## 一. svn地址

Native 开发：

svn://192.168.24.144:8888/ott/wsijk/trunk

APK开发：

svn://192.168.24.144:8888/ott/wsijk/trunk/android/ijkplayer

## 二. svn上的源码与ijkplayer官方下载下来的源码差异地方：

1. 去掉android/contrib里面的ffmpeg, openssl的多个CPU平台的源码版本

2. 去掉ijkmedia/ijkyuv的编译源码

3. extra/ffmpeg的源码用git checkout ff3.0--ijk0.5.0--dev0.4.5--rc11 -B ijkplayer切换到ijkplayer版本的ffmpeg

4. extra/openssl的源码用git checkout OpenSSL\_1\_0\_2g -B ijkplayer 切换到ijkplayer的openssl版本

5. extra/libyuv的源码用git checkout ijk-r0.2.1-dev 切换到Ijkplayer的libyuv版本

6. 添加copy\_code.sh脚本，主要是拷贝ffmpeg, openssl, libyuv源码到新的不同CPU平台的目录

7. 添加build\_android.sh脚本，可以：

./build\_android.sh 默认编译armv7a CPU平台

./build\_android.sh armv5/armv7a/arm64/armx86/armx86\_64 可以选择对应的CPU平台进行编译

./build\_android.sh all/all64 编译全部CPU平台

./build\_android.sh all32 编译所有32位CPU平台

./build\_android.sh clean 清除编译

## 三. Native编译方法

编译环境搭建参考《Ijkplayer环境搭建.doc》。

如上面提到的，可以直接用build\_android.sh来直接编译，如果单独更改ffmpeg,openssl, libyuv的话，可以先执行对应的模块编译，再来编译整个ijkplayer。需要记住：

compile-openssl.sh

compile-ffmpeg.sh

compile-ijk.sh

由上往下编译顺序，openssl是以静态.a的库的生成方式，只要编译了某个模块，后面的几个编译脚本都需要执行一遍，才能生效（简单点可以直接执行build\_android.sh）

编译脚本后面可以再优化。

## 四. APK开发

Ijkplayer目录就是一个完整的Android Studio开发环境，直接把ijkplayer导入就可以。目录结构内容参考《ijkplayer源码目录结构说明.doc》