

十分钟快速接入指引

本文档将帮助您快速接入大家都在使用的 MTA 统计功能，高级的功能接入请参考开发者手册。

1. 获取 AppKey

MTA 已经将开放平台的 appid 注册到系统上，开发者只需要在 appid 上添加前缀，即为 MTA 的 AppKey，其中规则如下表（大小写敏感）。

系统平台	前缀
Android	Aqc
iOS	Iqc

例如开放平台 appid 为 “123456” 的应用，对应于 MTA 的 Android AppKey 是 “Aqc123456”，iOS AppKey 是 “Iqc123456”。

2. 配置工程

- （1） 打开 eclipse，新建或选择现有的 Android 项目。解压下载的 MTA 压缩包，将其中 lib 目录下的 mta-sdk-x.x.x.jar 复制到应用工程库目录（通常名为 libs，如果不存在请新建一个）中。
- （2） 右键点击工程根目录→选择 Properties → Java Build Path →Libraries →点击 Add JARs...选中当前工程 libs 目录下的 mta-sdk-x.x.x.jar 文件，点击“OK”按钮即导入成功。如下图所示：

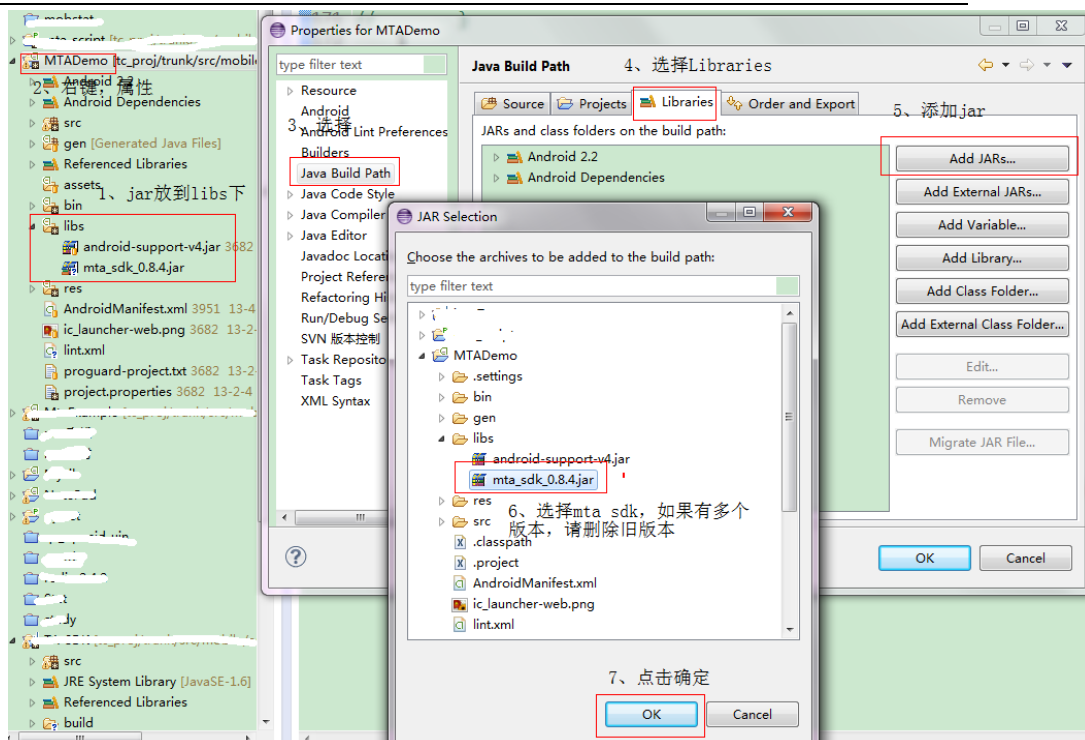


图 1 配置 MTA SDK 路径

(3) 双击工程目录下的 AndroidManifest.xml 文件,在打开的页面下面的标签点击

“AndroidManifest.xml”, 如下图所示:

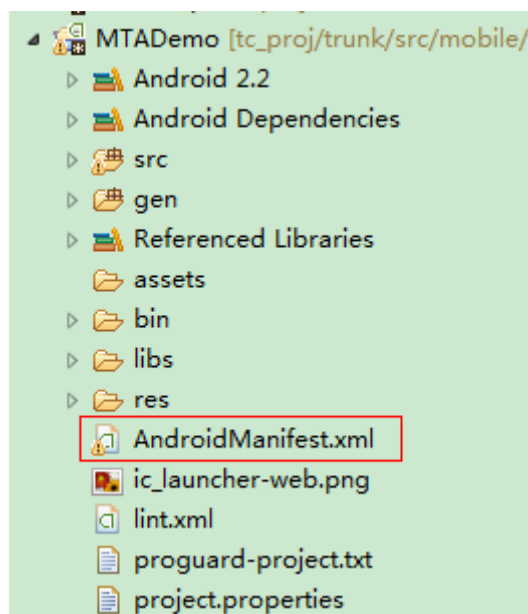


图 2 AndroidManifest.xml

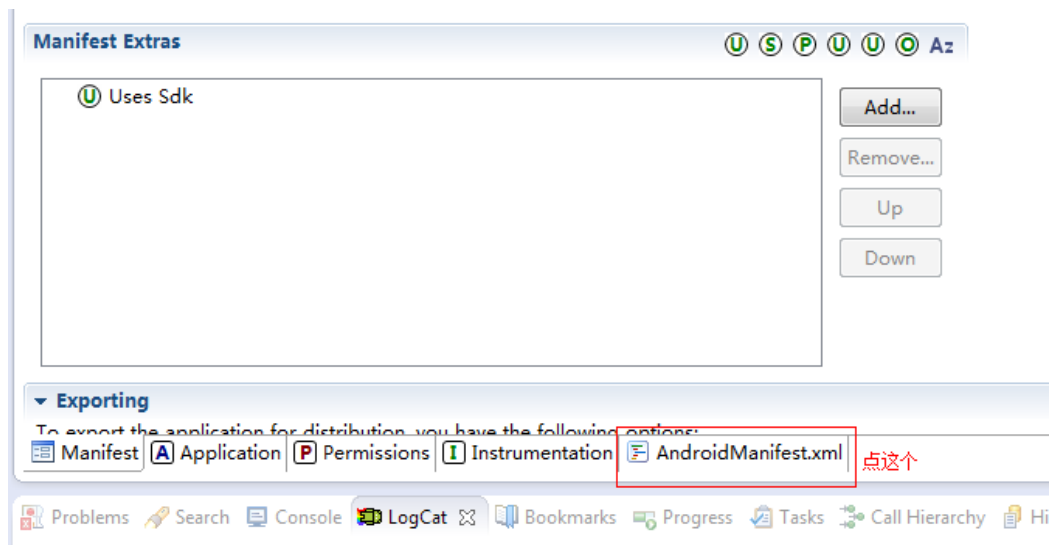


图 3 AndroidManifest.xml 编辑

- (4) 切换到源代码编辑界面，将 “</application>” 用以下内容替换。其中
“YOUR_APPKEY” 修改为开放平台 appid 加上 “Aqc” 前缀得到的 AppKey，
比如：“Aqc123456”，而 “YOUR_CHANNEL” 修改为发布的渠道（市场）。

```
<meta-data android:name="TA_APPKEY" android:value="YOUR_APPKEY" />
<meta-data android:name="InstallChannel" android:value="YOUR_CHANNEL" />
</application>

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
<meta-data
    android:name="TA_APPKEY" android:value="A6TPGR6K3V94" />
<meta-data
    android:name="InstallChannel" android:value="play" />
</application> 请注意标签
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

您的appkey

图 4 AndroidManifest.xml 配置

其中，AppKey 也可通过初始化代码设置。

```
String open_appid = "开放平台 appid";

StatConfig.setAppKey(this, "Aqc" + open_appid);
```

3. 嵌入 MTA 代码

```
<activity
    android:name="com.tencent.mtademo.MTAMainActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <!-- 指定应用程序最先启动的Activity -->
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name="com.tencent.mtademo.FirstActivity"
    android:label="@string/title_activity_first" >
</activity>
```

就是它了!

找到这2行标识

图 5 AndroidManifest 中找最先启动的 activity

切换到工程目录，双击 app 最先启动的 activity，即 MTAMainActivity。

双击 MTAMainActivity.java 源代码文件，在 onCreate 函数中输入以下 1 行 MTA

代码。

```
StatService.trackCustomEvent(this, "onCreate", "");
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_mtamain);
    // 打开debug开关, 可查看mta上报日志或错误
    StatConfig.setDebugEnable(true);
    // 调用统计接口, 触发MTA并上报数据
    StatService.trackCustomEvent(this, "onCreate", "");
```

图 6 嵌入 MTA 代码

4. 验证数据上报

成功嵌入 sdk 后, 在模拟器或手机运行您上面已经编译好的应用 (确保已执行 MTA 代码), 便会自动上报数据。

5. 还可以做什么

恭喜! 到这里您已经顺利接入MTA的统计功能, 需要获得更高级的服务功能, 请参考开发者手册。