# [Apk 动态写入信息的方案](http://www.cnblogs.com/tmlee/p/5484790.html)

## 1. 背景

我们在日常使用应用可能会遇到以下场景。   
场景1： 用户浏览 H5 页面时，通过该页面下载链接下载安装 App，启动会来到首页而不是用户之前浏览的 H5 页面，造成使用场景的割裂。  
场景2： 用户通过二维码把一个页面分享出去，没有装猫客的用户如果直接安装启动后，无法回到分享的页面。

如果用户在当前页面下载了应用，安装之后直接跳转到刚才浏览的界面，不仅可以将这一部分流量引回客户端，还可以让用户获得完整的用户体验。下面提出一种方案来满足这个业务需求。

## 2. 原理

Android 使用的 Apk 包的压缩方式是 zip，与 zip 有相同的文件结构，在 zip 文件的末尾有一个 Central Directory Record 区域，其末尾包含一个 File comment 区域，可以存放一些数据，所以 File comment 是 zip 文件一部分，如果可以正确的修改这个部分，就可以在不破坏压缩包、不用重新打包的的前提下快速的给 Apk 文件写入自己想要的数据。

comment 是在 Central Directory Record 末尾储存的，可以将数据直接写在这里，下表是 header 末尾的结构。

从表中可以看到定义 comment 长度的字段位于 comment 之前，当我们从文件最后开始读取时，由于 comment 数据是不确定的，我们无法知道它的长度，从而也无法从 zip 中直接获取 Comment length。

这里我们需要自定义 comment，在自定义 comment 内容的后面添加一个区域储存 comment 的长度，结构如下图。

这里可以将一个固定的结构写在 comment 中，然后根据自定义的长度分区获取每个部分的内容，还可以添加其它数据，如校验码、版本等。

## 3. 测试代码：

### 写入apk

|  |
| --- |
| package com.writeapk;  import java.io.ByteArrayOutputStream;  import java.io.File;  import java.io.IOException;  import java.io.RandomAccessFile;  import java.io.UnsupportedEncodingException;  import java.net.URLEncoder;  import java.nio.ByteBuffer;  import java.nio.ByteOrder;  import java.util.zip.ZipFile;    /\*\*   \* 类说明：   \* @author  lizy   \* @date    2016-5-12   \* @version 1.0.0   \*/  public class WriteApk {      /\*\*       \* @param args       \*/      public static void main(String[] args) {          String filePath = "../app-debug.apk";          String comment = "Hello Comment!";          try {              comment = URLEncoder.encode(comment, "utf-8");          } catch (UnsupportedEncodingException e) {              // TODO Auto-generated catch block              e.printStackTrace();          }          writeApk(new File(filePath), comment);      }        public static void writeApk(File file,String comment){          ZipFile zipFile = null;          ByteArrayOutputStream outputStrean = null;          RandomAccessFile accessFile = null;            try {              zipFile = new ZipFile(file);                String zipComment = zipFile.getComment();              if(zipComment != null){                  return;              }              byte[] byteComment = comment.getBytes();              outputStrean = new ByteArrayOutputStream();              outputStrean.write(byteComment);              outputStrean.write(short2Stream((short)byteComment.length));                byte[] data = outputStrean.toByteArray();                accessFile = new RandomAccessFile(file, "rw");              accessFile.seek(file.length() -2 );              accessFile.write(short2Stream((short)data.length));              accessFile.write(data);            }catch (IOException e) {              e.printStackTrace();          }finally{              try {                  if(zipFile != null){                      zipFile.close();                  }                  if(outputStrean != null){                      outputStrean.close();                  }                  if(accessFile != null){                      accessFile.close();                  }              } catch (Exception e) {                  e.printStackTrace();              }            }          }        /\*\*       \* short转换成字节数组（小端序）       \* @param data       \* @return       \*/      private static short stream2Short(byte[] stream, int offset) {          ByteBuffer buffer = ByteBuffer.allocate(2);          buffer.order(ByteOrder.LITTLE\_ENDIAN);          buffer.put(stream[offset]);          buffer.put(stream[offset + 1]);          return buffer.getShort(0);      }        /\*\*       \* 字节数组转换成short（小端序）       \* @param stream       \* @param offset       \* @return       \*/      private static byte[] short2Stream(short data) {          ByteBuffer buffer = ByteBuffer.allocate(2);          buffer.order(ByteOrder.LITTLE\_ENDIAN);          buffer.putShort(data);          buffer.flip();          return buffer.array();      }  } |

### 读取Comment

|  |
| --- |
| Private static String showComment(String filePath) {  Try {  File file = new File(filePath);  ZipFile zipFile = new ZipFile(file);  String comment = zipFile.getComment();  zipFile.close();  return comment;  } catch (Throwable th) {  }  Return null;  } |