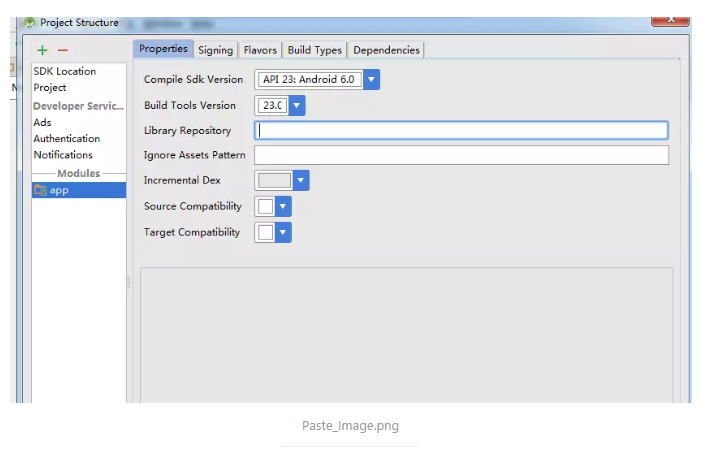
**url:** <https://www.jianshu.com/p/1d69b3ade53a?t=1493709593357>

**Android差分包生成和合成新包-bsdiff**

2017.04.30 14:40:40字数 510阅读 3673

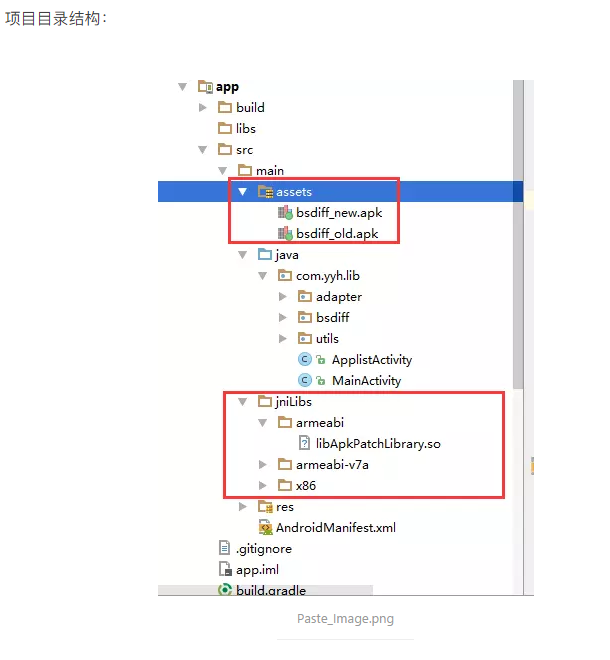
本文章中用到的两个下载链接：  
（1）电脑端差分包生成和合并工具  
[http://download.csdn.net/detail/suijing/9829716](https://link.jianshu.com/?t=http://download.csdn.net/detail/suijing/9829716)  
（2）测试Demo项目源码下载地址  
[http://download.csdn.net/detail/suijing/9829722](https://link.jianshu.com/?t=http://download.csdn.net/detail/suijing/9829722)

测试验证环境：



Paste\_Image.png

项目目录结构：



Paste\_Image.png

build.gradle文件：  
<pre>  
apply plugin: 'com.android.application'

android {  
compileSdkVersion 23  
buildToolsVersion '23.0.1'

defaultConfig {

applicationId "com.yuyh.inc.update"

minSdkVersion 14

targetSdkVersion 23

versionCode 1

versionName "1.0"

ndk{

moduleName "ApkPatchLibrary"

abiFilters "armeabi", "armeabi-v7a", "x86"

}

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

sourceSets {

main {

jni.srcDirs = ['src/main/jni', 'src/main/jni/']

// jniLibs.srcDirs = ['libs'] // 若不想编译jni代码，可直接引用so库，ndk编译相关脚本注释掉

}

}

}

dependencies {  
compile fileTree(include: ['\*.jar'], dir: 'libs')  
testCompile 'junit:junit:4.12'  
compile 'com.android.support:appcompat-v7:23.+'  
}  
</pre>

核心代码：  
<pre>  
package com.yyh.lib;

import android.app.Activity;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.os.Environment;  
import android.os.Handler;  
import android.os.Message;  
import android.view.View;  
import android.widget.ProgressBar;  
import android.widget.Toast;

import com.yuyh.inc.update.R;  
import com.yyh.lib.bsdiff.DiffUtils;  
import com.yyh.lib.bsdiff.PatchUtils;  
import com.yyh.lib.utils.FileUtils;

import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;

public class MainActivity extends Activity {

private ProgressBar loadding;

// 成功

private static final int WHAT\_SUCCESS = 1;

// 失败

private static final int WHAT\_FAIL\_PATCH = 0;

/\*\*

\* 老版本apk

\*/

private String oldApkDir = Environment.getExternalStorageDirectory().toString() + "/bsdiff\_old.apk";

/\*\*

\* 新版本apk

\*/

private String newApkDir = Environment.getExternalStorageDirectory().toString() + "/bsdiff\_new\_build.apk";

/\*\*

\* 根据新老版本差分包生成的新版本apk

\*/

private String destDir2 = Environment.getExternalStorageDirectory().toString() + "/bsdiff\_new.apk";

/\*\*

\* 新老版本的增量更新包

\*/

private String oldToNewPatchDir = Environment.getExternalStorageDirectory().toString() + "/old-to-new.patch";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

loadding = (ProgressBar) findViewById(R.id.loadding);

}

private Handler handler = new Handler() {

@Override

public void handleMessage(Message msg) {

switch (msg.what) {

case 0:

Toast.makeText(getApplicationContext(), "复制成功", Toast.LENGTH\_SHORT).show();

break;

case 1:

Toast.makeText(getApplicationContext(), "复制失败", Toast.LENGTH\_SHORT).show();

break;

case 2:

Toast.makeText(getApplicationContext(), "生成增新包成功", Toast.LENGTH\_SHORT).show();

break;

case 3:

Toast.makeText(getApplicationContext(), "生成增新包失败", Toast.LENGTH\_SHORT).show();

break;

case 4:

Toast.makeText(getApplicationContext(), "合成新版本成功", Toast.LENGTH\_SHORT).show();

break;

case 5:

Toast.makeText(getApplicationContext(), "合成新版本失败", Toast.LENGTH\_SHORT).show();

break;

default:

break;

}

}

};

/\*\*

\* 复制文件

\* @param view

\*/

public void copy(View view) {

loadding.setVisibility(View.VISIBLE);

new CopyTask().execute(oldApkDir, "bsdiff\_old.apk", newApkDir, "bsdiff\_new.apk");

}

/\*\*

\* 生成差分增量包

\* @param view

\*/

public void bsdiff(View view) {

loadding.setVisibility(View.VISIBLE);

new DiffTask().execute();

}

/\*\*

\* 合成新版本

\* @param view

\*/

public void bspatch(View view) {

loadding.setVisibility(View.VISIBLE);

new PatchTask().execute();

}

/\*\*

\* 安装老版本

\* @param view

\*/

public void installOld(View view) {

install(oldApkDir);

}

/\*\*

\* 安装新版本

\* @param view

\*/

public void installNew(View view) {

install(destDir2);

}

/\*\*

\* 复制文件线程

\*/

private class CopyTask extends AsyncTask<String, Void, Integer> {

@Override

protected Integer doInBackground(String... params) {

for (int i = 0; i < params.length; i += 2) {

try {

File file = new File(params[i]);

if (!file.exists())

FileUtils.createFile(file);

InputStream is;

OutputStream os = new FileOutputStream(params[i]);

is = getAssets().open(params[i + 1]);

byte[] buffer = new byte[1024];

int length = is.read(buffer);

while (length > 0) {

os.write(buffer, 0, length);

length = is.read(buffer);

}

os.flush();

is.close();

os.close();

} catch (Exception e) {

handler.obtainMessage(1).sendToTarget();

return null;

}

}

handler.obtainMessage(0).sendToTarget();

return null;

}

@Override

protected void onPostExecute(Integer integer) {

super.onPostExecute(integer);

loadding.setVisibility(View.GONE);

}

}

/\*\*

\* 生成差分包线程

\*

\* @author yuyuhang

\* @date 2016-1-25 下午12:24:34

\*/

private class DiffTask extends AsyncTask<String, Void, Integer> {

@Override

protected Integer doInBackground(String... params) {

try {

int result = DiffUtils.getInstance().genDiff(oldApkDir, newApkDir, oldToNewPatchDir);

if (result == 0) {

handler.obtainMessage(2).sendToTarget();

return WHAT\_SUCCESS;

} else {

handler.obtainMessage(3).sendToTarget();

return WHAT\_FAIL\_PATCH;

}

} catch (Exception e) {

e.printStackTrace();

}

return WHAT\_FAIL\_PATCH;

}

@Override

protected void onPostExecute(Integer integer) {

super.onPostExecute(integer);

loadding.setVisibility(View.GONE);

}

}

/\*\*

\* 差分包合成APK线程

\*

\* @author yuyuhang

\* @date 2016-1-25 下午12:24:34

\*/

private class PatchTask extends AsyncTask<String, Void, Integer> {

@Override

protected Integer doInBackground(String... params) {

try {

int result = PatchUtils.getInstance().patch(oldApkDir, destDir2, oldToNewPatchDir);

if (result == 0) {

handler.obtainMessage(4).sendToTarget();

return WHAT\_SUCCESS;

} else {

handler.obtainMessage(5).sendToTarget();

return WHAT\_FAIL\_PATCH;

}

} catch (Exception e) {

e.printStackTrace();

}

return WHAT\_FAIL\_PATCH;

}

@Override

protected void onPostExecute(Integer integer) {

super.onPostExecute(integer);

loadding.setVisibility(View.GONE);

}

}

/\*\*

\* 安装软件

\* @param dir

\*/

private void install(String dir) {

String command = "chmod 777 " + dir;

Runtime runtime = Runtime.getRuntime();

try {

runtime.exec(command); // 可执行权限

} catch (IOException e) {

e.printStackTrace();

}

Intent intent = new Intent(Intent.ACTION\_VIEW);

intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

intent.setDataAndType(Uri.parse("file://" + dir), "application/vnd.android.package-archive");

startActivity(intent);

}

/\*

public void appList(View view) {

Intent intent = new Intent(this, ApplistActivity.class);

startActivity(intent);

}

\*/

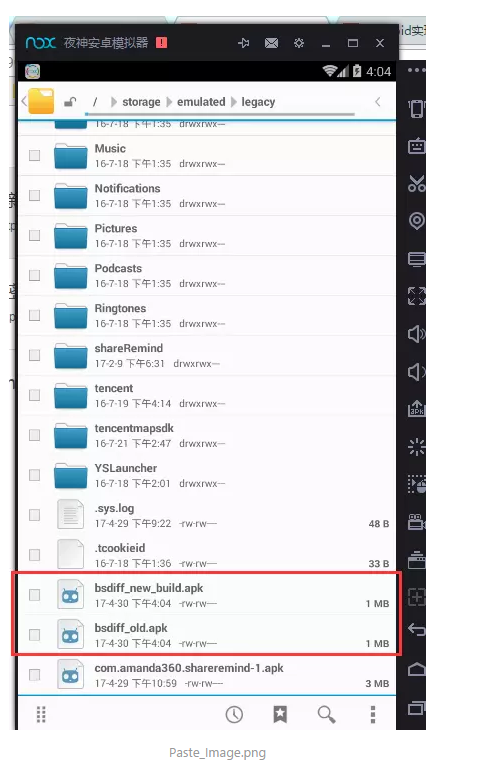
}  
</pre>

1、软件运行效果图



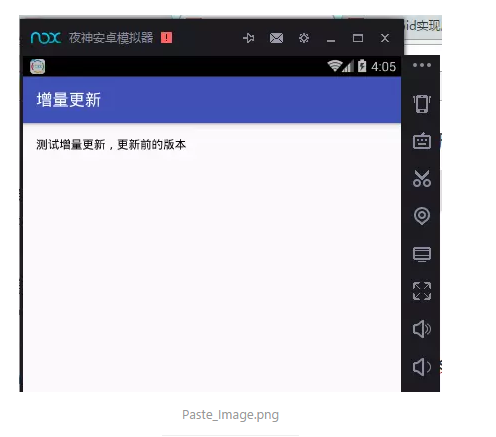
Paste\_Image.png

2、将Asset中的新旧版本复制到sd卡上



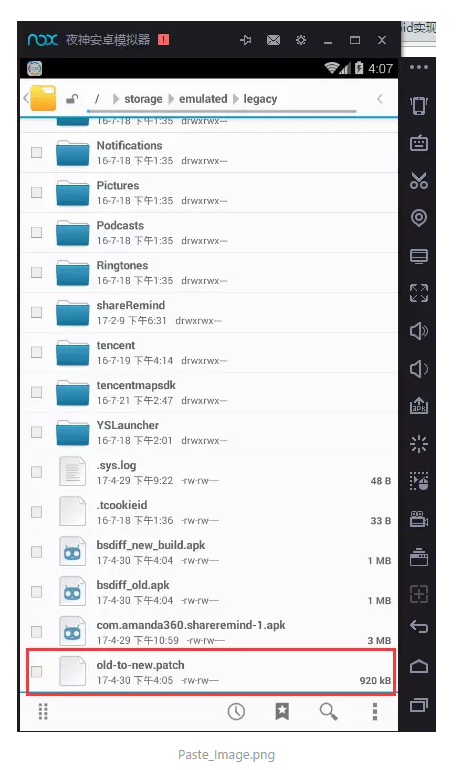
Paste\_Image.png

3、安装旧版本，效果图



Paste\_Image.png

4、差分生成增量更新包 old-to-new.patch



Paste\_Image.png

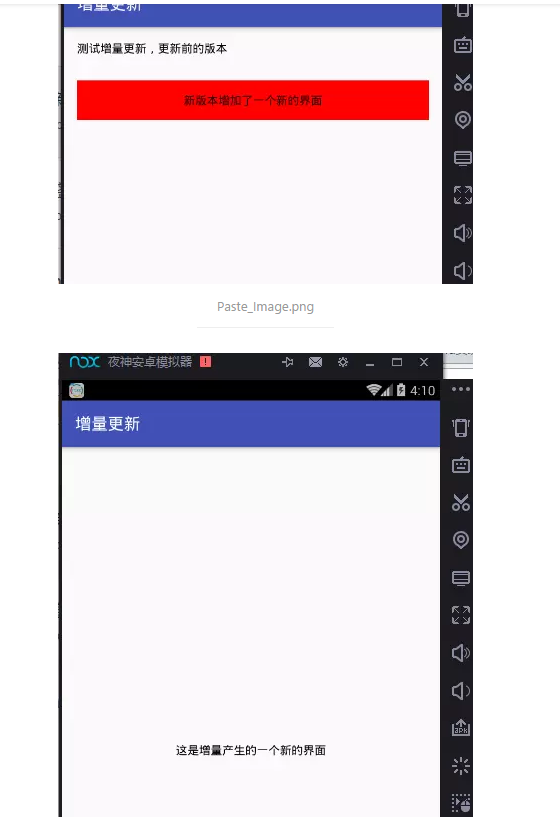
5、根据old-to-new.patch差分增量包生成新版本的安装包bsdiff\_new.apk

Paste\_Image.png



Paste\_Image.png

6、安装合成的新版本bsdiff\_new.apk

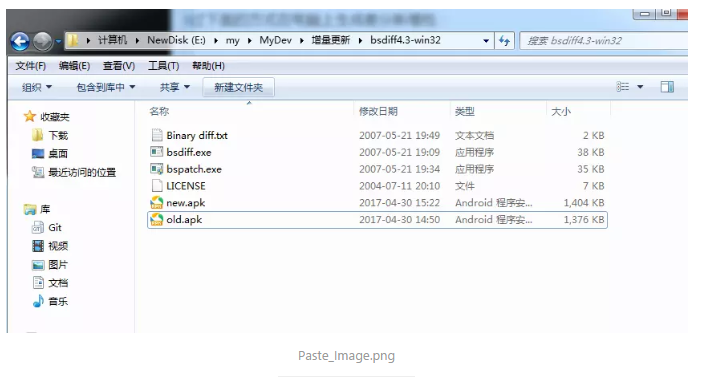


Paste\_Image.png

Paste\_Image.png

到此移动端的生成差分包和合成新版本就完成了。

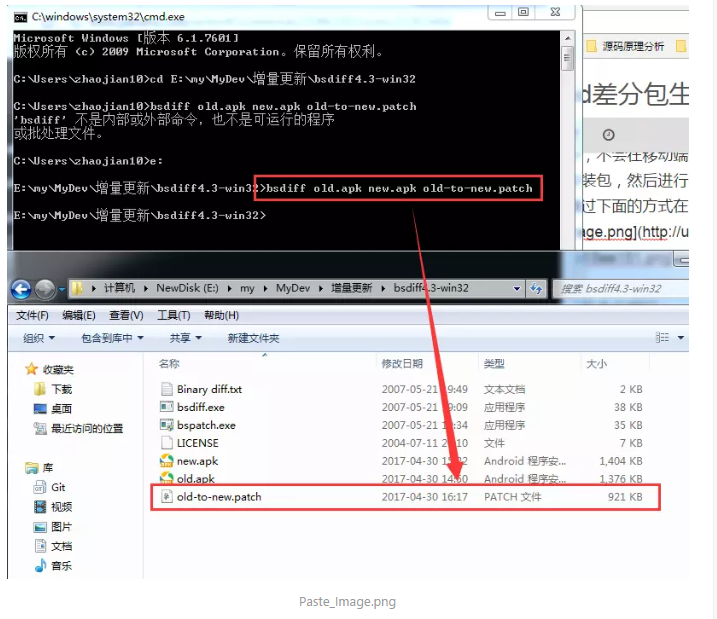
但是实际中，不会在移动端来生成差分更新包，而是生成差分包之后移动端下载，然后合成新的安装包，然后进行更新。  
所以可以通过下面的方式在电脑上生成差分新增包：  
工具下载：[http://download.csdn.net/detail/suijing/9829716](https://link.jianshu.com/?t=http://download.csdn.net/detail/suijing/9829716)



Paste\_Image.png

通过cmd打开命令窗口：  
进入到目录中  
（1）生成差量更新包文件old-to-new.patch：

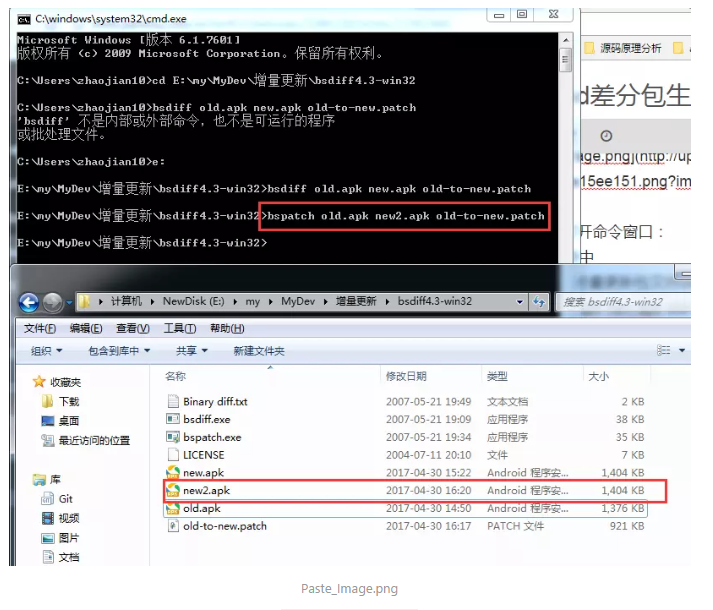
bsdiff old.apk new.apk old-to-new.patch  
这里生成的增量包 old-to-new.patch就可以用来跟老版本生成新的安装包了



Paste\_Image.png

（2）根据差量更新包old-to-new.patch生成新版本new2.apk ：

bspatch old.apk new2.apk old-to-new.patch



Paste\_Image.png