

千寻车载惯导 Android SDK

开发指南

V1.2.1

千寻位置网络有限公司

2016 年 12 月·上海

法律声明

版权所有© 2016，千寻位置网络有限公司。保留一切法律权利。本文档包含的所有内容除特别声明之外，版权均属于千寻位置网络有限公司所有，受《中华人民共和国著作权法》及相关法律法规和中国加入的所有知识产权方面的国际条约的保护。未经本公司书面许可，任何单位和个人不得以任何方式(电子或机械,包括影印)或理由对该文档或其包含的任何产品、服务、信息、材料的任何部分进行使用、复制、修改、抄录、传播或与其它产品捆绑使用、销售，否则将视为侵权，本公司必依法追究其法律责任。本文档并不代表供应商或其代理的承诺，千寻位置网络有限公司可在不作任何申明的情况下对本文档内容进行修改。本文档中提到的其它公司及其产品的商标所有权属于该商标的所有者。

千寻位置网络有限公司

联系邮箱： service@qxwz.com

官方网站： www.qxwz.com

目 录

1、 第一章 概述	5
1.1 产品简介	5
第二章 开发指南	6
2.1 开发准备	6
2.1.1 注册为开发者	6
2.1.2 创建应用	6
2.1.3 下载 SDK	6
2.2 接入流程图	6
2.3 配置工程	7
2.3.1 Jar 包引入	7
2.3.2 添加用户 Key	8
2.3.3 添加权限	8
2.3.5 配置结束	9
第三章 SDK 接口说明	10
3.1 SDK 接口说明	10
3.1.1 获取 WzVdrManager 对象	10
3.1.2 启动车载惯导	10
3.1.3 移除监听	11
3.1.4 获取 VDR 账号信息	11
3.1.5 关闭车载惯导	12

3.2 API 类说明	12
3.2.1 WzVdrFactory 类.....	12
3.2.2 VDR 操作类 WzVdrManager.....	13
3.2.3 VDR 启动类.....	13
3.2.4 接口及状态通知类 WzVdrListener.....	14
3.2.5 账号信息类 : WzVdrAccount.....	15
3.2.6 位置信息类 : WzVdrAccount.....	15
3.2.7 配置类 : QxwzInitParam.....	16
3.2.8 VDR code 码说明	17
 第四章 常见问题.....	 19
4.1 VDR 不能启动.....	19

历史版本	修改时间	作者	修改内容
V1.2.0	2016/11/09	俊逸	添加对外暴露 code
V1.2.1	2016/12/09	俊逸	添加 QxwzInitParam, 移 除部分对外暴露接口

1、第一章 概述

1.1 产品简介

千寻车载惯导 SDK (Android 版)是面向 Android 系统 5.0 及以上版本提供的开发包；是根据智能终端内置或者汽车前装 GNSS 芯片定位结果与 MEMS 惯性传感器 (陀螺仪，加速度计等) 的原始数据，通过松耦合的方式实现 GNSS / INS 的数据融合，从而达到组合定位效果的定位包；是第一个实现基于云、端一体化的惯导产品，本文简称为车载惯导 SDK。该产品致力于解决在没有卫星信号和网络信号覆盖情况下的定位问题，提升千寻定位类产品的综合能力。

千寻车载惯导 SDK (Android 版) 主要提供以下功能：

GPS 开启后，获取到当前的准确坐标并传给车载惯导引擎，根据当前传感器数据综合推算出运行中的当前坐标；当没有 GPS 数据时，根据当前持续的传感器数据和之前的位置信息推算出运行中的坐标。

千寻车载惯导 SDK 提供 iOS 和 Android 两个版本，本说明文档只针对 Android 版本的 SDK。

第二章 开发指南

2.1 开发准备

2.1.1 注册为开发者

开发者需要入驻千寻位置网，在千寻位置官网（qxwz.com）注册为千寻位置开发者，注册成功并通过个人/企业用户认证后，便可成为千寻位置网的开发者。详细信息请见千寻官网帮助文档。

（<https://www.qxwz.com/help-document-settled-platform.html#link-0>）

2.1.2 创建应用

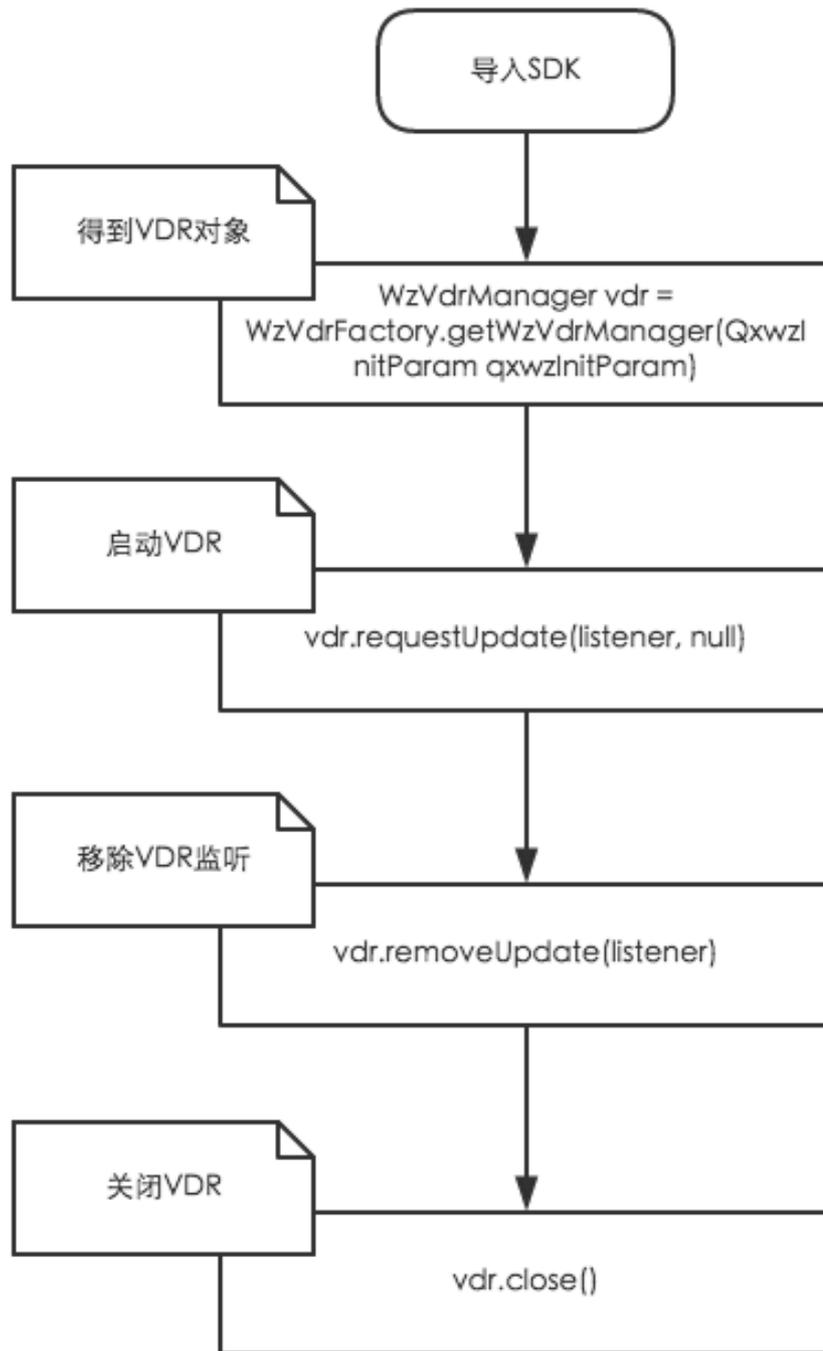
登录千寻位置网官网的开发者管理控制台创建应用，成功后会自动生成该应用的 AppKey 和 AppSecret；AppKey 和 AppSecret 是应用的凭证信息，请妥善保管。详细信息请见千寻官网帮助文档。

2.1.3 下载 SDK

在千寻位置网官网下载千寻车载惯导 SDK（Android 版）。

2.2 接入流程图

由于 API 接口调用有先后依赖关系，按照正常使用 API 调用顺序，整理出相关 API 的调用流程图，如下：



2.3 配置工程

2.3.1 Jar 包引入

1, 准备千寻提供的 SDK jar 包. 例如: "qxwz_vdr_v1.2.1.jar"

2, 将车载惯导 SDK 及相关 so 文件拷贝到 libs 根目录下, 拷贝完成的工程目录如下:



若您使用的是 Android Studio, 需要在 build gradle 中添加如下内容:

```
sourceSets {
    main {
        jni.srcDirs = []
        jniLibs.srcDirs = ['libs'] // include .so file into apk
        // reference: http://stackoverflow.com/questions/24357687/how-to-include-so-library-in-android-studio-really
    }
}
```

2.3.2 添加用户 Key

2.3.3 添加权限

由于SDK需要使用部分权限, 请在工程的“AndroidManifest.xml”文件中进行添加,

请直接拷贝以下示例代码:

```
<service android:name="com.qx.wz.sdk.control.WZLocationService"/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.BLUETOOTH"/>
```

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CHANGE_CONFIGURATION" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_LIFE_CARDS" />
<uses-permission android:name="android.permission.WRITE_LIFE_CARDS" />
<uses-permission android:name="android.permission.WAKE_LOCK"/>
```

android6.0 以上动态申请权限，需要开发者动态申请写 SD、GPS 定位、SENSOR 权限

2.3.5 配置结束

对整个工程进行 clean ，配置结束。

第三章 SDK 接口说明

3.1 SDK 接口说明

3.1.1 获取 WzVdrManager 对象

1, 接口定义

```
/**  
 * 此方法是单例模式，只会创建一次 WzdrManager。  
 * @param qxwzInitParam Refer to {@link QxwzInitParam}。  
 * @return WzVdrManager，VDR 对象，详细使用方式见 WzVdrManager 说明  
 * @throws Exception QxwzInitParam、context、appKey、appSecret、deviceId 为空时，会抛出异常出来，其  
它情况不会抛出异常  
 */  
public synchronized static WzVdrManager getWzVdrManager(QxwzInitParam qxwzInitParam)  
    throws Exception
```

2, 示例代码

```
//得到 VDR 对象  
String appKey = “从干寻官网申请 appKey” ;  
String appSecret = “从干寻官网申请 appSecret” ;  
String deviceId = “由用户提供” ;  
QxwzInitParm qxwzInitParam = new QxwzInitParam(context, appKey, appSercet, deviceId);  
WzVdrManager vdr = WzVdrFactory.getWzVdrManager(qxwzInitParam);
```

3.1.2 启动车载惯导

1, 接口定义

```

/**
 启动 VDR SDK。如果之前已经调用过此方法，此次调用则把 WzVdrListener 加入到 VDR SDK 监听队列里面。

 参数:
  listener - 通过它获取 VDR SDK 状态和运行结果。具体参见 WzVdrListener
  extras - 额外信息预留字段。目前直接填 null。
 抛出:
  java.lang.Exception - 当 listener 为空时，会抛出空指针异常。
 */

void requestUpdate(WzVdrListener listener,
                  java.util.Map<java.lang.String,java.lang.String> extras)
                  throws java.lang.Exception

```

2，示例代码：

```

// 示范: 启动车载惯导
vdr.start(listener, null);

```

3.1.3 移除监听

1，接口定义

```

/**
 移除监听。即使所有监听都移除了，VDR SDK 也不会关闭，要想关闭 VDR SDK 请调用 close()
 参数:
  listener - 想要移除的监听对象。
 抛出:
  java.lang.Exception
 */

void removeUpdate(WzVdrListener listener)
                  throws java.lang.Exception

```

2，示例代码：

```

// 示范: 启动车载惯导
vdr.removeUpdate(listener);

```

3.1.4 获取 VDR 账号信息

1，接口定义

```

/**
 获取 VDR 账号信息

```

```
*/  
WzVdrAccount getVdrAccount()
```

2, 示例代码 :

```
WzVdrAccount vdrAccount = getVdrAccount();
```

3.1.5 关闭车载惯导

1, 接口定义

```
/**  
 关闭 VDR SDK。  
 抛出:  
  java.lang.Exception  
 */  
 void close() throws java.lang.Exception
```

2, 示例代码 :

```
vdr.close();
```

3.2 API 类说明

3.2.1 WzVdrFactory 类

```
public class WzVdrFactory extends java.lang.Object
```

获取VDR对象,此类是单例模式。当调用 VDR close方法关闭VDR对象时,会释放VDR对象资源。

SDK获取安卓系统gps、sensor数据后传递给VDR算法推算出当前位置。

VDR SDK 需要使用以下权限 :

- (1) SD 卡写入权限 ;
- (2) 获取 gps 定位权限 ;
- (3) 获取 sensor 权限 ;

在 android 6.0 之前，开发者把这些权限写入 AndroidManifest.xml 里面，在用户同意安装后就可以正常获取相应权限；但在 android 6.0 之后，权限是应用运行时动态获取，用户需要在安装 VDR SDK 之前进行权限申请。

API 说明

限定符和类型	方法和说明
static WzVdrManager	<p>getWzVdrManager(QxwzInitParam qxwzInitParam)</p> <p>此方法是单例模式，只会创建一次 WzVdrManager。</p>

3.2.2 VDR 操作类 WzVdrManager

```
public interface WzVdrManager
```

3.2.3 VDR 启动类

限定符和类型	方法和说明
void	<p>close()</p> <p>关闭VDR SDK</p>
WzVdrAccount	<p>getVdrAccount()</p> <p>获取VDR SDK账号信息</p>
void	<p>removeUpdate(WzVdrListener listener)</p> <p>移除监听</p>
void	<p>requestUpdate(WzVdrListener listener,</p>

	<pre>java.util.Map<java.lang.String,java.lang.String> extras)</pre> <p>启动VDR SDK</p>
--	--

3.2.4 接口及状态通知类 WzVdrListener

```
public interface WzVdrListener
```

VDR SDK返回位置或者状态：

1, 返回位置, onLocation(com.qx.wz.algo_common.algorithm.WZLocation), 位置相关信息请见WZLocation

2, 返回VDR SDK状态, onStatus(int, String, Object), 状态码见3.2.8：

3, API说明

限定符和类型	限定符和类型
Void	<pre>onLocation(WZLocation location)</pre> <p>返回VDR吐出位置</p>
Void	<pre>onStatus(int status, java.lang.String message, java.lang.Object extras)</pre> <p>返回SDK状态信息 1002, AppKey认证失败, 请确保 appKey 和 appSecret 正确 1004, 网络不可用, 请确保网络正常, VDR SDK开始启动会请求网络 2008, 服务端内部错误, 碰到此问题请联系</p>

	debin.huang@wz-inc.com获取帮助
--	----------------------------

3.2.5 账号信息类 : WzVdrAccount

```
public class WzVdrAccount extends java.lang.Object
```

VDR SDK账号相关信息，返回appKey和手机deviceId

限定符和类型	方法和说明
java.lang.String	getAppKey() 返回appKey
java.lang.String	getDeviceId() 返回deviceId

3.2.6 位置信息类 : WzVdrAccount

```
public class WZLocation extends java.lang.Object
```

限定符和类型	方法和说明
double	getLatitude() 返回纬度
double	getLongitude() 返回经度
double	getAltitude() 返回高度
int	getNumOfAcc() 返回急加速次数

int	getNumOfDecel() 返回急减速次数
int	getNumOfSharpTurn() 返回急转弯次数
int	getNumOfSharpSwitch() 返回急变道次数
int	getNumOfOverSpeed() 返回超速次数

3.2.7 配置类：QxwzInitParam

```
public class QxwzInitParam extends java.lang.Object
```

限定符和类型	方法和说明
void	QxwzInitParam(context, appKey,appSecret,deviceId) 构造方法, 所有字段不能为空
Android.content.Context	getContext() 获取应用程序上下文
String	getAppKey()获取传入的 appKey
String	getAppSecret() 获取传入的 appSecret
String	getDeviceId() 获取传入的 deviceId

3.2.8 VDR code 码说明

code 码	说明
1002	App Key 验证失败
1003	APP KEY 验证成功
1004	网络不可用
1005	GPS 未打开
1021	加签库加载失败。请确保 libqxsignture.so 配置正确。
2008	服务端内部错误，碰到此问题请联系 debin.huang@wz-inc.com 获取帮助
8000	VDR 开始启动
8001	没有 GPS 权限
8002	没有 Sensor 权限
8003	没有 SD 卡写入权限
8004	得到 GPS 定位
8005	gps 2 分钟内没有定位
8006	没有加速传感器，使用 GPS 定位
8007	没有磁力计传感器，使用 GPS 定位
8008	没有陀螺仪传感器，使用 GPS 定位
8009	PDR 完成启动
8010	PDR 启动失败
8011	PDR 关闭成功

8012	PDR 关闭失败
8013	Service 绑定失败。请确保在 AndroidManifest.xml 里面注册 <service android:name=" com.qx.wz.sdk.control.WZLocationService" >
8014	VDR 关闭成功
8015	VDR 关闭失败
8016	VDR 完成启动
8017	VDR 启动失败

第四章 常见问题

4.1 VDR 不能启动

一般又以下几种原因：

1，没有网络。VDR 启动时需要从服务器获取配置，如果没有网络，无法启动 VDR，
返回 1004 错误码

2，AppKey、AppSecret 错误。AppKey 或 AppSecret 错误导致不能通过后台认证，
返回 1002

3，没有 GPS 权限。当 SDK 没有权限，不能从系统获取位置，导致不正常工作，返回
8005

4，GPS 未定位。SDK 需要 GPS 定位，若无定位则无法正常工作，返回 8005

5，必须在应用层的 AndroidManifest.xml 里面注册如下 Service:

```
<service android:name="com.qx.wz.sdk.control.WZLocationService"/>
```