# Android平台插件说明

Android平台插件说明文档说明如下(以cnLive引擎分支为例)：

## 地图插件(MapView)

**平台实现**：com.wondertek.video.map

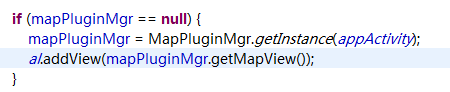
**Android组件注册**：无

**插件功能加载**：

1. 让AppActivity继承MapActivity；

d84dfd81.bmp

1. VenusActivity类sysInit()方法中创建对象并初始化：



1. VenusActivity类onResume()方法调用地图start()方法以开启相关监听：

81f2b945.bmp

1. VenusActivity类onPause()方法调用地图stop()方法以关闭相关监听：

149246c9.bmp

1. VenusActivity类onDestroy()方法调用地图destroyMap()方法释放资源：

7e99293d.bmp

**本地库(armeabi)**：

1. libminimapv320.so，高德矢量库，开启矢量图功能时必须加载；
2. libBMapApiEngine.so，百度地图本地库；

ps:自构建打包需要在build.xml中添加

<property name="native-dir" value="libs" />

<property name="native-dir-ospath" value="${basedir}/${native-dir}" />

<arg value="-nf" />

<arg value="${native-dir-ospath}" />

**依赖jar包：**

1. MapApi.jar，高德兼容包，此包是四个功能包的全集；
2. baidumapapi.jar，百度地图；
3. 高德地图MapApi.jar，可能会拆成以下四个包：
4. core.jar，高德地图核心包；
5. map.jar，地图相关功能包；
6. geocoder\_location.jar，地理编码与定位；
7. poisearch\_route.jar，POI搜索与路径搜索；

ps:自构建打包需要在build.xml中添加

<unzip src="${venusExternalJAR}/MapApi.jar" dest="${basedir}" />

<unzip src="${venusExternalJAR}/MapApi.jar" dest="${outdir-classes}" />

**依赖资源**：

见备注1；

**引擎插件接口**：位于引擎MapView目录；

**插件描述**：

在comrepository.xml里面添加如下记录：

<Component name="MapView" module="libmapview.so" path="WONDER:\\lib2\\mapview\\" enable="1" version="1.0.0" style="static" desc="地图组件" />

**引擎插件库**：

PJ/venus/lib2/mapview/libmapview.so

**业务框架接口**：

framework/map.lua

**备注**：

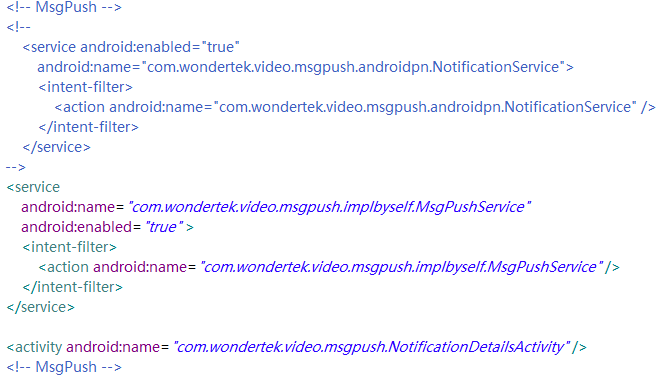
1. 引擎共实现两套地图，即高德地图与百度地图，分别在com.wondertek.video.map.gdmap与com.wondertek.video.map.bdmap包内；这两包中各自带有使用的资源，在选用其中一套地图时，请同步将包中资源复制到对应的资源目录或文件中，两套资源名都添加有gdmap或bdmap前缀，以方便管理；
2. 两套地图切换时，只需要更改MapPluginMgr类的MAP\_PLUGIN\_TYPE常量值，并在AppActivity中导入对应的MapActivity类，为节约空间计，在选用其中一套时，可以将别一套地图从工程中删除；
3. 此两套地图包中的资源在导入PJ资源目录或文件后，可以将这些资源删除；
4. 两套地图共用同一套业务接口；
5. 相关详情可参考网达论坛：

http://192.168.0.24:88/forum.php?mod=viewthread&tid=212&extra=page%3D1

## 消息推送(MsgPush)

**平台实现**：com.wondertek.video.msgpush

**Android组件注册**：

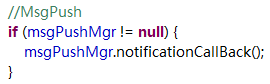


**插件功能加载**：

1. VenusActivity类sysInit()方法中创建对象并初始化：

66e7c417.bmp

1. VenusActivity类onResume()方法回调消息推送：



**本地库(armeabi)**：无

**依赖jar包**：

1. asmack.jar，Androidpn消息推送的依赖库；

**依赖资源**：

Drawable：notification.png；

**引擎插件接口**：位于引擎MsgPush目录；

**插件描述**：

在comrepository.xml里面添加如下记录：

<Component name="MsgPush" module="libmsgpush.so" path="WONDER:\\lib2\\msgpush\\" enable="1" version="1.0.0" style="static" desc="MsgPush"/>

**引擎插件库**：

PJ/venus/lib2/msgpush/libmsgpush.so

**业务框架接口**：

framework/msgpush.lua

**备注**：

1. 中间件引擎集成开发了两套消息推送机制，一个采用Androidpn开源项目，一个采用后台轮询请求数据实现，二者分别对应com.wondertek.video.msgpush.androidpn与com.wondertek.video.msgpush.implbyself。
2. Androidpn采用常连接的方式，服务端主动推送消息过来，能做到即推即收，时效性高，但大量常连接会给服务端带来巨大压力；后台轮询是客户端主动请求服务端数据，时效性决定轮询时机。
3. 两套机制之间切换，只需修改MsgPushManager类的MSG\_PUSH\_TYPE常量，并在AndroidMainfest.xml里注册对应的组件。
4. 选择其中之一后，可删除另外一套。两者共用同套业务代码。
5. 相关详情可参看网达论坛：

http://192.168.0.24:88/forum.php?mod=viewthread&tid=208&extra=page%3D1

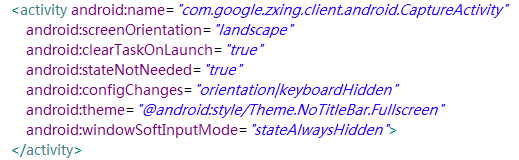
1. Androidpn用法也可参考如下邮件：



## 二维码扫描

**平台实现**：com.google.zxing.client.android

**Android组件注册**：



**插件功能加载**：

C层直接调用CaptureActivity的静态方法，实现Activity的切换；

**本地库(armeabi)**：无

**依赖jar包**：

1. core.jar，若高德地图若使用非兼容包，可能会冲突(高德地图也有core.jar包)，若如此，建议将core.jar更名为zxing.jar；

**依赖资源**：

layout：capture.xml

raw：beep.ogg

**引擎插件接口**：位于引擎BarcodeScanner目录；

**插件描述**：

在comrepository.xml里面添加如下记录：

<Componentname="BarcodeScanner"module="libbarcodescanner.so"path="WONDER:\\lib2\\barcodescanner\\" enable="1" version="1.0.0" style="static" desc="BarcodeScanner插件"/>

**引擎插件库**：

PJ/venus/lib2/barcodescanner/libbarcodescanner.so

**业务框架接口**：

framework/barcode.lua

## 应用管理(AppManager)

**平台实现**：com.wondertek.video.appmanager

**Android组件注册**：无

**插件功能加载**：

1. 在VenusActivity类的sysInit()方法创建对象：

3e8430d5.bmp

1. 在VenusActivity类中实现如下方法：

public boolean javaInstallAppEx(String apkPath)；

public boolean javaInstallAppSilent(String apkPath)

public boolean javaUnInstallApp(String packageName)

public boolean javaRunApp(String packageName)

public boolean javaCreateShortcut(String packageName)

public boolean javaRemoveShortcut(String packageName)

public boolean javaShortcutExists(String packageName)

public String javaGetStbId()

public String javaGetInstalledAppInfo()

public String javaGetInstalledAppInfoById(String id)

public boolean javaDealWithShare(String imagepath, String text, String subject, String title, String packagename, String activityname)

public boolean javaShareWebByWX(String url, String title, String imagePath, String desc, boolean isFriendCircle)

public boolean javaOpenFileByApplication(String fileType, String filePath)

**本地库(armeabi)**：无

**依赖jar包**：

1. libammsdk.jar，微信开放平台；

**依赖资源**：无

**引擎插件接口**：位于AppManager目录；

**插件描述**：

<Component name="MsgPush" module="libmsgpush.so" path="WONDER:\\lib2\\msgpush\\" enable="1" version="1.0.0" style="static" desc="MsgPush"/>

**引擎插件库**：

PJ/venus/lib2/appmanager/libappmanager.so

**业务框架接口**：

framework/ appmanager.lua

**备注**:

1. 应用管理插件集成了微信分享开放平台，不若不使用，可删除微信开放平台的相关内容，并将javaShareWebByWX函数实现置空即可；

## 讯飞语音

**平台实现**：com.wondertek.video.ifly

**Android组件注册**：无

**插件功能加载**：

1. 在VenusActivity类的sysInit()方法创建对象：

**本地库(armeabi)**：

1. libmsc.so
2. libvad-jni.so

ps:自构建打包需要在build.xml中添加

<property name="native-dir" value="libs" />

<property name="native-dir-ospath" value="${basedir}/${native-dir}" />

<arg value="-nf" />

<arg value="${native-dir-ospath}" />

**依赖jar包**：

1. voiceRecognizer.jar

**依赖资源**：无

**引擎插件接口**：位于VoiceInput目录

**插件描述**：

<Component name="voiceinput" module="libvoiceinput.so" path="WONDER:\\lib2\\voiceinput\\" enable="1" version="1.0.0" style="static" desc="语音" />

**引擎插件库**：

PJ/venus/lib2/voiceinput/ libvoiceinput.so

**备注**：

1. cnLive引擎分支讯飞语音是从无线城市项目同步过来的；
2. 无线城市讯飞语音是定制版本；

## 系统播放器

**平台实现**：com.wondertek.video.sysplayer

**Android组件注册**：

3f419e45.bmp

**插件功能加载**：

1. 在VenusActivity类的sysInit()方法创建对象：

25baeddf.bmp

**本地库(armeabi)**：无

**依赖jar包**：无

**依赖资源**：无

**引擎插件接口**：位于SysPlayer目录

**插件描述**：

<Component name=" SysPlayer" module=" libsysplayer.so" path="WONDER:\\lib2\\ sysplayer\\" enable="1" version="1.0.0" style="static" desc="系统播放器"/>

**引擎插件库**：

PJ/venus/lib2/sysplayer/libsysplayer.so

**业务框架接口**：

framework/nativeplayer.lua

**备注**：

1. 系统播放器是将Android的MediaPlayer封装入Service以支持后台播放；
2. 目前系统播放器仅支持音频播放；

# 附录1：移动代维定制插件

## 增强现实

**平台实现**：

1. 三方实现，源码SVN:

http://192.168.0.85/svn/mobile/PJ-397/code/trunk/dwmobile/code/TestArProject

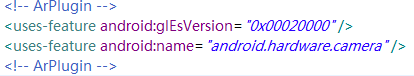
导入时，请将源码PJ的src全部导入工程；

1. 整合导入PJ实现：com.wondertek.video.arplugin
2. 将源码PJ的assets目录下的资源全部导入工程PJ的assets目录：
3. ahmobile.dat
4. ahmobile.xml
5. blue\_glow.png
6. green\_glow.png
7. Stones.dat
8. Stones.xml
9. texture\_android.png
10. 将源码PJ的相关res资源导入工程PJ的res对应目录；

**Android组件注册**：

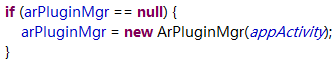


在根元素下添加：



**插件功能加载**：

1. 在VenusActivity类的sysInit()方法创建对象：



**本地库(armeabi)**：

1. libQCAR.so：高通Vorfria AR SDK库；
2. libDominoes.so：增强现实的实现库；

ps:自构建打包需要在build.xml中添加

<property name="native-dir" value="libs" />

<property name="native-dir-ospath" value="${basedir}/${native-dir}" />

<arg value="-nf" />

<arg value="${native-dir-ospath}" />

**依赖jar包**：

1. QCAR.jar

**依赖资源**：请参见源码PJ；

**引擎插件接口**：http://192.168.0.85/svn/mobile/PJ-397/code/trunk/dwmobile/code/ArPlugin

可放入cnLive引擎编译引擎插件库。

**插件描述**：

<Component name="ArPlugin" module="libarplugin.so" path="WONDER:\\lib2\\arplugin\\" enable="1" version="1.0.0" style="static" desc="ArPlugin"/>

**引擎插件库**：

PJ/venus/lib2/arplugin/libarplugin.so

**业务框架接口**：无

## Ums行为分析

**平台实现**：com.wbtech

**Android组件注册**：

在AndroidManifest.xml的Application标签下，添加如下meta-data

be86dedb.bmp

**插件功能加载**：无

**本地库(armeabi)**：无

**依赖jar包**：无

**依赖资源**：无

**引擎插件接口**：http://192.168.0.85/svn/mobile/PJ-397/code/trunk/dwmobile/code/UmsAgent

**插件描述**：

<Component name="UmsAgent" module="libumsagent.so" path="WONDER:\\lib2\\umsagent\\" enable="1" version="1.0.0" style="static" desc="Cobub Razor" />

**引擎插件库**：

PJ/venus/lib2/umsagent/umsagent.so

**业务框架接口**：

**备注：**

1. 请注意查阅http://192.168.0.85/svn/mobile/PJ-397/code/trunk/dwmobile/code/UmsAgent下的官方文档。

## Authentic

**平台实现**：com.wondertek.video. authentic

**Android组件注册**：无

**插件功能加载**：

1. 在VenusActivity类的sysInit()方法创建对象：

C:\Users\ADMINI~1\AppData\Local\Temp\msohtmlclip1\01\clip_image001.png

**本地库(armeabi)**：无

**依赖jar包**：无

**依赖资源**：无

**引擎插件接口**：位于引擎SafetyAuthentic目录；

**插件描述**：

<Component desc="安全认证插件" style="static" version="1.0.0" enable="1" path="WONDER:\\lib2\\safetyAuthentic\\" module="libsafetyauthentic.so" name="SafetyAuthentic"/>

**引擎插件库**：PJ/venus/lib2/ safetyAuthentic/libsafetyauthentic.so

**业务框架接口**：无