64k方法数限制解决方案

Eclipse下Android 64k方法数的解决方案

eclipse Android工程方法超过65535，一般使用MultiDex库进行分包处理。 而eclipse中使用multidex库一般分为以下步骤：

- 下载android-support-multidex.jar
- jar包合并
- 合并后的jar包转换为classes.dex文件
- 将classes.dex文件放置到项目的src目录
- 配置MultiDexApplication

android-support-multidex.jar下载

百度下载android-support-multidex.jar放置在主工程使用

jar包合并

jar包合并有很多种方式，可以使用jdk提供的jar命令、ant脚本等。这里介绍使用ant命令。

> ant命令: ant -buildfile b.xml

执行命令获取合并后的libs.jar文件（使用ant脚本需要安装ant，如何安装可以自行百度）

注意：合并后的jar包还需要引用参与编译，可以通过引用第三方库的方式进行引用，这样jar包不会打入到apk中（放置在libs目录下的默认都会打入到apk）。

操作路径如下：

- 主工程Build Path => Add External Archives...

编写的b.xml文件如下：

> basedir 为所合并jar的目录 destfile 为输出合并后的jar包全路径 zipfileset 为需要合并的jar包

```
<?xml version="1.0" encoding="utf-8"?>
<project name="libs" basedir="E:\AntPackager\test\smjq" default="makeSuperJar">
<target name="makeSuperJar"  description="description">
    <jar destfile="libs.jar">
        <zipfileset src="support-annotations-25.0.0.jar"/>
        <zipfileset src="support-compat-25.0.0.jar"/>
        <zipfileset src="support-core-ui-25.0.0.jar"/>
        <zipfileset src="support-core-utils-25.0.0.jar"/>
        <zipfileset src="support-fragment-25.0.0.jar"/>
        <zipfileset src="support-internal_impl-25.0.0.jar"/>
        <zipfileset src="support-media-compat-25.0.0.jar"/>
                <zipfileset src="swipe-recyclerview.jar"/>
    </jar>
</target>
</project>
```

合并后的jar包转换为classes.dex文件

jar包转化为dex文件，可以使用sdk提供dx工具。 dx工具位于sdk/build-tools/26.0.2目录下。 dx命令： dx --dex --output=outputfile srcfile 示例：dx --dex --output=E:\libs\classes.dex E:\libs\all.jar

> outputfile为输出文件的dex文件 srcfile为所转化的jar包

将classes.dex文件放置到项目的src目录

MultiDexApplication配置

如果您没有替换 Application 类，请编辑清单文件，按如下方式设置 <application> 标记中的 android:name：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapp">
    <application
            android:name="android.support.multidex.MultiDexApplication" >
        ...
    </application>
</manifest>
```

如果替换了 Application 类，请按如下方式对其进行更改以扩展 MultiDexApplication（如果可能）：

```
public class MyApplication extends MultiDexApplication { ... }
```

或者，如果替换了 Application 类，但无法更改基本类，则可以改为替换 attachBaseContext() 方法并调用 MultiDex.install(this) 来启用 Dalvik 可执行文件分包：

```
public class MyApplication extends SomeOtherApplication {
  @Override
  protected void attachBaseContext(Context base) {
     super.attachBaseContext(context);
     Multidex.install(this);
  }
}
```

Android Studio下Android 64k方法数的解决方案

gradle配置

- 在主工程的build.gradle文件中增加配置：

```
dependencies {
    compile 'com.android.support:multidex:1.0.2'
}
android {
    defaultConfig{
        multiDexEnabled true
    }
}
```

AndroidManifest配置

- 在主工程的AndroidManifest.xml文件中，找到游戏Application标签，增加一行配置：`android:name="android.support.multidex.MultiDexApplication"`
- 如果代码中有声明游戏Applicaition，请让Applicaition继承MultiDexApplication类

```
<application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:name="android.support.multidex.MultiDexApplication"
        android:label="@string/app_name" >
```