RTMAP 定位SDK-Android概要设计文档

部门：LBS平台事业部

项目：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改内容 | 修改人 | 时间 |
| V1 | 初稿 | 张能元 | 2015/3/11 |
| V1.1 | 程序流程图修改 | 张能元 | 2015/4/3 |

目录

目录 3

软件概要设计说明书 4

1 Introduction简介 5

1.1 Purpose目的 5

1.2 Scope范围 5

1.2.1 Name软件名称 5

1.2.2 Function软件功能 5

1.2.3 Applications软件应用 5

2 High Level Design概要设计 5

2.1 Level 0 Design Description第0层设计描述 5

2.1.1 Software System Context Definition软件系统上下文定义： 5

2.2 Level 1 Design Description第一层设计描述 6

2.2.1 Decomposition Description分解描述 6

2.2.2 Interface Description 接口描述 7

软件概要设计说明书

Keywords关键词：

Abstract摘要：

List of abbreviations缩略语清单：

# Introduction简介

## Purpose目的

本文档主要面向技术管理相关人员以及和产品,相关技术人员.

## Scope范围

### Name软件名称

Rtmap SDK-Android离线版

### Function软件功能

Android 定位 SDK 是一套简单的LBS服务定位接口，通过定位 SDK 开发者可以迅速为应用程序实现定位功能。

### Applications软件应用

可单独使用定位 SDK，也可以结合智慧图地图 SDK 或其他地图 SDK。

# High Level Design概要设计

## Level 0 Design Description第0层设计描述

### Software System Context Definition软件系统上下文定义：

#### Design Methodology设计方法

#### Design Alternatives设计方案



智慧图Android定位SDK是利用设备当前的Wi-Fi信息（[Wi-Fi定位](http://developer.baidu.com/map/index.php?title=android-locsdk/qa#.E4.BB.80.E4.B9.88.E6.98.AFGPS.E5.AE.9A.E4.BD.8D.E3.80.81.E5.9F.BA.E7.AB.99.E5.AE.9A.E4.BD.8D.E5.92.8CWi-Fi.E5.AE.9A.E4.BD.8D.EF.BC.9F)）和蓝牙信息（iBeacon定位）完成定位的。定位SDK会根据设备当前的实际情况（如是否连接网络，是否扫描到Wi-Fi信息或beacon信息）进行定位。

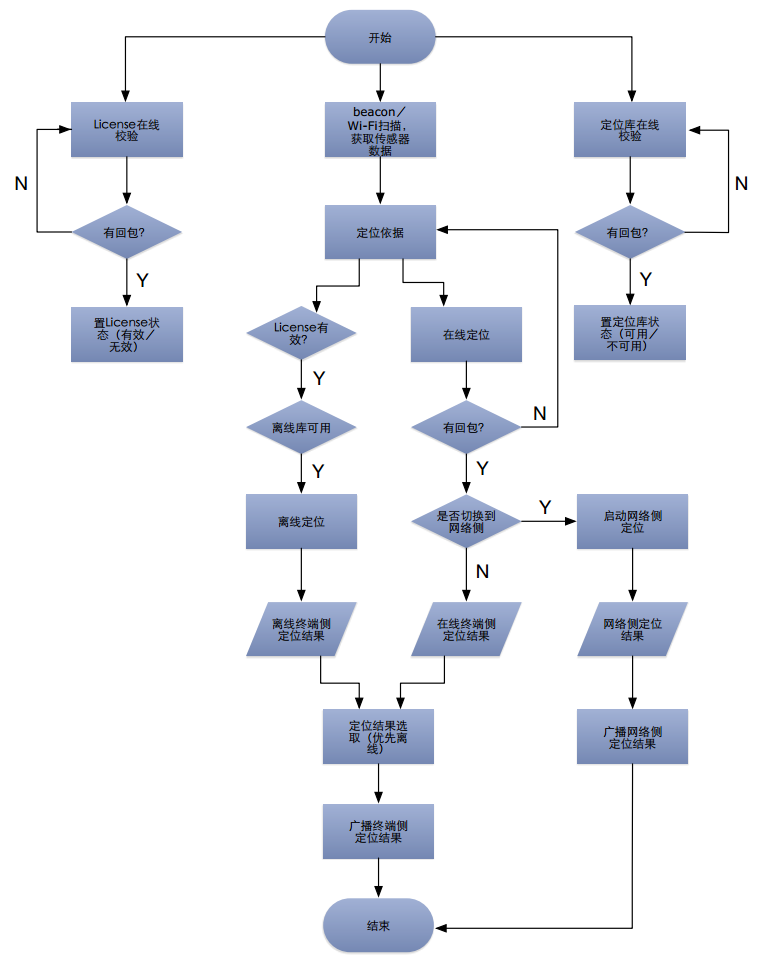
#### Design Constraints设计约束

|  |  |
| --- | --- |
| 定位模式 | 要求 |
| Wi-Fi定位 | Android2.2及以上 |
| 蓝牙定位（beacon） | Android4.3及以上，设备支持蓝牙4.0 |

#### Other Design Considerations其他

## Level 1 Design Description第一层设计描述

### Decomposition Description分解描述



#### Module/Subsystem Decomposition 模块/子系统分解

#### Concurrent Process Decomposition 并发进程处理分解

#### Module/subsystem Dependencies模块/子系统间的依赖关系

#### Process Dependencies 进程间依赖关系

#### Data Dependencies 数据依赖关系

### Interface Description 接口描述

LocationApp是一个单例，必须在主线程中声明，需要Context类型的参数。Context需要全进程有效的context,推荐用getApplicationConext。

**private** MyBroadcastReceiver myBroadcastReceiver;

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

// 定位服务初始化

LocationApp.*getInstance*().init(getApplicationContext());

// 注册广播，接收定位结果

myBroadcastReceiver = **new** MyBroadcastReceiver();

IntentFilter filter = **new** IntentFilter();

filter.addAction("BroadcastAction");

registerReceiver(myBroadcastReceiver, filter);

}

LocationApp是定位的SDK的核心类，具体方法如下：

获取SDK版本号：

public String getVersion();

获取so库版本号：

public String getLibsoVersion();

初始化：

public void init(Context context); // 须在主线程中申明

设置定位服务器地址：

public void setServerAddress(String ip, String port);

启动定位：

public void start();

停止定位：

public void stop();

实现MyBroadcastReceiver，接收定位SDK返回信息：

**class** MyBroadcastReceiver **extends** BroadcastReceiver {

@Override

**public** **void** onReceive(Context context, Intent intent) {

String action = intent.getStringExtra("ACTION");

**if** (action.equals("RTMAP\_LICENSE")) {

Log.*i*(***TAG***, "License Check");

} **else** **if** (action.equals("RTMAP\_LOCATE")) {

ResultEntityParcelable output = intent

.getParcelableExtra("LOCATE");

ResultEntity locResult = output.getPosition();

StringBuffer sb = **new** StringBuffer();

// 定位时间戳

sb.append("time : ");

sb.append(locResult.getTimestampPDR());

sb.append("error code : ");

sb.append(locResult.getError());

sb.append("build id : ");

sb.append(locResult.getBuildID());

sb.append("floor id : ");

sb.append(locResult.getFloorID());

sb.append("indoor x : ");

sb.append(locResult.getCoordX());

sb.append("indoor y : ");

sb.append(locResult.getCoordY());

sb.append("indoor accuracy : ");

sb.append(locResult.accuracy);

Log.*i*(***TAG***, sb.toString());

}

}

}

ResultEntity类封装了定位SDK的定位结果，具体如下：

获取定位时间戳：

public long getTimestampPDR(); // 服务器定位时间，单位秒

获取定位错误码：

public int getError();

返回值：

|  |  |
| --- | --- |
| 错误码 | 错误信息 |
| 0 | 室内定位结果可用 |
| 1001、1 | 定位数据问题 |
| 1004、3 | 传感器数据问题 |
| 3018、3020、4 | 服务器忙 |
| 3017、5 | 无当前建筑物定位数据 |
| 601 | License验证失败 |
| 其他 | 服务器问题 |

获取建筑物ID：

public String getBuildID();

获取楼层号：

public int getFloorID();

返回值：

第1位：1代表地下(B)，2代表地上(F)

2-4位：代表楼层值

第5位：0代表无夹层，1代表夹层

例如：20120代表F12，20011代表F1.5，10010代表B1

获取定位坐标：

public int getCoordX(); // 单位毫米（地图左上角为坐标原点）

public int getCoordY(); // 单位毫米（地图左上角为坐标原点）

获取定位精度：

public int getAccuracy(); // 单位米

#### Module/Subsystem Interface模块/子系统接口

#### Process Interfaces进程接口