

Gradle入门系列（4）：创建二进制发布版本

2015-07-07 安卓应用频道

在创建了一个实用的应用程序之后，我们可能想将其与他人分享。其中一种方式就是创建一个可以从网站上下载的二进制文件。

这篇教程描述了如何创建一个二进制发布版本，满足以下需求：

- 二进制发布一定不能使用所谓的“fat jar”方式。换句话说，我们应用程序中的所有依赖一定不能被打包到该程序相同的jar包中。
- 二进制发布必须包含针对*nix和Windows操作系统的启动副本。
- 二进制发布的根目录必须包含许可证。

我们开始吧。

创建二进制发布文件

Application插件是一种Gradle插件，让我们可以运行、安装应用程序并用非“fat jar”方式创建二进制发布版本。

还记得我们在上篇教程中提到的一个例子吗？在它的build.gradle文件中做一些相应的更改，就可以进行二进制发布了。

1. 移除jar任务的配置。
2. 为项目应用application插件。
3. 对应用程序的主类进行配置，设置mainClassName属性。

在build.gradle文件中作出以上更改后，结果如下（相关部分已经高亮）：

```
apply plugin: 'application'
```

```
apply plugin: 'java'
```

```
repositories {
```

```
    mavenCentral()
```

```
}
```

```
dependencies {
```

```
compile 'log4j:log4j:1.2.17'  
testCompile 'junit:junit:4.11'  
}  
  
mainClassName = 'net.petrikainulainen.gradle.HelloWorld'
```

Application插件在项目中添加了5个任务：

- run任务用以启动应用程序。
- startScripts任务会在build/scripts目录中创建启动脚本，这个任务所创建的启动脚本适用于Windows和*nix操作系统。
- installApp任务会在build/install/[project name]目录中安装应用程序。
- distZip任务用以创建二进制发布并将其打包为一个zip文件。可以在build/distributions目录下找到。
- distTar任务用以创建二进制发布并将其打包为一个tar文件。可以在build/distributions目录下找到。

我们可以通过在项目根目录下运行以下命令：gradle distZip或gradle distTar 创建二进制文件。假设我们创建了一个打包为zip文件的二进制文件，输出如下：

```
> gradle distZip  
:compileJava  
:processResources  
:classes  
:jar  
:startScripts  
:distZip
```

BUILD SUCCESSFUL

Total time: 4.679 secs

如果将application插件创建的二进制文件解压缩，可以得到以下目录结构：

- bin目录：包括启动脚本。
- lib目录：包括应用程序的jar文件以及它的依赖。

你可以阅读Gradle Application插件用户指南（第45章）了解更多关于Application插件信息。

现在，我们可以创建一个几乎能满足所有需求的二进制发布了。不过，我们仍然需要在我们二进制发布的根目录下添加应用程序的许可证。下面我就来看一下，如何做到这一点。

在二进制发布版本中添加应用程序许可证

我们可以通过以下步骤，在二进制发布中添加应用程序许可证：

1. 创建一个任务，将许可证从项目的根目录复制到build目录下。
2. 将许可证加入到所创建的二进制发布的根目录下。

我们来仔细看一下这些步骤的详情。

将许可证文件复制到build目录下

LICENSE文件包含了我们应用程序的许可信息，可以在项目的根目录下找到它。

可以通过以下步骤将许可证文件复制到build目录下：

1. 创建一个新的Copy任务，名为copyLicense。
2. 使用CopySpec接口中的from()方法配置源文件，将“LICENSE”作为参数调用。
3. 使用CopySpec接口中into()方法配置target目录，将\$buildDir属性作为参数调用。

在完成这些步骤以后，build.gradle文件如下（相关部分已高亮）：

```
apply plugin: 'application'
apply plugin: 'java'

repositories {
    mavenCentral()
}

dependencies {
    compile 'log4j:log4j:1.2.17'
    testCompile 'junit:junit:4.11'
}

mainClassName = 'net.petrikainulainen.gradle.HelloWorld'

task copyLicense(type: Copy) {
```

```
from "LICENSE"
into "$buildDir"
}
```

现在，我们已经创建了一个任务，将LICENSE文件从项目的根目录复制到build目录下。然而，当我们在项目根目录下运行命令：gradle distZip，会看到以下输出：

```
> gradle distZip
:compileJava
:processResources
:classes
:jar
:startScripts
:distZip
```

BUILD SUCCESSFUL

Total time: 4.679 secs

换句话说，我们新的任务还没有被引入。所以理所当然的，二进制发布中也没有包含许可证。让我们来修复这个问题。

将许可证文件加入到二进制发布文件中

我们可以通过以下步骤将许可证文件加入到二进制发布文件中：

1. 将copyLicense任务从一个Copy任务改为正常的Gradle任务，只需在它的声明中移除“(type: Copy)”字符串。
2. 按照以下步骤修改copyLicense任务
3. 配置copyLicense任务输出。创建一个新的文件对象，指向build目录的许可证文件，并将其设置为outputs.file属性值。
4. 将许可证文件从项目的根目录复制到build目录下。
5. Application插件在项目中设置了一个CopySpec属性，名为applicationDistribution。我们可以使用这个属性在已创建的二进制文件中加入许可证文件，步骤如下：
6. 使用CopySpec接口中的from()方法配置许可证文件的位置，将copyLicense任务的输出作为方法参数。
7. 使用CopySpec接口中into()方法配置target目录，将一个空的字符串作为参数调用方法。

在实现了这些步骤以后，build.gradle文件如下（相关部分已高亮）：

```
apply plugin: 'application'
apply plugin: 'java'

repositories {
    mavenCentral()
}

dependencies {
    compile 'log4j:log4j:1.2.17'
    testCompile 'junit:junit:4.11'
}

mainClassName = 'net.petrikainulainen.gradle.HelloWorld'

task copyLicense {
    outputs.file new File("$buildDir/LICENSE")
    doLast {
        copy {
            from "LICENSE"
            into "$buildDir"
        }
    }
}

applicationDistribution.from(copyLicense) {
    into ""
}
```

在项目根目录下运行命令gradle distZip时，会看到以下输出：

```
> gradle distZip
:copyLicense
:compileJava
:processResources
:classes
:jar
:startScripts
```

```
:distZip
```

BUILD SUCCESSFUL

Total time: 5.594 secs

可以看到，copyLicense任务现在已经被引入了。我们可以对二进制文件解压缩，在根目录下就能发现LICENSE文件了。

最后，我们对这篇教程中所学到的东西进行总结。

总结

这篇教程为我们教授了三方面内容：

- 学会使用application插件创建一个二进制发布。
- 学会使用Copy任务将一个文件从源目录复制到目标目录。
- 学会如果将文件加入到由Application插件创建的二进制发布文件中。

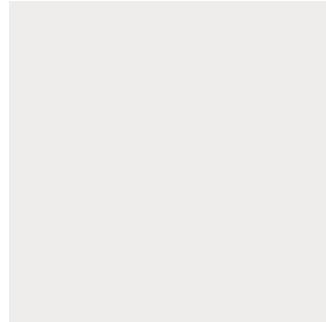
如果你想运行本教程中的示例程序，请从Github中获取。

来源：[伯乐在线 - Justin Wu](#)

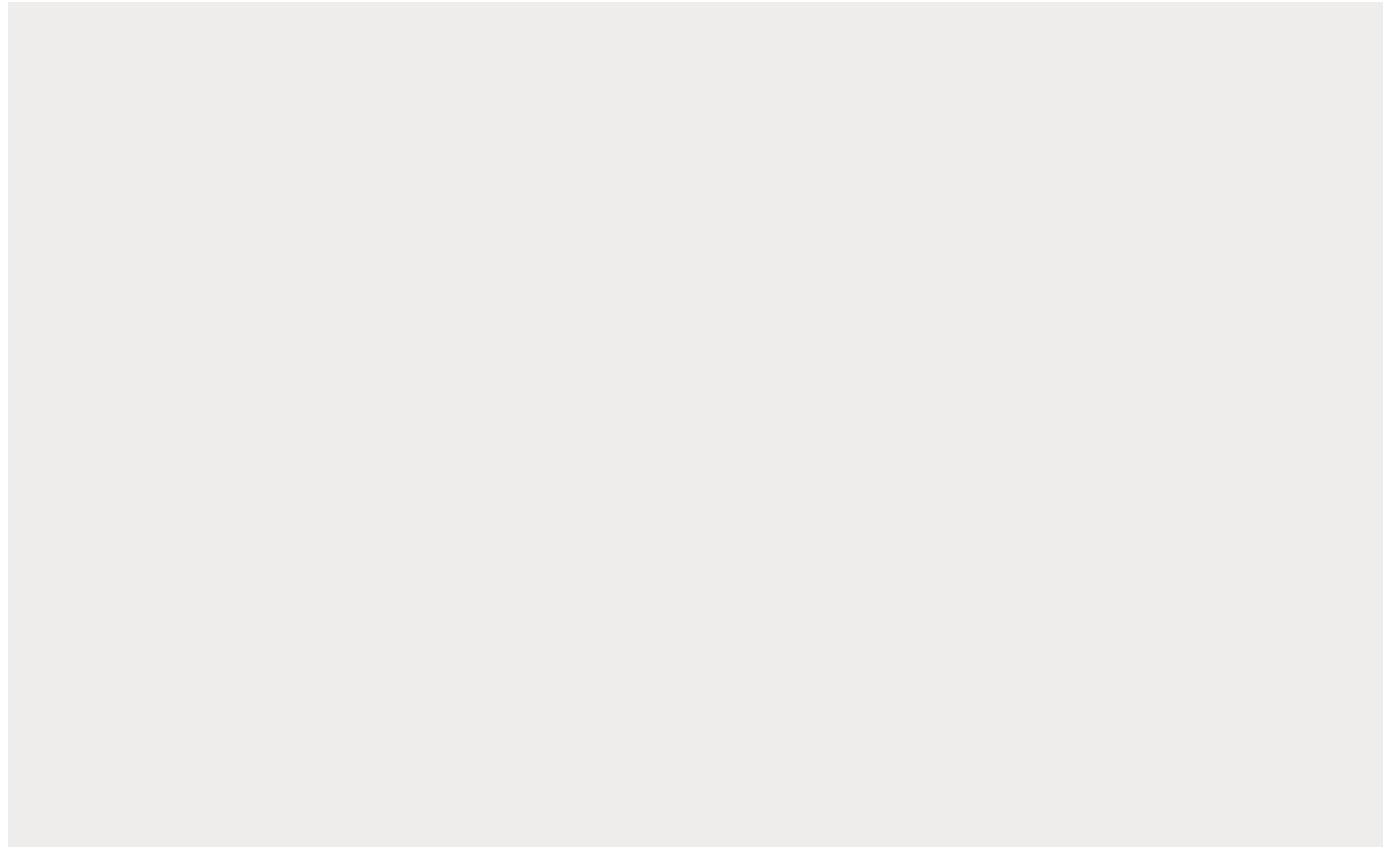
链接：<http://android.jobbole.com/80340/>

「安卓应用频道」专注 Android 技术、设计和市场推广相关的内容分享。如果你期望了解 Android 应用的全流程知识，欢迎关注我们。

微信号：AndroidPD



（长按上图↑可自动识别二维码）



[阅读原文](#)



微信扫一扫
关注该公众号