

缩减应用大小

随着业务迭代，apk体积逐渐变大。项目中积累的无用资源，未压缩的图片资源等，都为apk带来了不必要的体积增加。而APK的大小会影响应用加载速度、使用的内存量以及消耗的电量。

了解 APK 结构

在讨论如何缩减应用的大小之前，有必要了解下应用 APK 的结构。APK 文件由一个 Zip 压缩文件组成，其中包含构成应用的所有文件。这些文件包括 Java 类文件、资源文件和包含已编译资源的文件。

APK 包含以下目录：

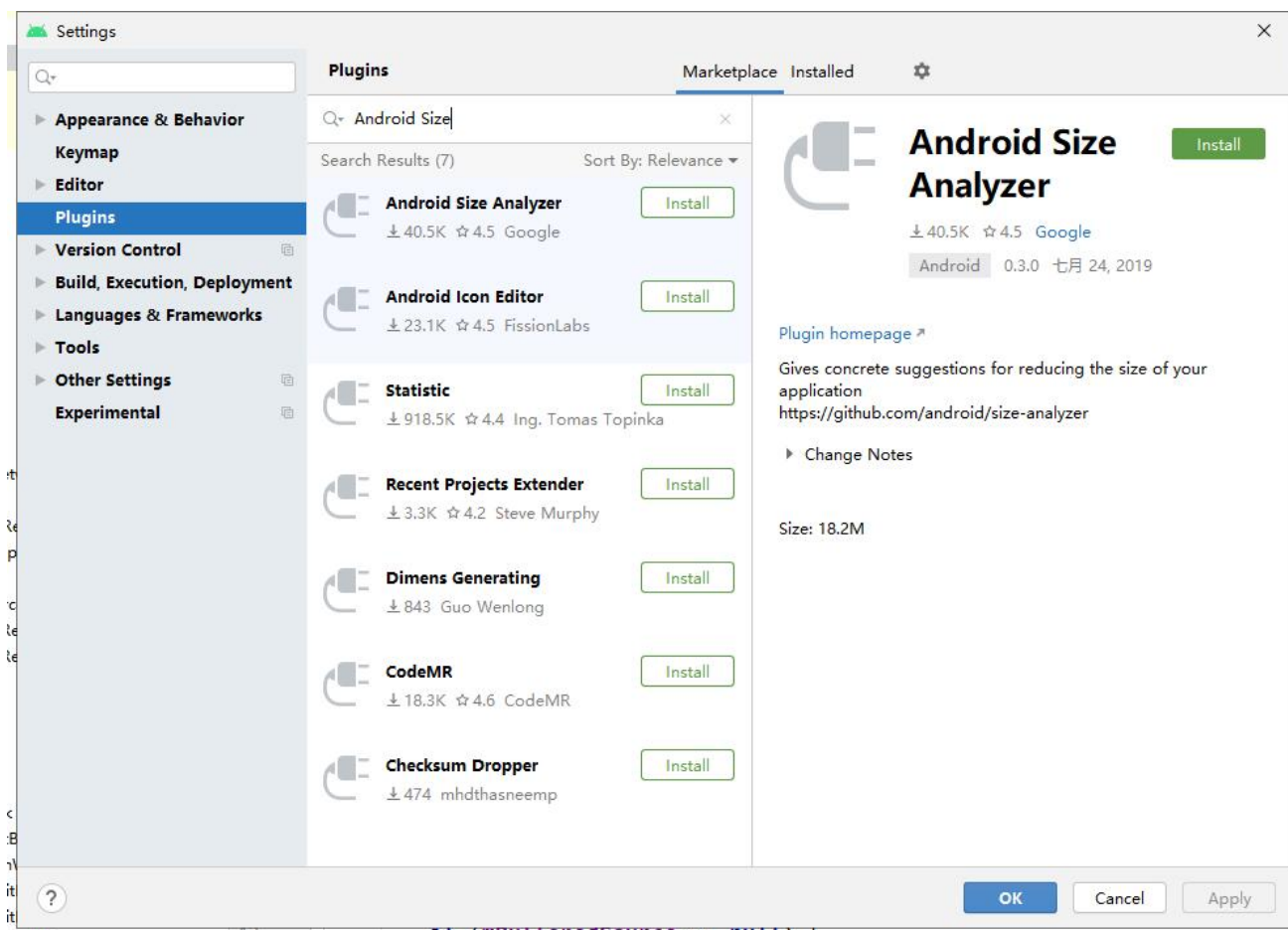
- `META-INF/`：包含 `CERT.SF` 和 `CERT.RSA` 签名文件，以及 `MANIFEST.MF` 清单文件。
- `assets/`：包含应用的资源；应用可以使用 `AssetManager` 对象检索这些资源。
- `res/`：包含未编译到 `resources.arsc` 中的资源（图片、音视频等）。
- `lib/`：包含特定于处理器软件层的已编译代码。此目录包含每种平台类型的子目录，如 `armeabi`、`armeabi-v7a`、`arm64-v8a`、`x86`、`x86_64` 和 `mips`。

APK 还包含以下文件。在这些文件中，只有 `AndroidManifest.xml` 是必需的。

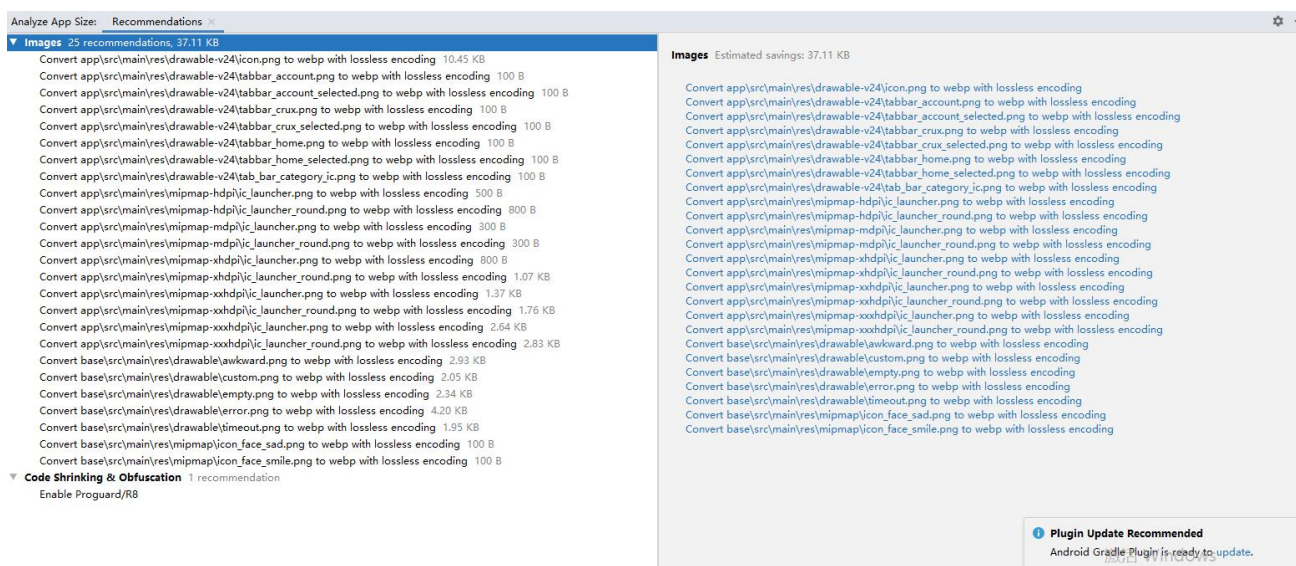
- `resources.arsc`：包含已编译的资源。此文件包含 `res/values/` 文件夹的所有配置中的 XML 内容。打包工具会提取此 XML 内容，将其编译为二进制文件形式，并压缩内容。此内容包括语言字符串和样式，以及未直接包含在 `resources.arsc` 文件中的内容（例如布局文件和图片）的路径。
- `classes.dex`：包含以 Dalvik/ART 虚拟机可理解的 DEX 文件格式编译的类。
- `AndroidManifest.xml`：包含核心 Android 清单文件。此文件列出了应用的名称、版本、访问权限和引用的库文件。该文件使用 Android 的二进制 XML 格式。

Android Size Analyzer

Android Size Analyzer 工具可轻松地发现和实施多种缩减应用大小的策略。



首先在 Android Studio 中的插件市场下载安装 Android Size Analyzer 插件。安装插件后，从菜单栏中依次选择 **Analyze > Analyze App Size**，对当前项目运行应用大小分析。分析了项目后，系统会显示一个工具窗口，其中包含有关如何缩减应用大小的建议。



移除未使用资源

APK瘦身关键就在一个字：删！没用的就删了。

启用资源缩减（不打包）

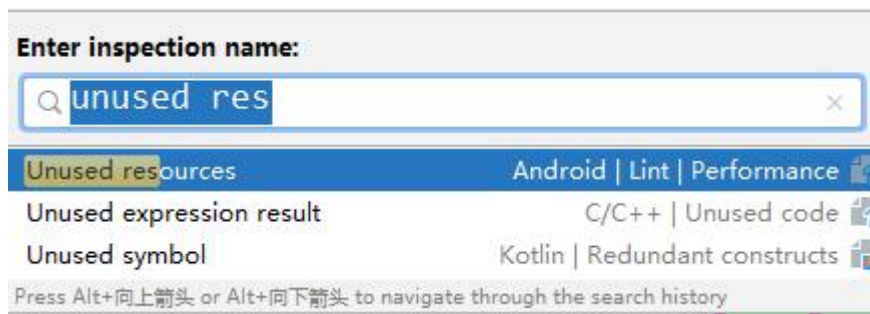
如果在应用的 `build.gradle` 文件中启用了资源缩减：`shrinkResources`，则 Gradle 在打包APK时可以自动忽略未使用资源。资源缩减只有在与代码缩减：`minifyEnabled` 配合使用时才能发挥作用。在代码缩减器移除所有不使用的代码后，资源缩减器便可确定应用仍要使用的资源。

```
android {
    // Other settings
    buildTypes {
        release {
            minifyEnabled true
            shrinkResources true
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
```

使用Lint分析器（物理删除）

`lint` 工具是 Android Studio 中附带的静态代码分析器，可检测到 `res/` 文件夹中未被代码引用的资源。

从菜单栏中依次选择 **Analyze > Run Inspection By Name**



分析完成后会弹出：



`lint` 工具不会扫描 `assets/` 文件夹、通过反射引用的资源或已链接至应用的库文件。此外，它也不会移除资源，只会提醒您它们的存在。与资源缩减不同，这里点击删除，那就是把文件删了。

反射引用资源：`getResources().getIdentifier("layout_main","layout",getPackageName());`

自定义要保留的资源

如果有想要特别声明需要保留或舍弃的特定资源，在项目中创建一个包含 `<resources>` 标记的 XML 文件，并在 `tools:keep` 属性中指定每个要保留的资源，在 `tools:discard` 属性中指定每个要舍弃的资源。这两个属性都接受以逗号分隔的资源名称列表。还可以将星号字符用作通配符。

```
<?xml version="1.0" encoding="utf-8"?>
<resources xmlns:tools="http://schemas.android.com/tools"
    tools:keep="@layout/l_used*_c,@layout/l_used_a,@layout/l_used_b*"
    tools:discard="@layout/unused2" />
```

将该文件保存在项目资源中，例如，保存在 `res/raw/keep.xml` 中。构建系统不会将此文件打包到 APK 中。

一键删除无用资源

Android Studio 给我们提供了一键移除所有无用的资源。从菜单栏中依次选择 **Refactor > Remove Unused Resources**，但是这种方式不建议使用，因为如果某资源仅存在动态获取资源id 的方式，那么这个资源会被认为没有使用过，从而会直接被删除。

移除未使用的备用资源

Gradle 资源缩减器只会移除未由应用代码引用的资源，这意味着，它不会移除用于不同设备配置的备用资源。可以使用 Android Gradle 插件的 `resConfigs` 属性移除应用不需要的备用资源文件。

例如，如果使用的是包含语言资源的库（如 AppCompat），那么 APK 中将包含这些库中所有已翻译语言的字符串。如果只想保留应用正式支持的语言，则可以使用 `resConfig` 属性指定这些语言。系统会移除未指定语言的所有资源。

com.xiangxue.arch_demo (Version Name: 1.0, Version Code: 1)

APK size: 2.5 MB, Download Size: 1.9 MB

Compare with previous APK...

File	Raw File Size	Download Size	% of Total Download Size
classes.dex	1.3 MB	1.3 MB	67.1%
res	318.3 KB	309.2 KB	16.2%
resources.arsc	499.8 KB	98.1 KB	5.1%
kotlin	97.8 KB	97.7 KB	5.1%
META-INF	86.8 KB	86.8 KB	4.5%

Package: com.xiangxue.arch_demo

Resource Types

There are 150 string resources across 87 configurations

	ID	Name	default	ca	da	fa	ja	ka	pa	ta
anim	0x7f100000	abc_action_bar_home_description	Navigate ...	Navega a...	Find hjem	پیمایش به صفحه...	ホーム(に...	გთავაზობ...	000 00 00 000000	
animator	0x7f100001	abc_action_bar_up_description	Navigate ...	Navega c...	Gå op	رفتن به بالا	前に戻る	გაგრძელო...	0000 0000 0000 ...	
array	0x7f100002	abc_action_menu_overflow_description	More opti...	Més opci...	Flere valg...	گزینه‌های بیشتر	その他の...	სხვა ვარიან...	000 000000 000000	
attr	0x7f100003	abc_action_mode_done	Done	Fet	Udfør	تمام	完了	შესრულდა	00 0000 000000	
bool	0x7f100004	abc_activity_chooser_view_see_all	See all	Mostra-h...	Se alle	نمایش همه	すべて表示	ყველა ნა...	00 0000 000000	
color	0x7f100005	abc_activitychooserview_choose_applica...	Choose a...	Seleccion...	Vælg en a...	انتخاب برنامه	アプリの...	პირჩინეთ ...	000 00 00 000000	
dimen	0x7f100006	abc_capital_off	OFF	DESCRIPTIVA	FRA	خاموش	OFF	გამორთვა	000 0000	
drawable	0x7f100007	abc_capital_on	ON	ACTIVA	TIL	روشن	ON	ჩართვა	0000 000	
id	0x7f100008	abc_menu_alt_shortcut_label	Alt+	Alt+	Alt+	Alt+	Alt+	Alt+	Alt+	Alt 000
integer	0x7f100009	abc_menu_ctrl_shortcut_label	Ctrl+	Ctrl+	Ctrl+	Ctrl+	Ctrl+	Ctrl+	Ctrl+	Ctrl 000
interpolator	0x7f10000a	abc_menu_delete_shortcut_label	delete	Supr	slet	حذف	Delete	delete	000000 delete	
layout	0x7f10000b	abc_menu_enter_shortcut_label	enter	Retorn	enter	enter	Enter	enter	enter	enter
menu	0x7f10000c	abc_menu_function_shortcut_label	Function+	Funció+	Fn+	Function+	Function+	Function+	Function+	Function
mipmap	0x7f10000d	abc_menu_meta_shortcut_label	Meta+	Meta+	Meta+	Meta+	Meta+	Meta+	Meta+	Meta 00
plurals	0x7f10000e	abc_menu_shift_shortcut_label	Shift+	Maj+	Shift+	Shift+	Shift+	Shift+	Shift+	Shift 00
string	0x7f10000f	abc_menu_space_shortcut_label	space	Espai	mellemrum	فاصله	Space	პრობლემა	space	space
style	0x7f100010	abc_menu_sym_shortcut_label	Sym+	Sym+	Sym+	Sym+	Sym+	Sym+	Sym+	Sym 00
xml	0x7f100011	abc_prepend_shortcut_label	Menu+	Menú+	Menu+	منو +	Menu+	Menu+	Menu+	Menu 0
	0x7f100012	abc_search_hint	Search...	Cerca...	Søg...	جستجو...	検索...	ძიება...	0000...	000000
	0x7f100013	abc_searchview_description_clear	Clear query	Esborra l...	Ryd fores...	پاک کردن پرس...	検索キー一...	შოვნისგინ...	00000000 (0000...	
	0x7f100014	abc_searchview_description_query	Search qu...	Consulta ...	Søgefore...	درخواست جستجو	検索キー一...	შოვნისგინ...	000 000000 00000	
	0x7f100015	abc_searchview_description_search	Search	Cerca	Søg	جستجو	検索	ძიება	000 000000 000000	
	0x7f100016	abc_searchview_description_submit	Submit qu...	Envia la c...	Indsend f...	ارسال پرسن...	検索キー一...	შოვნისგინ...	00000000 (0000...	
	0x7f100017	abc_searchview_description_voice	Voice sea...	Cerca per...	Talesøgni...	جستجوی گفتاری	音声検索	ბგერითი ძ...	000000 00 000000	
	0x7f100018	abc_shareactionprovider_share_with	Share with	Comparte...	Del med	هم‌رسانی با	共有	გაზიარების...	00 000 000 00000	
	0x7f100019	abc_shareactionprovider_share_with_ap...	Share wit...	Comparte...	Del med ...	هم‌رسانی با %s	%s と共有	%s-ით გა...	%s 000 000 %s 000	
	0x7f10001a	abc_toolbar_collapse_description	Collapse	Replega	Skjul	تکریک کردن	折りたたむ	შეკუმცე	000000 000000	

```

android {
    defaultConfig {
        ...
        resConfigs "zh-rCN"
    }
}

```

配置resConfigs 只打包默认与简体中文资源。

build.gradle (:app) × app-release.apk × MainActivity.java × keep.xml × AndroidManifest.xml ×

com.xiangxue.arch_demo (Version Name: 1.0, Version Code: 1)

APK size: 2.3 MB, Download Size: 1.8 MB

File	Raw File Size	Download
classes.dex	1.3 MB	1.3
res	318.5 KB	309.4
kotlin	97.8 KB	97.7
META-INF	86.9 KB	86.9
resources.arsc	350.2 KB	69.1
okhttp3	33.2 KB	33.2
AndroidManifest.xml	2.8 KB	2.8
assets	735 B	73
com	170 B	17

Package: com.xiangxue.arch_demo

Resource Types

id

integer

interpolator

layout

menu

mipmap

plurals

raw

string

style

xml

There are 150 string resources across 3 configurations

ID	Name	default	zh	zh-rCN
0x7f110000	abc_action_bar_home_description	Navigate ...		转到首页
0x7f110001	abc_action_bar_up_description	Navigate ...		转到上一...
0x7f110002	abc_action_menu_overflow_description	More opti...		更多选项
0x7f110003	abc_action_mode_done	Done		完成
0x7f110004	abc_activity_chooser_view_see_all	See all		查看全部
0x7f110005	abc_activitychooserview_choose_applica...	Choose a...		选择应用
0x7f110006	abc_capital_off	OFF		关闭
0x7f110007	abc_capital_on	ON		开启
0x7f110008	abc_menu_alt shortcut label	Alt+		Alt+

动态库打包配置

so文件是由ndk编译出来的动态库，是 c/c++ 写的，所以不是跨平台的。ABI 是应用程序二进制接口简称（Application Binary Interface），定义了二进制文件（尤其是.so文件）如何运行在相应的系统平台上，从使用的指令集，内存对齐到可用的系统函数库。在Android 系统中，每一个CPU架构对应一个ABI，目前支持的有：armeabi-v7a, arm64- v8a, x86, x86_64。目前市面上手机设备基本上都是arm架构，armeabi-v7a 几乎能兼容所有设备。因此可以配置：

```
android{
    defaultConfig{
        ndk{
            abiFilters "armeabi-v7a"
        }
    }
}
```

对于第三方服务，如百度地图、Bugly等会提供全平台的cpu架构。进行了上面的配置之后，表示只会把armeabi-v7a打包进入Apk。从而减少APK大小。

对于arm64架构的设备，如果使用armeabi-v7a也能够兼容，但是不如使用arm64的so性能。因此现在部分应用市场会根据设备提供不同架构的Apk安装。此时我们需要打包出针对arm64的apk与armv7a的apk，可以使用productFlavor。

```
flavorDimensions "default"
productFlavors{
    arm32{
        dimension "default"
    }
}
```

```

        ndk{
            abiFilters "armeabi-v7a"
        }
    }
    arm64{
        dimension "default"
        ndk{
            abiFilters "arm64-v8a"
        }
    }
}

```

也可以使用：

```

splits {
    abi {
        enable true
        reset()
        include 'arm64-v8a','armeabi-v7a'
        // exclude 'armeabi'
        universalApk true //是否打包一个包含所有so的apk
    }
}

```

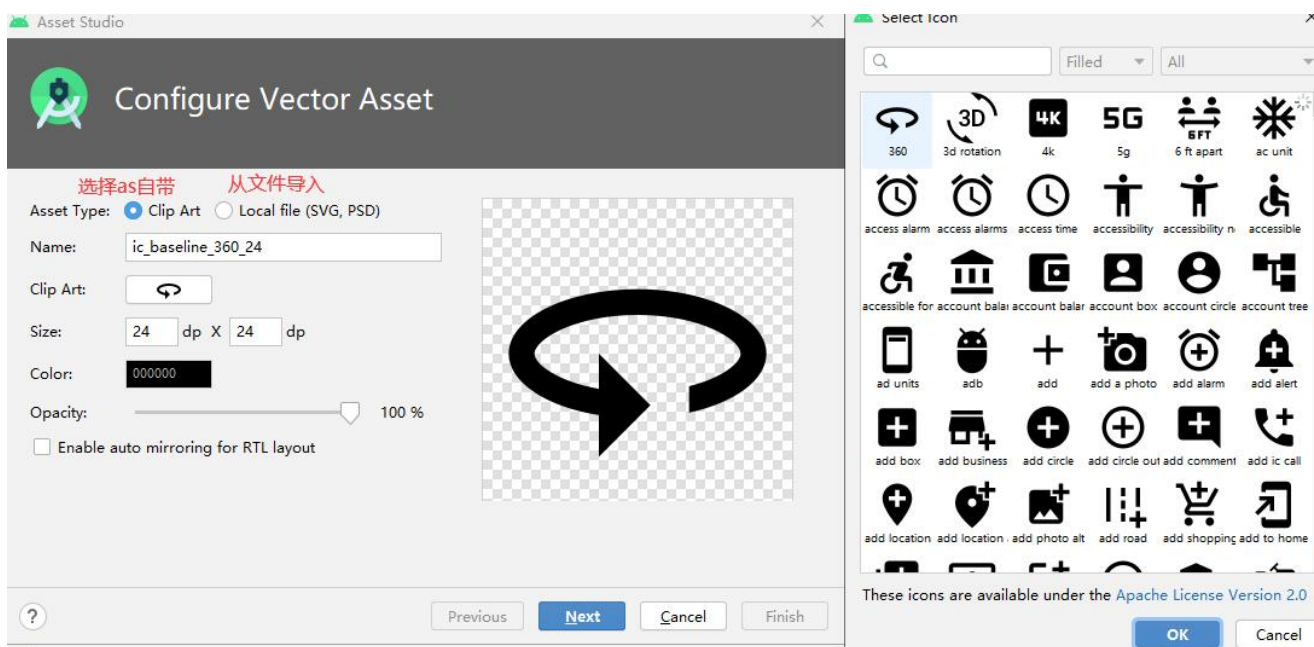
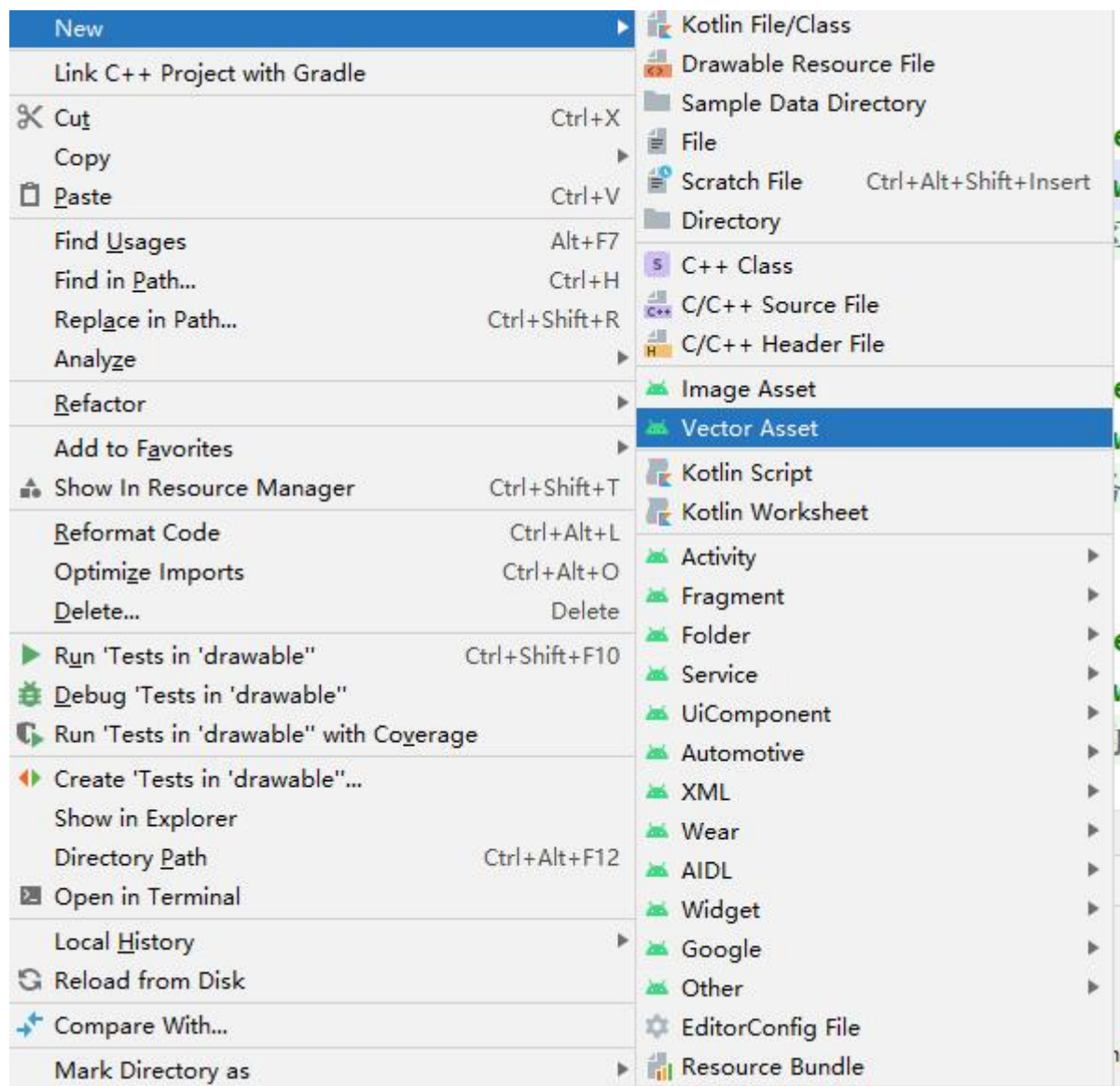
使用矢量图

Apk中图片应该算是占用空间最多的资源。我们可以使用webp减少png、jpg图片占用空间的大小。对于小图标也可以使用矢量图。

矢量图可以创建与分辨率无关的图标和其他可伸缩媒体。使用这些图形可以极大地减少 APK 占用的空间。矢量图片在 Android 中以 `VectorDrawable` 对象的形式表示。借助 `VectorDrawable` 对象，100 字节的文件可以生成与屏幕大小相同的清晰图片。

不过，系统渲染每个 `VectorDrawable` 对象需要花费大量时间，而较大的图片则需要更长的时间才能显示在屏幕上。因此，建议仅在显示小图片时使用这些矢量图。

新工程默认Icon就是矢量图。



重复使用资源

现在我们有一个矢量图：

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24"
    android:viewportHeight="24"
    android:tint="?attr/colorControlNormal">
    <path
        android:fillColor="@android:color/white"
        android:pathData="M10,20v-6h4v6h5v-8h3L12,3 2,12h3v8z"/>
</vector>
```

它的显示效果为：



如果我们需要让矢量图显示红色怎么办？这种情况，我们不需要再去创建一个新的矢量图。可以直接给 `ImageView` 设置 `android:tint` 属性 来完成颜色的修改。

```
<ImageView
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:tint="@color/colorAccent"
    android:src="@drawable/tabbar_home_vector" />
```



选择器

如果需要让矢量图实现触摸变色。只需要创建selector，设置给tint即可

```
<!-- tabbar_home_tint_selector -->
<?xml version="1.0" encoding="utf-8"?>
```

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:color="@color/colorPrimary" android:state_pressed="true" />
    <item android:color="@color/colorAccent" />
</selector>


<ImageView
    android:clickable="true"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:src="@drawable/tabbar_home_vector"
    android:tint="@color/tabbar_home_tint_selector" />
```

阿里矢量图库：

<https://www.iconfont.cn/help/detail?spm=a313x.7781069.1998910419.d8d11a391&helptype=code>

其他

- 使用精简版本的依赖：如protobuf-lite版本；对于分模块的库按需引入：如netty分模块引入；
- 主动移除无用代码（开启R8/Proguard自动移除）
- 避免使用枚举，使用 `@IntDef` 代替。
- 开启资源混淆：<https://github.com/shwenzhang/AndResGuard>
- 支付宝删除Dex debugItem <https://juejin.im/post/6844903712201277448>

对于发布Google play的应用选择使用：AAB <https://developer.android.google.cn/guide/app-bundle>