**1、Spatialite for Android介绍**

Sqlite是一款短小精悍的数据库，在Android、IOS等都有支持。Spatialite是基于Sqlite的扩展，增加了对空间数据的支持，可以按照OGC的Simple Feature Access标准存取空间数据，可参考<http://www.gaia-gis.it/gaia-sins/spatialite-sql-3.0.0.html>。Spatialite for Android（<http://code.google.com/p/spatialite-android>）则是为Android平台提供的Spatialite实现，用于在Android平台上使用空间数据库。

Spatialite使用C++开发，Spatialite for Android包括两方面内容：使用Android NDK对Spatialite进行编译，生成.so库文件；在Android平台通过JNI调用相关接口，进行封装。

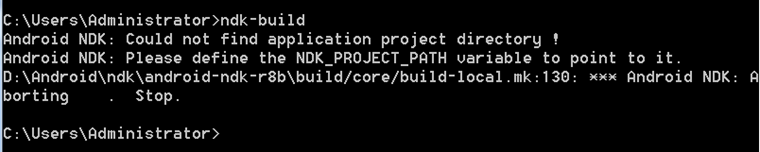
**2、Spatialite for Android编译**

**1）安装Cygwin**

目前，在单纯的Windows环境下还不能进行NDK编译，需要借助于Linux环境。Cygwin是一个为Windows操作系统提供Linux环境的软件。官方下载地址：http://cygwin.com/setup.exe，CSDN下载地址：<http://download.csdn.net/detail/arcgis_mobile/4737414>。具体安装过程不再说明。

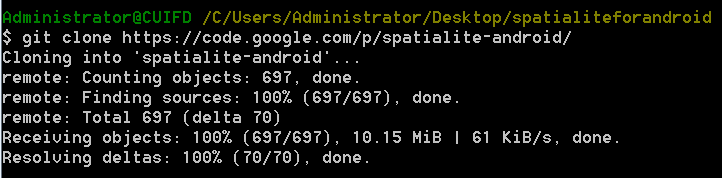
**2）安装Android NDK**

NDK需要单独安装，r8版本官网下载地址<http://developer.android.com/tools/sdk/ndk/index.html>。将NDK直接解压缩至某一目录，并将该目录添加到环境变量Path中。添加完成后，在cmd中运行ndk-build，如果出现下图的信息，则说明安装成功。



**3）下载Spatialite for Android**

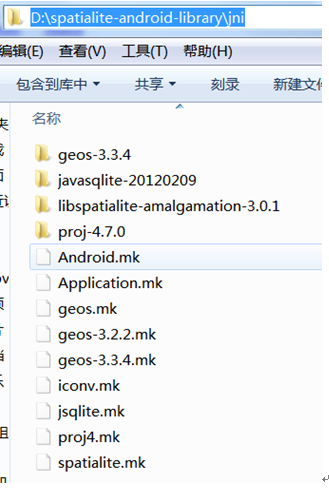
首先安装Git，官方下载地址：<http://git-scm.com/downloads>，CSDN下载地址：<http://download.csdn.net/detail/arcgis_mobile/4733383>。然后获取Spatialite for Android源代码，如下图所示：



Spatialite for Android源码包含三部分：spatialite-android，这是个测试工程；spatialite-android-library，这是核心内容，包含C/C++源码和Java接口封装，主要是编译这个工程；spatialite-android-test，包含一些单元测试。

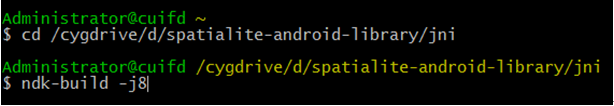
**4）下载geos-3.3.4和proj-4.7.0**

Spatialite for Android需要geos-3.3.4和proj-4.7.0。geos-3.3.4官网下载地址：<http://download.osgeo.org/geos/geos-3.3.4.tar.bz2> ，CSDN下载地址：<http://download.csdn.net/detail/gispace/4738742>。proj-4.7.0官网下载地址：<ftp://ftp.remotesensing.org/proj/proj-4.7.0.tar.gz> ，CSDN下载地址：<http://download.csdn.net/detail/gispace/4738736>。下载完成后，解压缩，将geos-3.3.4和proj-4.7.0两个文件夹拷贝到spatialte-android-library中的jni目录下，如下图所示。需要说明的是，spatialite-android-library所在的目录不能太深，最好放在某个盘符的根目录下，比如D盘的根目录，否则编译过程中会报出“参数列表太长”的错误。至此，编译前的准备工作已经全部完成。

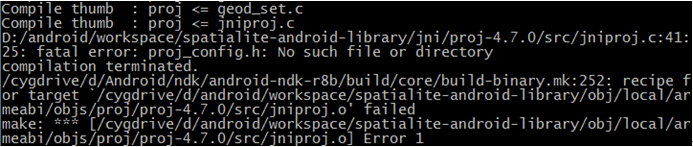


**5）编译Spatialite for Android**

打开Cygwin，进入spatialite-android-library中的jni目录，运行命令 ndk-build –j8。如下图所示。



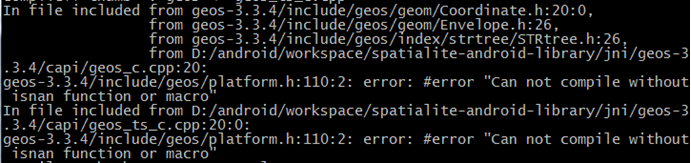
使用Cygwin编译，会遇到一些错误。第一个错误如下图所示，无法找到proj\_config.h文件。



解决办法是把proj-4.7.0/src/jniproj.c，第一行代码注释掉；然后重新编译。

http://img.my.csdn.net/uploads/201211/06/1352214925_9589.png

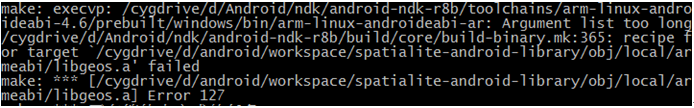
然后会遇到第二个错误，如下图所示。缺少对方法ISNAN的定义。



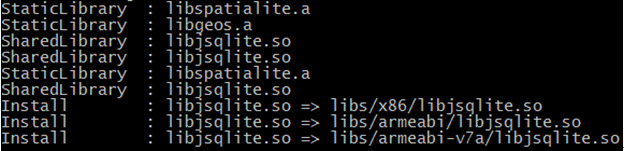
解决办法是为geos-3.3.4/include/geos/platform.h添加ISNAN的定义，如下图所示。重新编译。

http://img.my.csdn.net/uploads/201211/06/1352214988_4577.png

如果spatialite-android-library所处目录过深，则会遇到第三个问题，如下图所示。解决办法前面已经提到，将spatialite-android-library放到某个盘符的根目录下即可。重新编译。



如果一切顺利，在经过一段不短的时间的等待，支持三种cpu的so文件会最终编译成功，如下图所示。



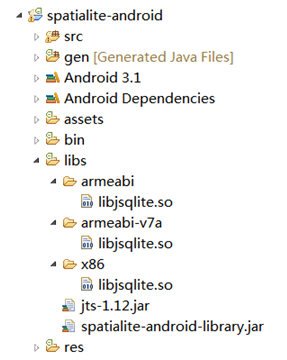
至此，编译过程结束。生成的so文件就可以在Android项目中使用了。

**3、在Eclipse中编译**

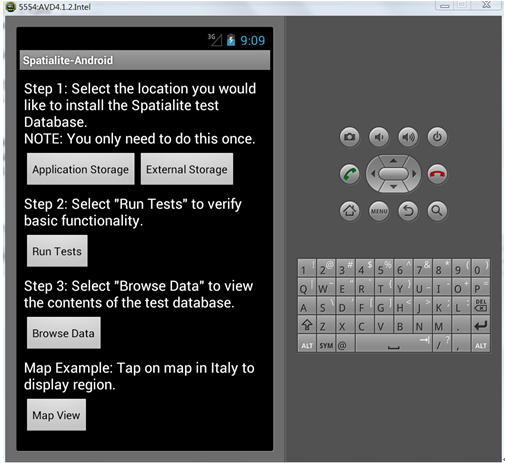
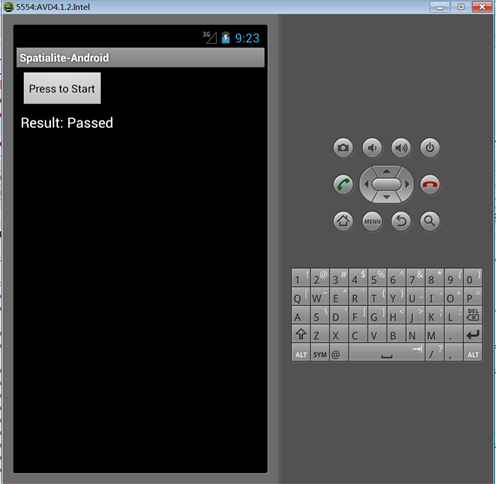
实际上，ADT-20.0.3提供了在Eclipse中进行NDK编译的插件，这样我们就可以在Eclipse中编译spatialite-android-library了。但是编译过程仍然需要Cygwin来提供Linux环境。实际上，在Eclipse中进行编译，依然是通过Cygwin的命令来完成的，这不是本篇博客的重点，所以不再详细介绍。在Eclipse编译的结果，除了三种so文件，spatialite-android-library中封装的Java代码，也会被打包成jar包。为方便大家使用，生成的jar包和so库文件已经上传到CSDN，大家下载后可直接在Android项目中使用了。下载地址：<http://download.csdn.net/detail/gispace/4738897>。

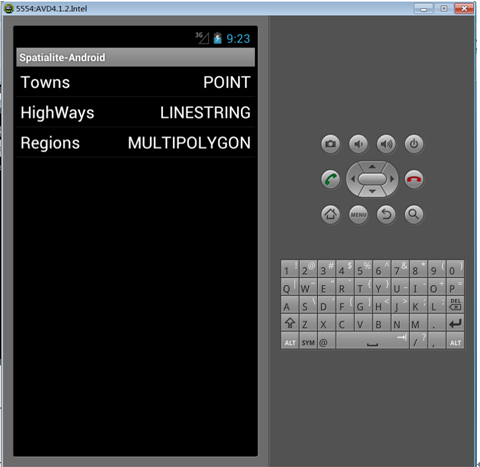
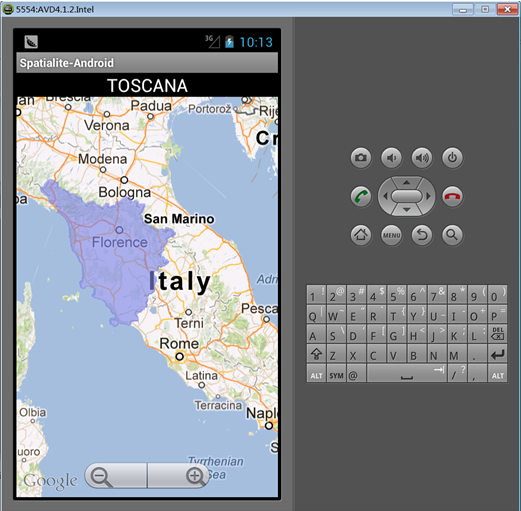
**4、运行测试程序**

有了编译好的jar包和so库文件，就可以运行spatialite-android工程来进行测试了。首先将spatialite-android导入到Android开发环境Eclipse中，新建libs目录，将编译好的jar包和so库文件放在libs文件夹中。spatialite-android默认依赖于spatialite-android-library，这里我们使用编译好的库文件和jar包，所以在build path中去掉对spatialite-android-library的依赖如下图所示。

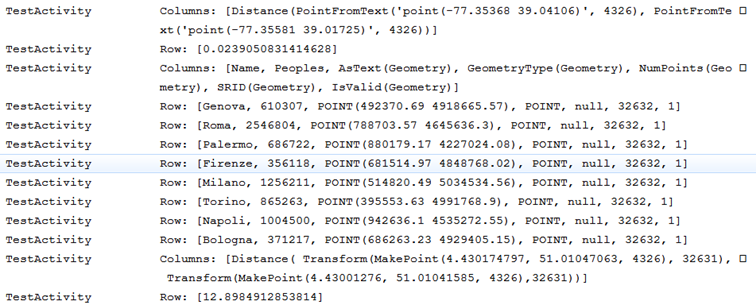


在模拟器上运行该程序的截图如下。

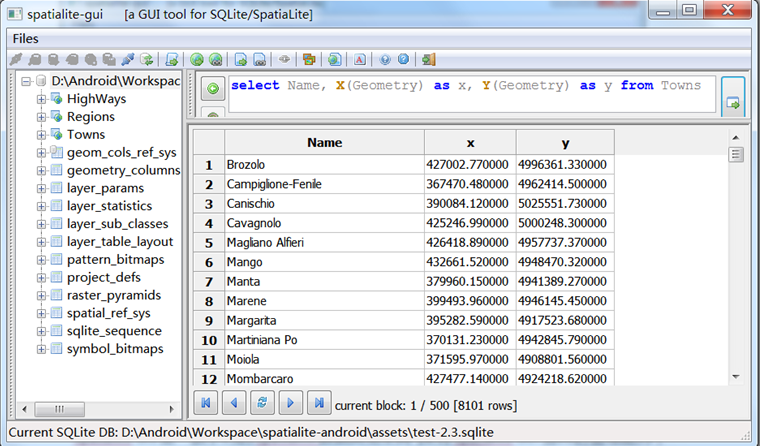
上图显示通过Google Map API渲染从Spatialite中读取的空间数据。需要说明的是，Google Map API需要根据debug.keystore的MD5指纹信息生成新的APIKey，具体请参考。<https://developers.google.com/android/maps-api-signup>。从LogCat输出的信息可以看到从spatialite读取的数据。



**5、Spatialite实用工具**

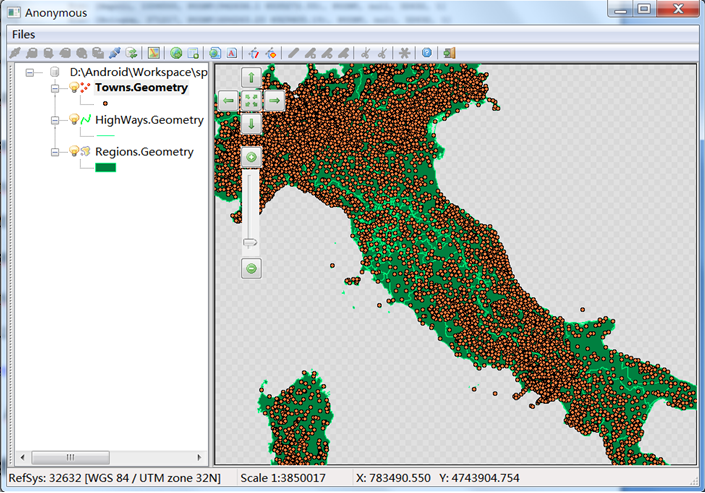
**1）Spatialite GUI**

Spatialiete GUI是一个Spatialite管理工具，可以新建、删除Spatialite数据库，通过SQL语句对数据表进行操作。下载地址：<http://download.csdn.net/detail/gispace/4739859>。应用截图如下图所示：



**2）Spatialite GIS**

Spatialite GIS是一款小巧的Spatialite数据浏览工具，没有Spatialite GUI那样的管理能力，但是可以对Spatialite中的空间数据进行渲染，方便浏览。下载地址：<http://download.csdn.net/detail/gispace/4739848>。应用截图如下图所示。



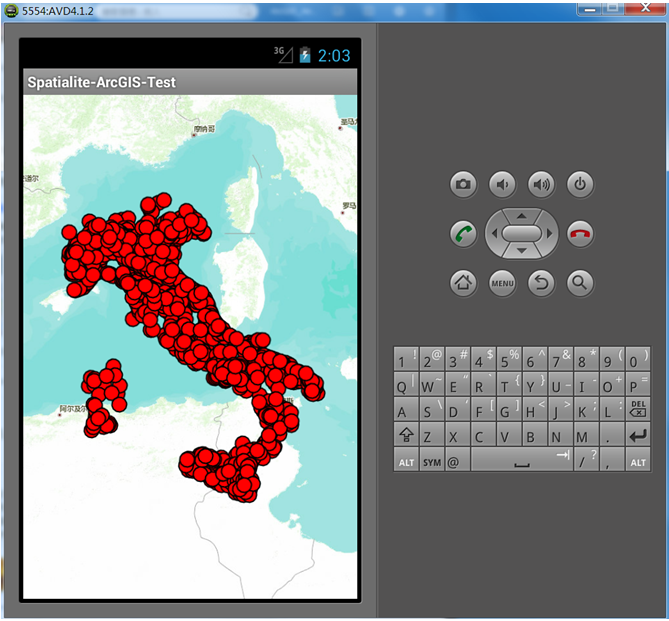
**6、在ArcGIS Android项目中测试Spatialite for Android**

在ArcGIS Android项目中使用Spatialite for Android其实非常简单，示例中把spatialite中的点数据，通过sql语句取出x、y值，在ArcGIS Android的GraphicsLayer已Graphic的形式显示即可。此处，只需要将之前编译好的so库和jar包放到项目中的libs文件夹下即可。源码下载地址：<http://download.csdn.net/detail/gispace/4740100>。核心代码如下所示：

**[java]** [view plaincopy](http://blog.csdn.net/gispace/article/details/8155503)

1. <span style="font-size:18px;">**public** **class** SpatialiteTestActivity **extends** Activity {
3. **private** MapView mapView;
4. **private** GraphicsLayer graphicsLayer;
5. **private** SimpleMarkerSymbol markerSymbol;
7. @Override
8. **public** **void** onCreate(Bundle savedInstanceState) {
9. **super**.onCreate(savedInstanceState);
10. setContentView(R.layout.main);
11. mapView = (MapView) **this**.findViewById(R.id.map);
12. **this**.mapView.addLayer(**new** ArcGISTiledMapServiceLayer("http://www.arcgisonline.cn/ArcGIS/rest/services/ChinaOnlineCommunity/MapServer"));
13. graphicsLayer = **new** GraphicsLayer();
14. **this**.mapView.addLayer(graphicsLayer);
16. // copy database to application folder
17. **try** {
18. AssetHelper.CopyAsset(**this**, ActivityHelper.getPath(**this**, **false**), getString(R.string.test\_db));
19. } **catch** (IOException e) {
20. ActivityHelper.showAlert(**this**, getString(R.string.error\_copy\_failed));
21. }
23. // symbol
24. markerSymbol = **new** SimpleMarkerSymbol(Color.RED, 16, SimpleMarkerSymbol.STYLE.CIRCLE);
25. markerSymbol.setOutline(**new** SimpleLineSymbol(Color.BLACK, 1));
27. //
28. loadDataFromSpatialite();
29. }
31. **private** **void** loadDataFromSpatialite() {
32. **try** {
33. String dbFile = ActivityHelper.getDataBase(**this**,
34. getString(R.string.test\_db));
35. Class.forName("jsqlite.JDBCDriver").newInstance();
36. jsqlite.Database db = **new** jsqlite.Database();
37. db.open(dbFile, jsqlite.Constants.SQLITE\_OPEN\_READONLY);
39. Callback cb = **new** Callback() {
40. @Override
41. **public** **void** columns(String[] coldata) {
42. }
44. @Override
45. **public** **void** types(String[] types) {
46. }
48. @Override
49. **public** **boolean** newrow(String[] rowdata) {
50. // 爸从spatialite中得到的空间数据在GrpahicsLayer中显示出来
51. **double** x = Double.valueOf(rowdata[2]);
52. **double** y = Double.valueOf(rowdata[3]);
53. Point pnt = **new** Point(x, y);
54. Graphic graphic = **new** Graphic(pnt, markerSymbol);
55. graphicsLayer.addGraphic(graphic);
56. **return** **false**;
57. }
58. };
60. // sql语句，取出x、y坐标
61. String query = "SELECT name, peoples, x(Geometry), y(Geometry) from Towns where peoples > 8000";
62. db.exec(query, cb);
64. } **catch** (IllegalAccessException e) {
65. e.printStackTrace();
66. } **catch** (InstantiationException e) {
67. e.printStackTrace();
68. } **catch** (ClassNotFoundException e) {
69. e.printStackTrace();
70. } **catch** (FileNotFoundException e) {
71. e.printStackTrace();
72. } **catch** (Exception e) {
73. e.printStackTrace();
74. }
75. }
77. @Override
78. **protected** **void** onDestroy() {
79. **super**.onDestroy();
80. }
82. @Override
83. **protected** **void** onPause() {
84. **super**.onPause();
85. mapView.pause();
86. }
88. @Override
89. **protected** **void** onResume() {
90. **super**.onResume();
91. mapView.unpause();
92. }
94. }</span>

在模拟器中运行截图如下。



**7、小结**

文章中介绍了如何使用Android NDK编译Spatialite，这也是Android开发中常用的一种方式。ArcGIS Runtime SDK for Android也是通过这种方式开发的。Spatialite for Android使得在Android设备上使用空间数据库成为可能。Spatialite可以通过sql语句的方式对空间数据进行操作，提供了很对针对空间数据的方法。目前ArcGIS还没有提供矢量数据离线编辑的方案，使用Spatialite和ArcGIS结合的方式实现矢量数据离线编辑，不失为一种选择。后面的博客会对这部分内容展开研究和介绍。