[**［Android Studio 权威教程］多渠道打包和一键完成（全部产品）打包并签名**](http://blog.csdn.net/yy1300326388/article/details/48296221)

分类： [Android Studio](http://blog.csdn.net/yy1300326388/article/category/2891739)2015-09-08 20:16 778人阅读 [评论](http://blog.csdn.net/yy1300326388/article/details/48296221#comments)(3) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/yy1300326388/article/details/48296221#report)

[android](http://www.csdn.net/tag/android)[android studio](http://www.csdn.net/tag/android%20studio)[渠道](http://www.csdn.net/tag/%e6%b8%a0%e9%81%93)[打包签名](http://www.csdn.net/tag/%e6%89%93%e5%8c%85%e7%ad%be%e5%90%8d)[多渠道打包](http://www.csdn.net/tag/%e5%a4%9a%e6%b8%a0%e9%81%93%e6%89%93%e5%8c%85)

目录[(?)[+]](http://blog.csdn.net/yy1300326388/article/details/48296221)

好久没有更新blog了，今天给大家带来的是AS 的多渠道打包，并且我们还要验证是否实现了多渠道打包的功能，最后在让大家爽一下实现一键打包所有的渠道包并且给apk签名

多渠道打包

方法1

第一步：添加渠道表示标签

<!-- UMeng 配置-->

<meta-data android:value="${UMENG\_CHANNEL\_VALUE}" android:name="UMENG\_CHANNEL"/>

* 1
* 2
* 完整截图



第二步：添加渠道

productFlavors {

xiaomi {}

qihu360 {}

baidu {}

wandoujia {}

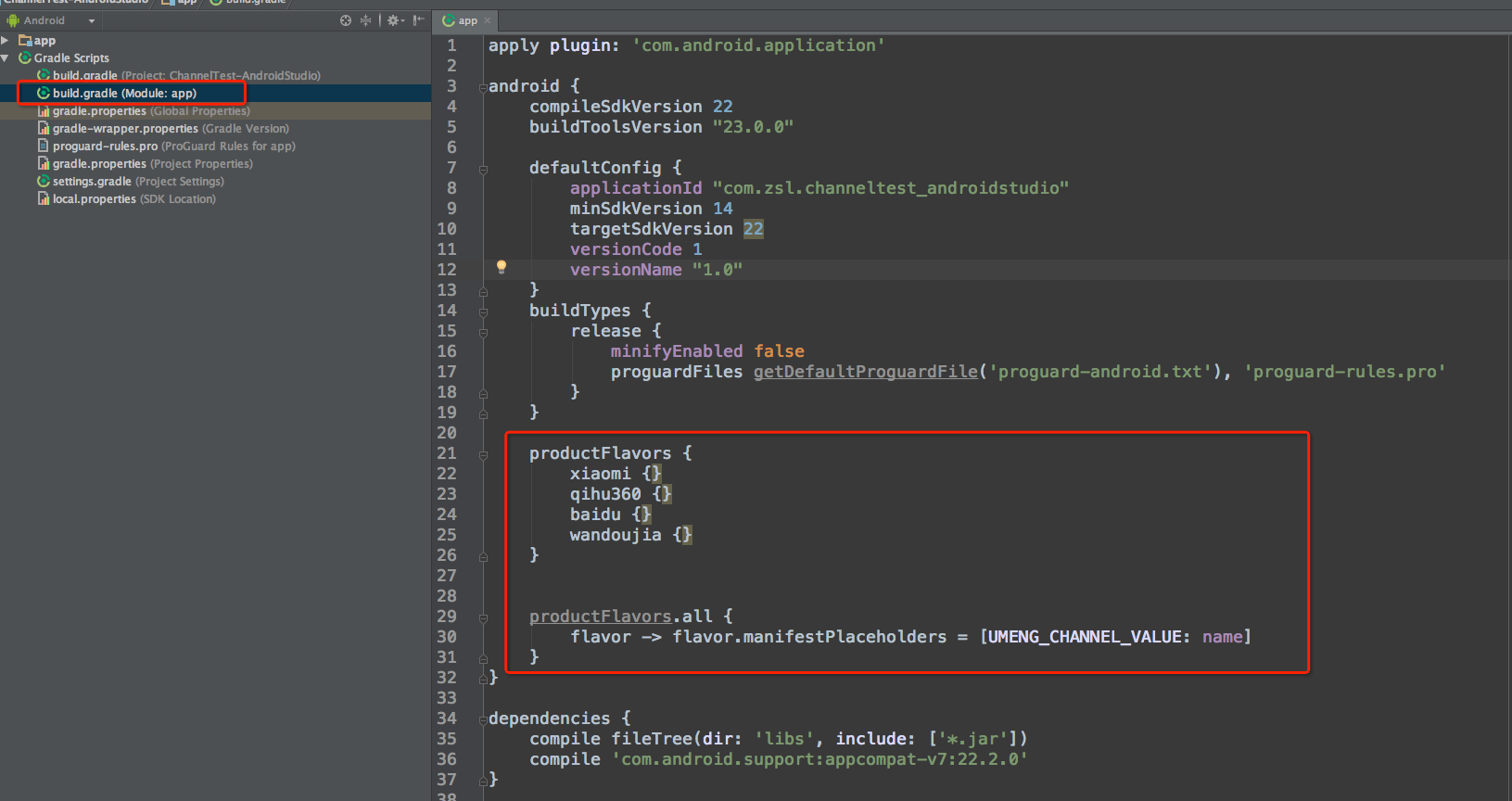
}

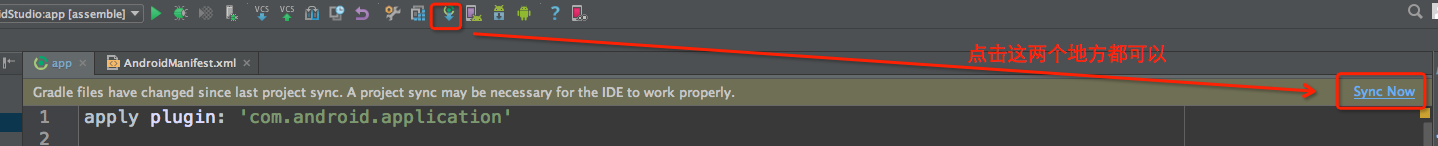
productFlavors.all {

flavor -> flavor.manifestPlaceholders = [UMENG\_CHANNEL\_VALUE: name]

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 完整的截图



添加完之后我们在点击一下Gradle 的构建按钮重新构建一下   


第三步：测试

我们通过java代码获取到meta标签中的value来测试是否成功实现了多渠道打包：

public class MainActivity extends AppCompatActivity {

TextView tv\_channel;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tv\_channel= (TextView) findViewById(R.id.main\_tv\_channel);

tv\_channel.setText(getApplicationMetaValue("UMENG\_CHANNEL"));

}

private String getApplicationMetaValue(String name) {

String value= "";

try {

ApplicationInfo appInfo =getPackageManager()

.getApplicationInfo(getPackageName(),

PackageManager.GET\_META\_DATA);

value = appInfo.metaData.getString(name);

} catch (PackageManager.NameNotFoundException e) {

e.printStackTrace();

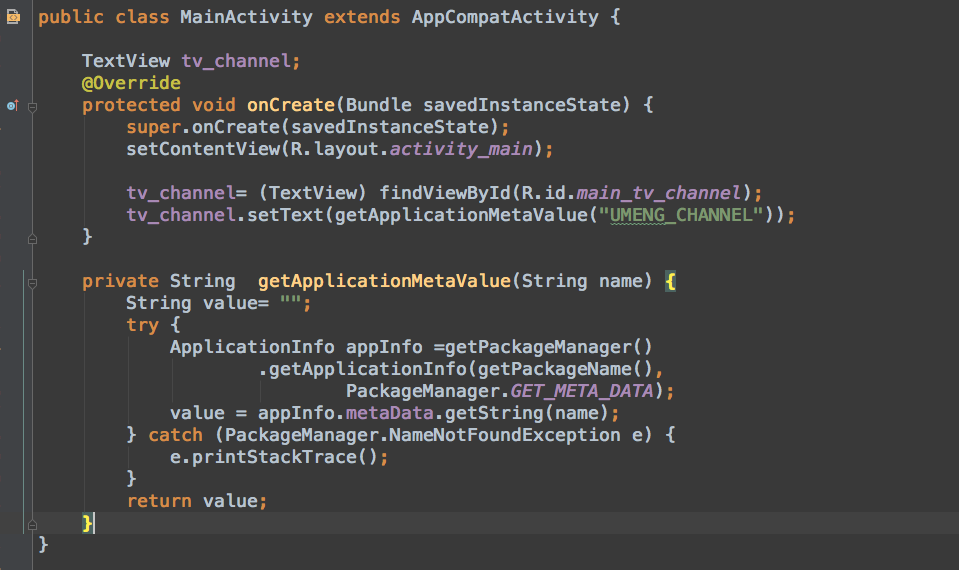
}

return value;

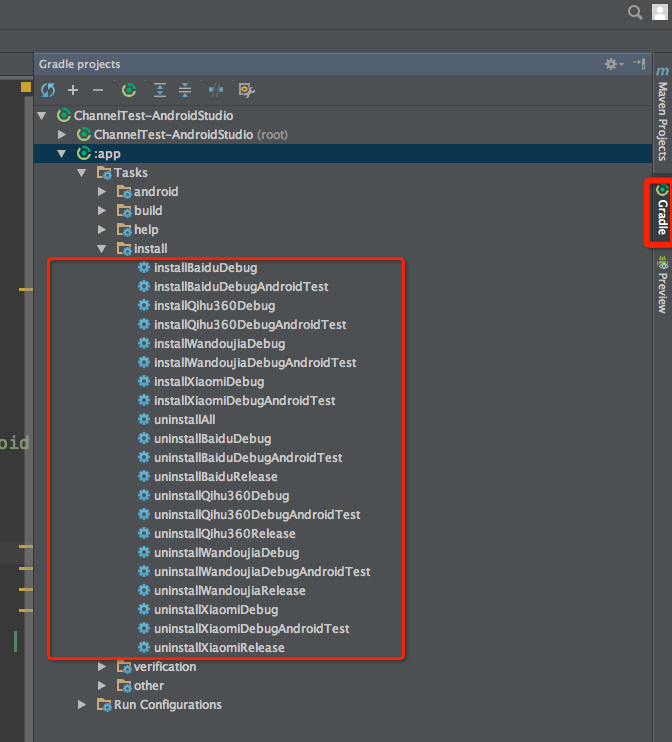
}

}

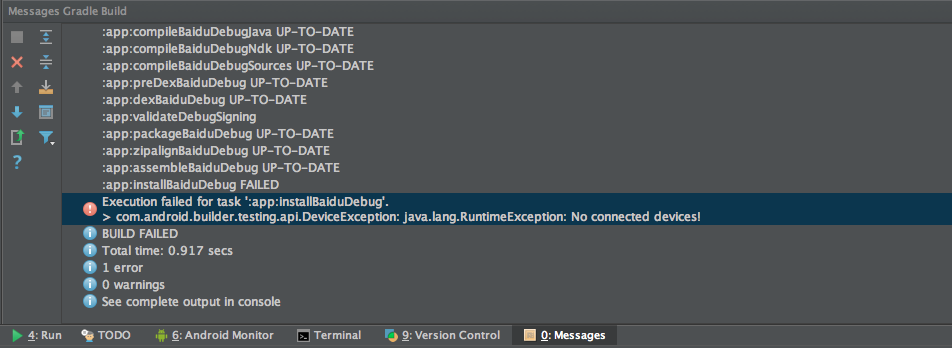
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 完整截图



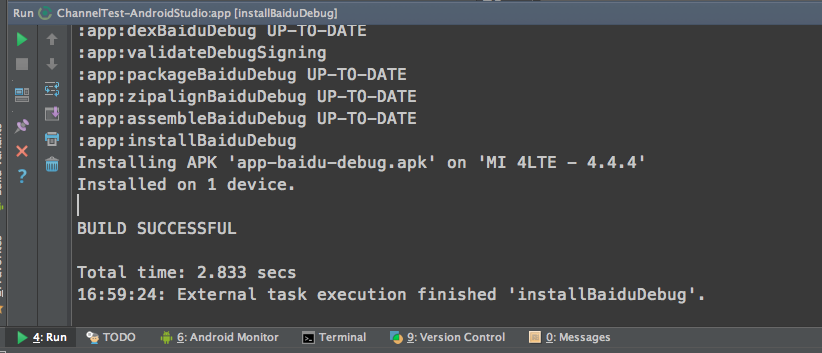
测试代码添加完毕之后，我们现在点击AS右侧的Gradle，我们此时就可以看到我们刚才添加的xiaomi、qihu360、baidu、wandoujia都出现了，如下图



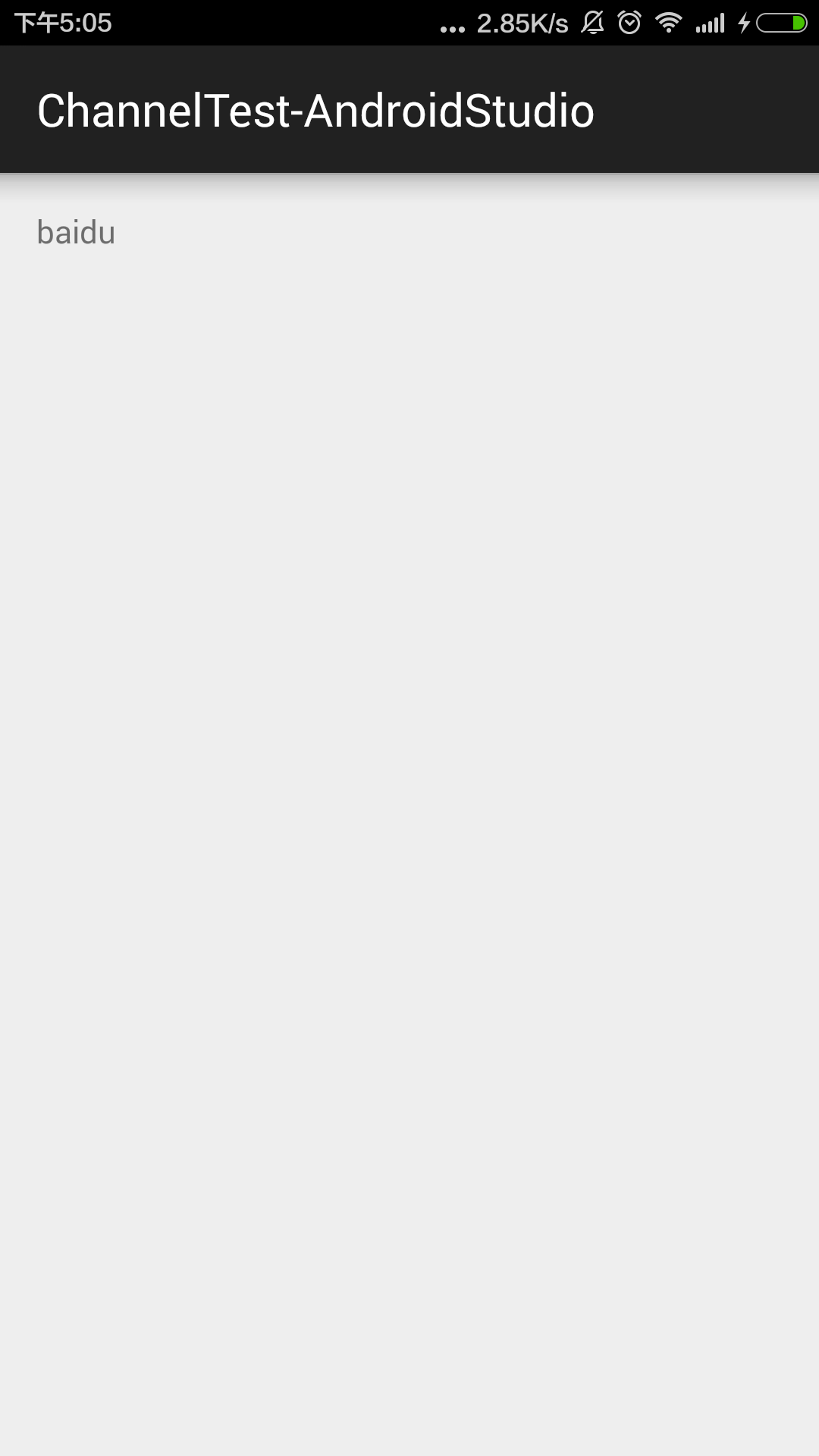
此时我们双击［installBaiduDebug］,然后就可以看到如下结果：



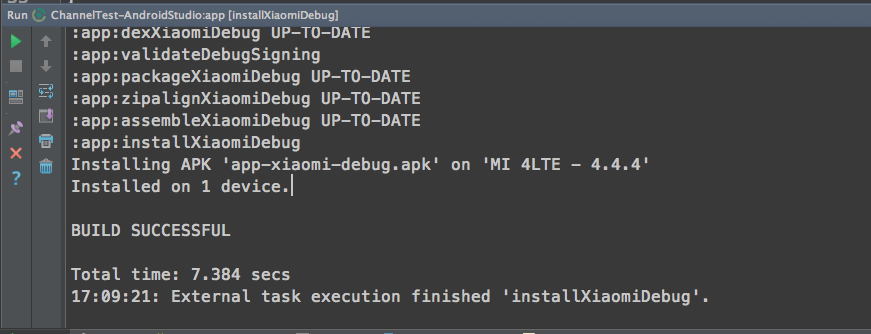
这说明你没有链接手机，链接上手机继续重新双击一下，结果如下



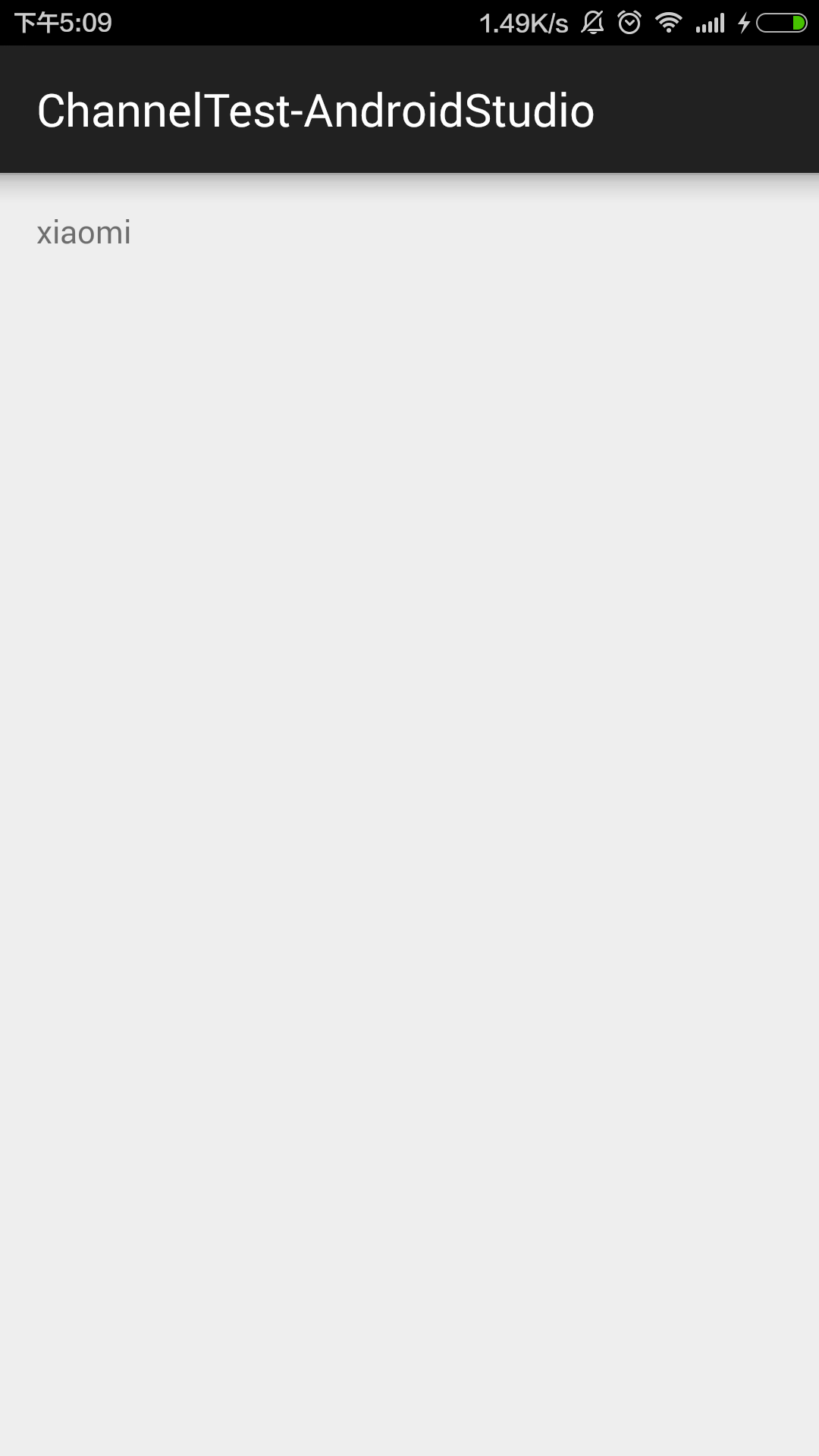
这就说明你成功的安装到你的手机上了，此时不会自动给你打开app，需要手动打开，效果如下：

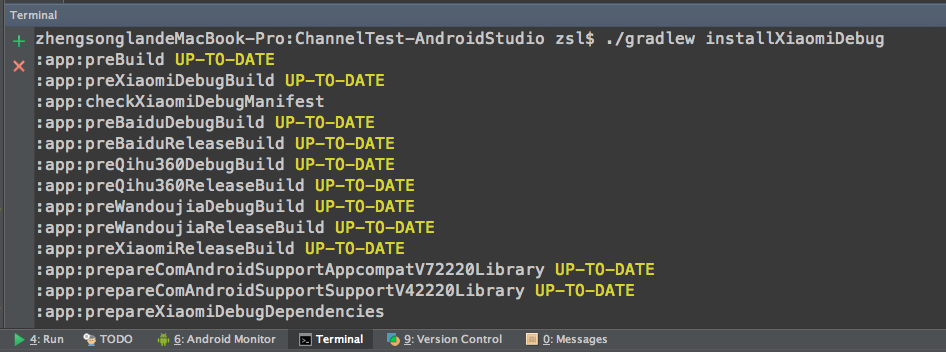


然后我们在双击［installXiaomiDebug］，然后看看



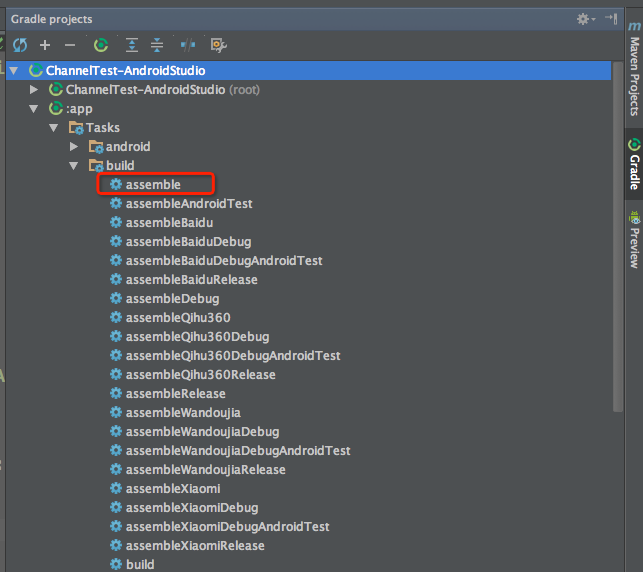
效果



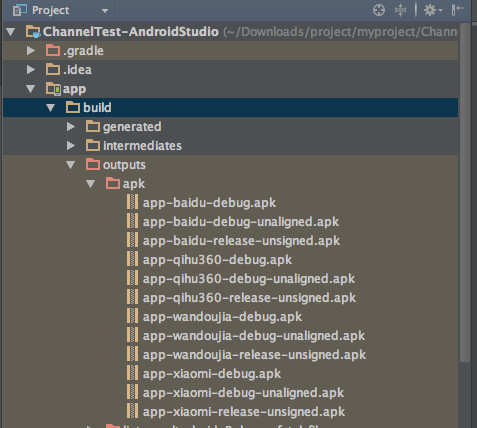
到这里我们来做一个小结，这里我们在ManiFest中添加了meta标签、在gradle(app)中添加了渠道产品对应的渠道名称 ，最后我们测试多渠道打包，这里还最重要的一点就是我这里没有让大家去运行什么命令，全部是让大家点击来实现的，其实点击就是在运行命令：./gradew installXiaomiDebug 效果如下:   


一键完成打包（全部产品）并签名

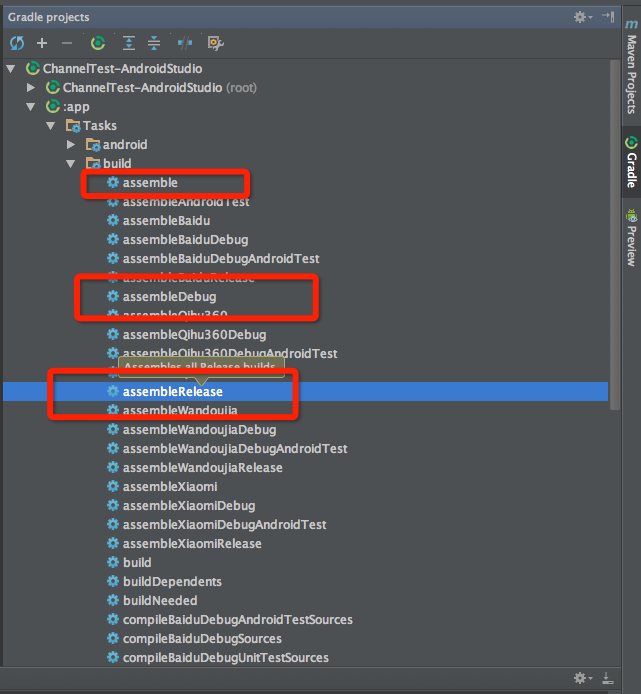
正如标题所示，我们如果要打包所有的产品，肯定不能像刚才一样一个一个来打，那么现在来使用另外一个按钮(命令)assemble



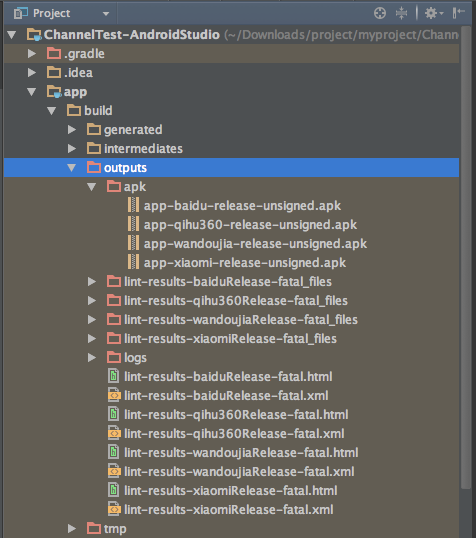
然后切换项目到Project视图，找到app>build>outputs>apk,结果如下我们看到所有版本的apk都有了，有debug版的，debug没有签名的，release没有签名的。



如果我们现在只想打包所有的release版本怎么办呢？看下面的图，如果想打release版就双击[assembleRelease] ,如果想打所有的debug版本就双击[assembleDebug]



* 小测试（我现在把apk目录下的所有apk都删除掉，然后只打所有release版本），效果如下：

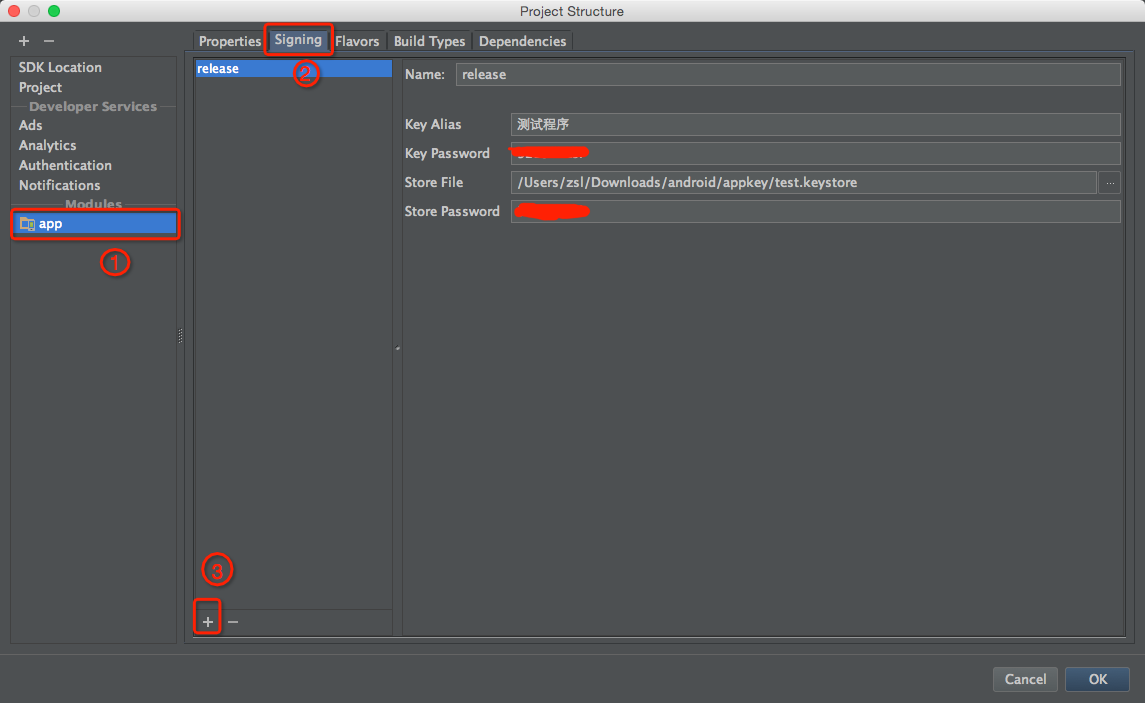


* 问题（所有的apk生成了，但是没有签名是发布不了的，肿莫办？）

点击项目架构按钮，如下：

这里写图片描述

然后点击app>Signing>+，然后填写上签名信息，最后点击OK



我们做的上述操作其实就是在gradle文件中添加如下信息，不信你打开你的gradle看看：

signingConfigs {

release {

keyAlias '测试程序'

keyPassword '证书密码'

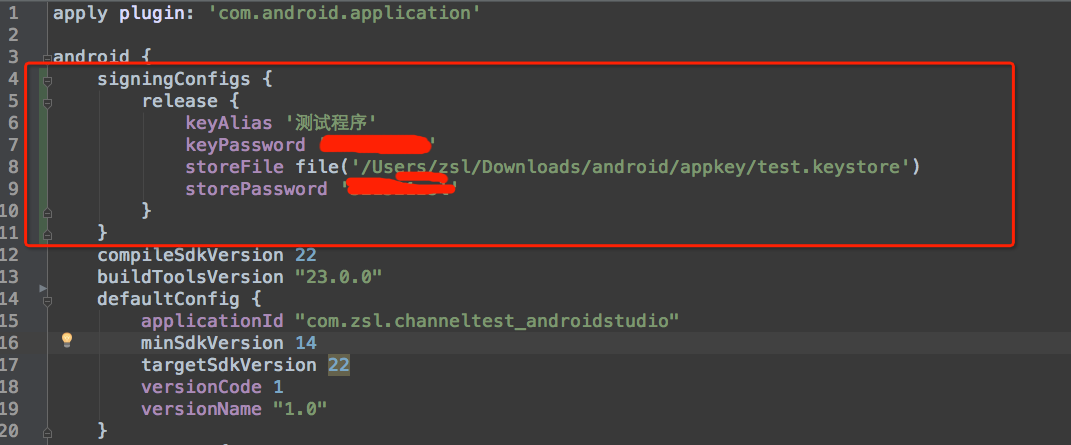
storeFile file('/Users/zsl/Downloads/android/appkey/test.keystore')

storePassword '签名密码'

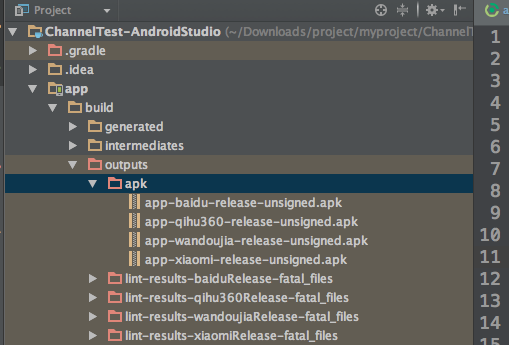
}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 完整截图



此时我们把apk下面的apk文件删掉，然后双击[assembleRelease]，结果发现还是没有签名啊

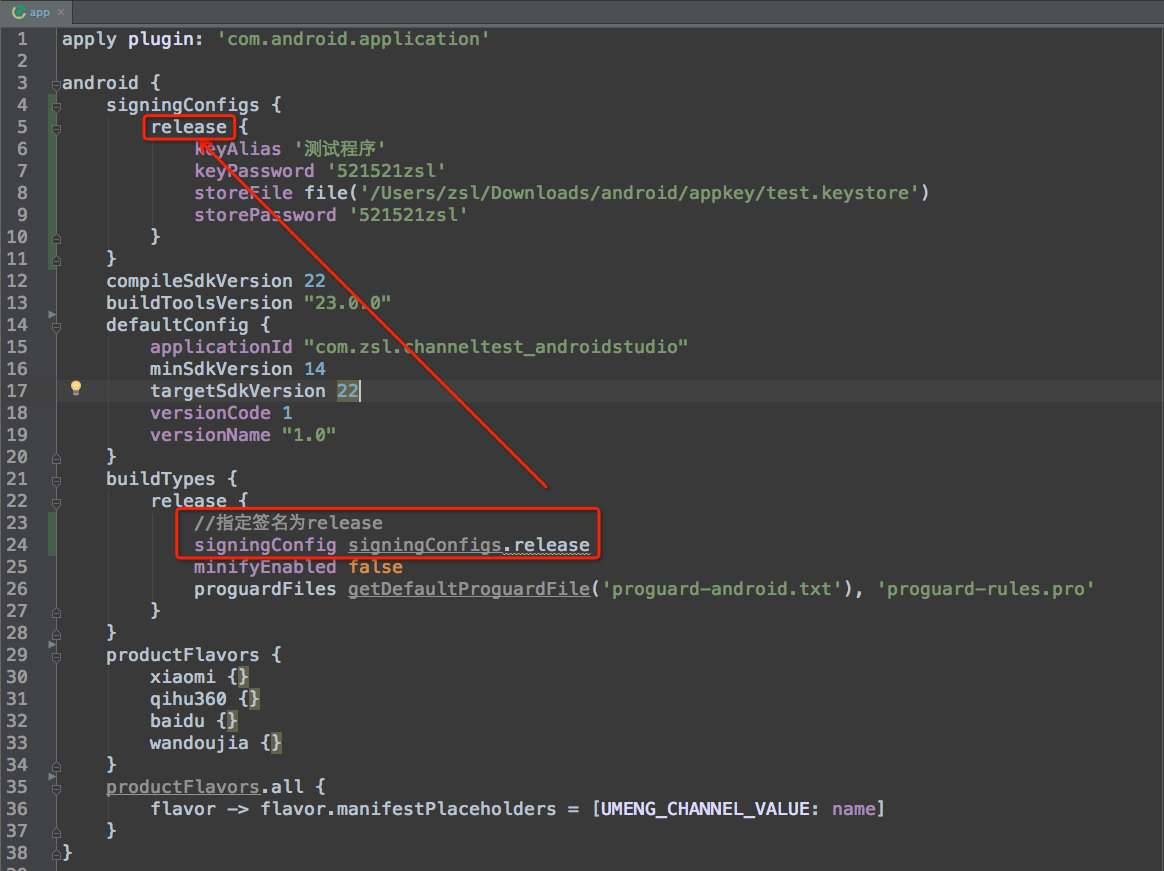


这是因为我们只是配置了签名信息，但是我们没有给产品(apk)指定，我们只需要在Gradle文件中添加如下的代码就 ok了，我们来试试看看

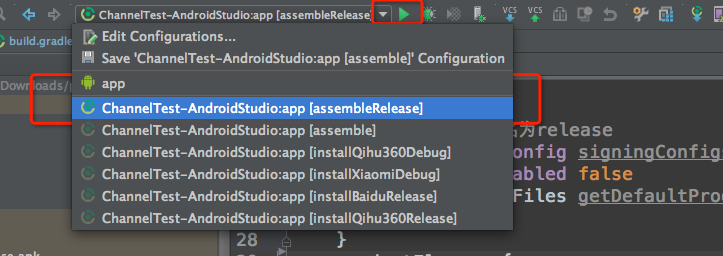
//指定签名为release

signingConfig signingConfigs.release

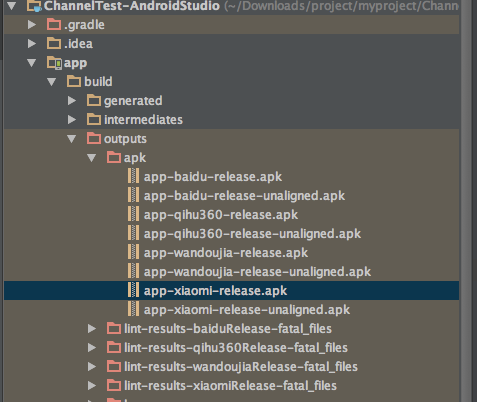
* 1
* 2
* 完整截图



我们现在删掉apk文件夹下所有的apk，然后选择［assembleRelease］然后点击run按钮，如下图，这里的run会记录我们最近使用的run脚本名称，



现在我们来看看apk目录下的安装包是否签名：



总结

开始我们在manifast文件中添加了meta渠道标签，然后修改了gradle的文件添加了不同的渠道产品，最后我们在代码中读取了meta标签的value并且显示在一个textview上，我们做了测试。   
接下来我们又添加了签名配置，最后我们指定了给release版本的签名，签名的名称是release，最后我们终于搞定了一键多渠道打包签名，下一篇给大家分享AS生成jks签名证书，并且apk签名。