

# 是时候迁移至 AndroidX 了！



谷歌开发者



已认证的官方帐号

已关注

Talk8 等 53 人赞同了该文章

长久以来，我们致力于做到三件事：实践指南、减少模板代码和简化任务流程，我们希望帮助开发者们集中精力专注在真正需要考虑的逻辑中去。Jetpack 为此而生，它所包含的库、工具和指南，可以帮助您更轻松地编写高质量的应用。

Jetpack 和 AndroidX 有什么关系呢？Jetpack 中所有库都使用 AndroidX 作为包名，我们把 AndroidX 作为一个开发、测试和发布 Jetpack 库的开源工程。

在 2018 年的 I/O 大会上我们宣布了把 Support Library 重构至 AndroidX 命名空间的计划。在 Support Library 28，我们完成了重构，并且发布了 AndroidX 1.0。为了能够享受 Jetpack 所带来的便利，您需要将旧的 Support Library 迁移至 AndroidX。

## 为什么有必要迁移至 AndroidX

您可能会想：既然 AndroidX 只是 Support Library 28 的重构，那为什么要迁移呢？关于这个问题，我们有下面几个理由：

1. Support Library 已经完成了它的历史使命，28 将会是它的最后发布版。我们接下来将不会继续在 Support Library 中修复 bug 或发布新功能；
2. 更好的包管理：独立版本、独立命名以及更高频率的更新。以上优点，AndroidX 开箱既得；
3. 目前已经有许多我们耳熟能详的工具库已经迁移至 AndroidX，例如 Google Play 服务、Firebase、Bufferknife、Mockito 2、SQL Delight，我们后面会提到如何迁移它们的依赖；
4. 我们正在努力推广 AndroidX 命名空间，未来所有新推出的组件库，例如 Jetpack Compose 和 CameraX，都将成为 AndroidX 的一员。

## 如何迁移至 AndroidX

### 前期准备

在开始迁移之前，为了使接下来的工作可以更加顺畅平滑，我们希望您可以做到以下几点：

- 首先，备份整个工程。大部分开发者都在使用代码版本控制系统，但由于迁移工作牵扯到大量的文件修改，建议您备份整个工程；
- 其次，我们希望您尽量减少同时进行的功能开发；
- 最后还是建议您：在单独的分支进行迁移工作。

### 开始迁移

在整个迁移步骤中，我们将着重于解决错误，使您的应用编译并通过所有测试。

下面是迁移工作的流程示意图，虽然步骤不少，但是本文会对其中的每一步都做出说明：



## 第一步: 将 Support Library 升级至 28

首先, 我们希望您把当前的 Support Library 依赖升级至版本 28。如果您从早期版本的 Support Library 进行迁移, 可能会在需要修改命名空间的同时遭遇 API 不兼容的问题; 而 Support Library 28 的 API 与 AndroidX 之间只有命名空间上的不同。所以我们建议, 先尝试将 Support Library 升级至版本 28, 处理过所有 API 变更, 并且确保编译通过后, 再进行下一步, 这样所做的修改是最少的。

## 第二步: 开启 Jetifier

接下来需要做的是开启 Jetifier。Jetifier 可以帮助您迁移第三方依赖库的依赖至 AndroidX。正如字面意思所说, Jetifier 会修改这些第三方依赖库的代码, 从而使其与使用 AndroidX 的工程兼容。不过 Jetifier 不会修改您的源码和自动生成的代码, 因此不用担心它会造成额外的不良影响。

开启 Jetifier 十分的简单, 您只需要在 gradle.properties 文件中加入 "android.useAndroidX = true" 和 "android.enableJetifier = true" 即可。"useAndroidX" 设置用于开启 AndroidX 库的自动导入, 当您自动补全或导入依赖库时, 会自动导入 AndroidX 库。

## 第三步: 检查第三方库版本的兼容性

当您开启 Jetifier 之后, 就要着手升级第三方依赖库到兼容的版本。在您真的开始迁移之前, 最好把所有依赖升级到最新。

为什么要这么做? 其实我们自己就在这方面 "栽过跟头", 我们有一个演示应用: Plaid, 它依赖了图片加载库 Glide, 我们本来打算使用 Plaid 来演示如何迁移应用至 AndroidX, 但当我们没有检查 Glide 依赖库版本就开始迁移时, 我们遭遇了一堆编译错误。检查后才发现, 当时依赖的那个版本的 Glide 无法兼容 AndroidX。

而当我们把 Glide 和其他依赖库版本都升级后, 再做迁移工作, 就没有再出现相同的错误。所以, 建议在开始迁移前, 先检查和升级应用的第三方依赖, 新版本的第三方库可能已经兼容 AndroidX。由于 Jetifier 不会帮您迁移自动生成代码的依赖库, 所以您还是需要自己检查这类依赖是否兼容 AndroidX。

如果跳过了前面两步, 您可能会遇到一些问题:

- 如果您当前使用的第三方库不兼容 AndroidX, 您将会看到它依然在尝试拉取旧版本的 Support Library;
- 而如果您的工程被部分迁移, 可能还会遇到类型重复的错误, 这是因为工程正在尝试从 Support Library 和 AndroidX 拉取相同的代码。



#### 第四步: 将 Support 库依赖转换为 AndroidX

这一步开始前, 您应该完成了前面三个步骤: 升级 Support Library 到 28 版; 开启 Jetifier; 升级和检查第三方依赖库。确定这些都没问题后, 我们终于可以开始真正的迁移工作了。这一步有以下三个方法供您参考:

##### 1. 使用 Android studio 自动迁移工具

我们在 Android 3.2 稳定版中加入了 "Migrate to AndroidX" 选项, 方便大家迁移。您可以在 "Refactor" 菜单中找到 "Migrate to AndroidX" 选项:



这个按钮的功能，就是迁移源码中的依赖到 AndroidX，理想情况下，它会帮您完成绝大部分工作。

## 2. 使用自动迁移脚本

我们也意识到有些团队使用的不是 Android Studio，而且也会有一些应用的结构过于复杂，使我们的工具无法生效。

所以还有两种选择，其中之一便是使用 bash 脚本中的 grep 和 sed 命令。在介绍如何使用脚本进行迁移之前，我们要特别感谢 Dan Lew 为我们提供了这个工具。

您可以通过短链接: [goo.gle/androidx-migration-script](https://goo.gle/androidx-migration-script) 去到脚本源码的 GitHub 页面，在那里您也可以找到更多的社区贡献内容。

脚本的工作原理并不复杂，如下所示，您需要手动做的是配置好类型映射表 "androidx-class-mapping.csv" 和工程路径地址，而脚本中真正有效的部分，就只是 grep 命令后跟着一个 sed 命令来替换工程中导入的包名：

由于脚本的处理十分简单粗暴，所以可能会在某些情况下造成一些错误。使用这种方式一定要自己心里有数。

### 3. 人工迁移

另一个选择，是人工进行迁移工作。在迁移到 AndroidX中，您能看到前文提到过的 Support Library 与 AndroidX 的类型映射关系表。如下图，有了这个映射关系表，您就可以根据具体情况进行替换：

这一步做完之后，只要您重新编译工程，并且修复那些迁移工作中损坏的测试，就可以获得一个基于 AndroidX 的工程。可喜可贺！

## 可能遇到的问题

当然，真实的情况往往不会那么一帆风顺。下面我们收集了一些迁移过程中常见的问题，希望能帮到您。

### 常见的需要手动处理的情况

以下图为例，我们看到这里依赖的仍然是 Support Library，其中 drawerLayout 和 recyclerview 的版本是用一组变量设置的：



遇到这种情况时，自动迁移不会理会您之前的变量配置，它会直接把这些库替换成一个确定的 AndroidX 版，如果您仍然想要使用变量管理这些库的版本号，就需要手动把 AndroidX 的依赖库版本改为使用变量设置。

自动迁移工具也不会修改您的混淆文件和构建脚本。如果这些文件中包含相关的包名，您需要手动去把它们改好。

## 冲突处理

我们前面有提到，一定要在一个新的分支中处理迁移工作，关于这点还有一些和大家分享的内容。

由于迁移工作会修改大量的文件，所以我们建议减缓或停止手头的开发工作。虽然要求整个开发团队停工听起来十分离谱，但是这样确实可以大大减少可能产生的合并冲突。

退而求其次的话，如果条件允许，最好能安排一些人手在一个单独的分支上专注于迁移的工作。与此同时，也要向团队中的其他成员预警即将到来的合并冲突。

在迁移依赖时，要专注于错误的修改，以编译成功和通过所有测试为首要目标。不要在迁移的同时进行重构或者引入新的功能。

## 检查自动迁移工具导入的库版本

当您运行完自动迁移功能后，您可能会发现新的依赖库中既有稳定版，又有 Alpha 版。这其实取决于我们最新发布的版本。您需要手动修改这些依赖库的版本，以满足自己工程的特定需要。

## 文档资源

我们总结了一些与本文相关的文档放在最后，来方便您回顾和查找。

[AndroidX 概览](#)包括: AndroidX 总览、迁移指南以及 Support Library 到 AndroidX 库稳定版和 Alpha 版的映射关系表。如果您想要使用脚本处理迁移，这里也提供映射关系表的 CSV 文件。

我们有一篇文章介绍 Kotlin & Jetpack 最佳实践技巧: [把 "格子衫" 改造得更时尚](#)，描述了示例工程 Plaid 迁移至 AndroidX 的过程。在这篇文章中，我们说明了迁移的步骤，遇到的问题和对应的解决方案。

我们还提供了[问题追踪页](#)，您可以在这个页面看到我们正在解决的问题，也可以通过左上角的按钮建立新的问题给我们。

祝大家都能顺畅地迁移至 AndroidX!



[点击这里查看 Android 官方中文文档](#) —— AndroidX 概览

发布于 04-26

[Android](#) [Android 开发](#)

推荐阅读



[\[再见\]SharedPreferences 拥抱...](#)

ii-dhl

关于Android编译，你需要了解什么

本文来自张绍文老师的《Android开发高手课》，我把我认为比较好的文章整理分享给大家。作为一名Android 工程师，我们每天都会经历无数次编译。对于小项目来说，半分钟或者1，2分钟即可编...

xiang... 发表于xiang...



[开源整理：有趣的Android动画交互设计](#)

D-clo... 发表于技术视界



[Android 手机掉帧、卡顿、续航变短的锅应该谁来背？](#)

转角一只喵 发表于Andro...

20 条评论

[切换为时间排序](#)

写下你的评论...



举个例子

04-26

[惊喜]希望有个插件，能更换各种三方库的依赖。

👍 2



Tolcol 回复 举个例子

04-28

enableJetifier打开，会给你改第三方的依赖的

👍 1

▲ 赞同 53 ▼

💬 20 条评论

🔗 分享

♥ 喜欢

★ 收藏

📄 申请转载

...

谢谢[大笑]

👍 1



qastscjyv

04-26

道理我都懂 但是G00g1e不是倒闭了吗¿

👍 13



图门月朗

04-26

只能新项目用，老项目不敢迁

👍 1



高渐离

04-26

所以实在点，迁移有什么好处吗

👍 2



didikee 回复 高渐离

04-26

好处没啥，就是不迁移的话老的supports库不更新了，有bug也不修了[飘泪笑]

👍 1



柳HF 回复 didikee

04-27

哈哈哈[惊喜]

👍 1



资深CV工程师

04-26

在国内做android开发好难，翻墙越来越艰难，从生产工具到实际开发，步步维艰。往往了解到的信息都是N手后的。

👍 6



lury 回复 资深CV工程师

04-27

国外开发者擅长创造新知识，国内开发者擅长学习新知识。

👍 1



四月的少年

04-26

去年就已经只用 AndroidX 了

👍 1



peerless2012

04-26

已经用了快两年了。

👍 1



GavinNovate

04-27

老项目太艰难了，特别是有些三方库只支持support

👍 1



陈金栋

04-27

谷歌Android X是怎么回事呢？谷歌相信大家都很熟悉，但是谷歌Android X是怎么回事呢，下面就让小编带大家一起了解吧。

谷歌Android X，其实就是一种新的开源工程，大家可能会很惊讶谷歌怎么会Android X呢？但事实就是这样，小编也感到非常惊讶。

这就是关于谷歌Android X的事情了，大家有什么想法呢，欢迎在评论区告诉小编一起讨论哦！

👍 11



子非鱼 回复 陈金栋

05-03

哈哈哈哈哈 总结的不错

▲ 赞同 53 ▼

💬 20 条评论

➦ 分享

♥ 喜欢

★ 收藏

📄 申请转载

...





爱国

04-27



加油，我支持你们！

👍 1



白勺

04-27

谷歌不是倒闭了吗？官网都打不开[调皮]

👍 2



Noone

04-28

去年我已经把代码全部迁移到AndroidX，说实话Google 对代码迁移的支持真是太完美了，迁移过程当中虽然出现一些错误，但搜索一下都能找到答案。

👍 1



品小猴

05-09

自动迁移一点用没有，最后还是我自己手动一个个改

👍 1



Wildlily

06-17

问一下，我要写.mk文件用mm编译apk，那Androidx包怎样在.mk文件中引用呢？我在LOCAL\_STATIC\_ANDROID\_LIBRARISE中引入对应Androidx包后，编译会报错package R不存在错误！

👍 赞