello World!，通过名字相信大家已经猜出来了，它其实就是 System.out.println("Hello World!")的简写方式。Gradle 可以识别它，是因为 Groovy 已经把 println()这个方法添加到 java.lang.Object，而在 Groovy 中，方法的调用可以省略签名中的括号，以一个空格分开即可，所以就有了上面的写法。还有一点要说明的就是，在 Groovy 中，单引号和双引号所包含的内容都是字符串；不像 Java 中，单引号是字符，双引号才是字符串。

1.3 Gradle Wrapper

Wrapper，顾名思义，其实就是对 Gradle 的一层包装，便于在团队开发过程中统一 Gradle 构建的版本，这样大家都可以使用统一的 Gradle 版本进行构建，避免因为 Gradle 版本不统一带来的不必要的问题。

在这里特别介绍的目的是因为，我们在项目开发过程中，用的都是 Wrapper 这种方式，而不是我们在 1.1 节里介绍的自己下载 ZIP 压缩包，配置 Gradle 的环境的方式。Wrapper 在 Windows 下是一个批处理脚本，在 Linux 下是一个 shell 脚本。当你使用 Wrapper 启动 Gradle 的时候，Wrapper 会检查 Gradle 有没有被下载关联，如果没有将会从配置的地址（一般是 Gradle 官方库）进行下载并运行构建。这对我们每个开发人员是非常方便的，因为你不用去专门配置环境了，只要执行 Wrapper 命令，它会帮你搞定一切。这种方式也方便我们在服务器上做持续集成（CI），因为我们不用在服务器上配置 Gradle 环境。

1.3.1 生成 Wrapper

Gradle 提供了内置的 Wrapper task 帮助我们自动生成 Wrapper 所需的目录文件，在一个项目的根目录中输入 gradle wrapper 即可生成：

```
$ gradle wrapper
:wrapper

BUILD SUCCESSFUL

Total time: 2.804 secs

This build could be faster, please consider using the Gradle Daemon: http://gradle.
org/docs/2.14.1/userguide/gradle_daemon.html
```

生成的文件如下：

```
├─gradle
│   └─wrapper
│       ├─gradle-wrapper.jar
│       └─gradle-wrapper.properties
├─gradlew
└─gradlew.bat
```


此为试读，需要完整PDF请访问：www.ertongbook.com