

UCMAP

开发手册

UCMAP

目 录

一、UCMAP简介.....	2
1.1 前言.....	2
1.2 预备知识.....	3
1.3 ucmmap简介.....	3
1.4 基本功能展示.....	5
1.5 公司简介.....	8
二、服务器部分.....	8
2.1 服务器端软件安装与配置.....	8
2.2 如何保护数据.....	10
2.3 如何添加数据.....	10
2.4 如何配置地图样式.....	12
2.5 如何发布WMS地图.....	14
三、客户端部分.....	15
3.1 配置UCMAP开发环境.....	15
3.2 利用UCMAP地图组件建立客户端应用程序.....	16
3.3 如何部署应用程序.....	19
3.4 客户端如何浏览服务器发布的地图.....	20
3.5 如何进行空间查询以及属性查询.....	21
3.6 编辑工具如何使用.....	26
四、GPS读取、解析、显示.....	27
4.1 GPS读取.....	27
4.2 GPS协议解析.....	28
4.3 GPS显示.....	28
五、license授权.....	28
5.1 免费授权.....	28
5.2 部署授权.....	28

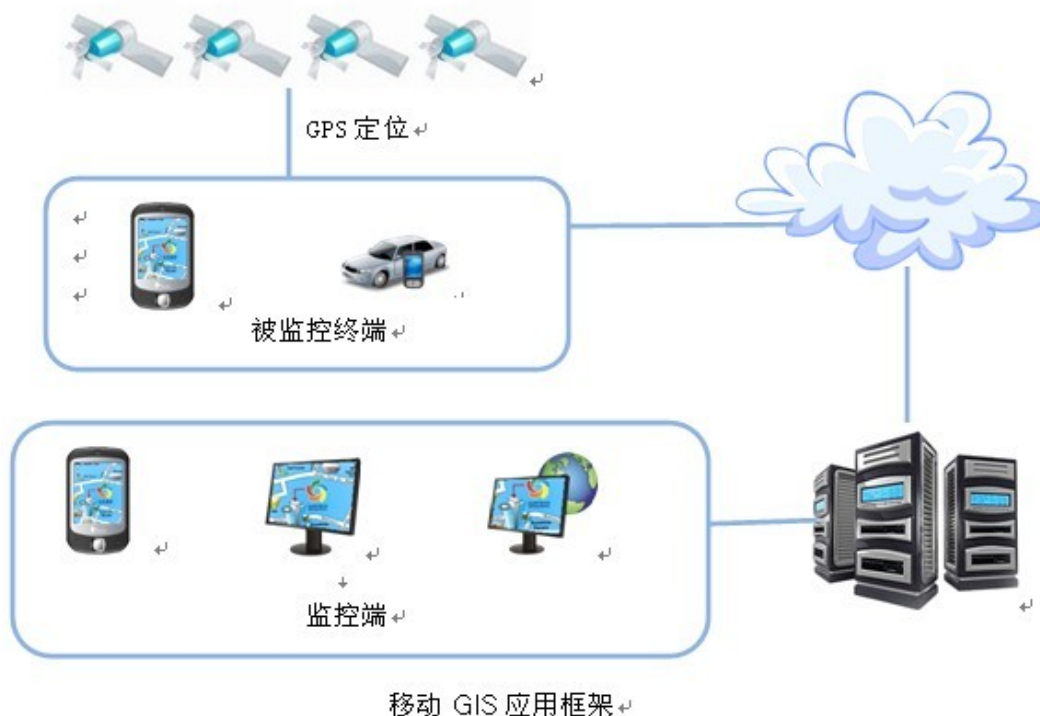
一、 UCMAP 简介

1.1 前言

移动 GIS, 是以移动互联网为支撑、以 GPS 智能手机为终端的 GIS 系统, 是继桌面 GIS、WEBGIS 之后又一新的技术热点, 移动定位、移动 MIS、移动办公等越来越成为企业或个人的迫切需求, 移动 GIS 就是其中的集中代表, 使得随时随地获取信息变得轻松自如。

随着 SOA 架构思想的渗透、OGC 标准的成熟, 移动 GIS 已成为互连、共享、互操作的开放系统, GIS 服务端对外发布 OGC 标准服务, 客户端按 OGC 的协议标准发送请求, 以调用服务端发布的服务, 获取所需要的结果。这种架构及标准化所带来的优势: 一, 屏蔽了服务端、客户端的软硬件差异, 用户不再关心服务端、客户端的具体实现, 不论服务端采用哪种 GIS 服务器, 也不论客户端是桌面程序、WEB 程序、还是手机程序, 只要服务端和客户端都遵守相同的协议标准, 就可以实现之间的交互操作; 二, 便于服务聚合, 实现数据共享与互操作, 为解决多系统的信息共享和交互操作, 需要解决系统与系统之间的接口问题, web services 技术给出了很好解决方案, 互相之间通过调用服务的方式来获取对方的信息, 方便了移动 GIS 系统本身以及与其他信息系统之间的交互与集成; 三, 分布式计算, web 服务分布于互联网不同的服务器上, 利用 Http+xml 实现交互通信, BPEL 整合业务逻辑, 实现计算的透明性和分布式。

以移动 GIS 为代表的移动互联网行业正快速发展, 移动 GIS 在各行业的应用正逐步展开, 实现随时随地的信息互通, 这将进一步提升行业信息化水平, 规范行业更好更快发展。



1.2 预备知识

移动 GIS，包括服务端和客户端，以及 OGC 相关标准。

服务端——主要是 J2EE 环境，包括 JDK 虚拟机环境、web 应用服务器 Tomcat、GIS 服务器 Geoserver、瓦片缓存服务器 Geowebcache、空间数据库以及各种扩展服务等。

客户端——主要是移动 GIS 客户端 ucmmap、windows mobile 相关开发等。

OGC 标准——ucmmap 已经支持了 WMS、WFS、WFS-T、WCS、WPS 以及 Filter Encoding 等标准，客户只需要调用相关接口即可，同时，开发者也可以依据 OGC 相关标准自行扩展，以实现相关功能。

1.3 ucmmap 简介

ucmmap 是一款面向 3G、支持 OGC 标准的移动 GIS 开发平台，延伸了 GIS 在手机端的应用，它采用先进的 SOA 架构，支持 WMS、WFS、WFS-T、WPS 等 OGC 标准，实现了无缝的 GIS 集成。

系统组成：

数据源：PostGIS 、 Shapefile 、 ArcSDE 、 Oracle 、 VPF 、 MySQL 、 MapInfo 等；

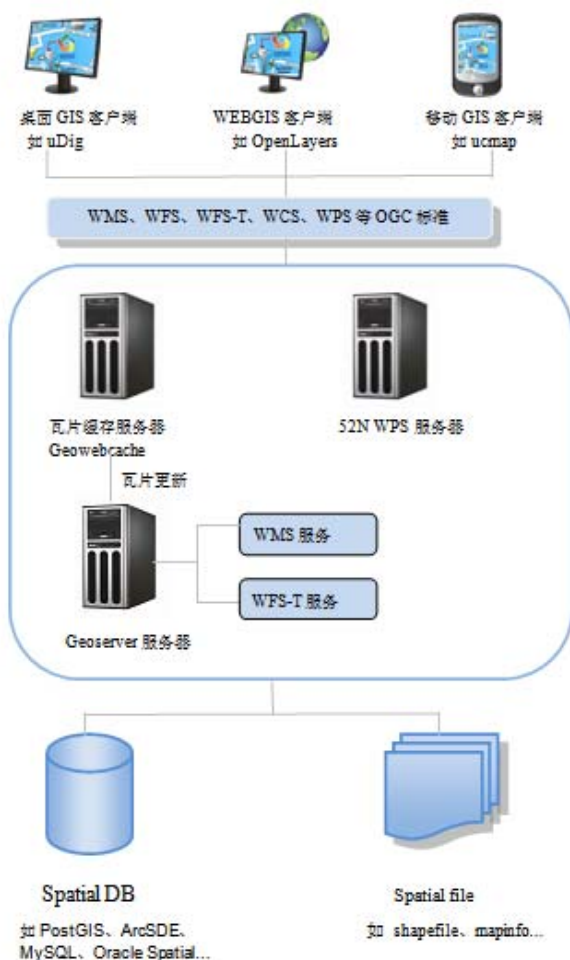
GIS 服务器：Geoserver、Mapserver、MapGuide、arcgis server、supermap iserver 等；

瓦片缓存服务器：Geowebcache；

手机客户端：ucmmap 手机客户端；

WEBGIS 客户端：Openlayers；

系统架构图



特点:

开放——采用 SOA 架构，以服务的方式对外发布 GIS 矢量和栅格数据，实现了空间数据和 GIS 功能的分布式存储、维护、分发、聚合和共享；

标准——遵守 OGC 标准，实现了 GIS 共享与互操作，进一步提升了服务的透明性，隐藏了系统软硬件的差异以及服务的具体实现细节；

3G——结合 3G、开放、标准的特点，实现与移动 MIS、移动 OA 等系统无缝集成，丰富了 3G、GIS、GPS 等新技术在各行业的广泛应用。

功能:

地图浏览，支持地图的放大、缩小、平移；

地图渲染，支持地图样式的配置、矢量要素的查询渲染；

地图查询，支持属性查询、空间查询、以及属性和空间的混合查询；

要素编辑，支持要素的添加、删除、修改，其中包括要素几何的节点编辑、属性编辑修改等；

数据缓存，支持瓦片的服务器缓存、瓦片更新、客户端缓存、以及矢量数据的本地存储；

影像叠加，支持遥感影像数据的叠加，并在其上采集、编辑数据；

数据同步，支持移动 GIS、WEBGIS、桌面 GIS 的数据同步，及时获取更新过的最新数据；

空间分析，支持各种空间查询分析、网络分析；
GPS 定位，支持获取手机 GPS 定位数据，实现 GPS 定位监控；
动态图层，支持动态图层，在底图上叠加动态要素点，比如 GPS 等动态刷新的点
扩展定制，高可扩展性，支持 GPS 语音导航，视频、图象等采集、显示、上传，与移动 MIS、移动 OA 的无缝集成，以及各种其他服务的组合；
环境配置要求：
服务端：Apache Tomcat 6.0、PostgreSQL 8.3、PostGIS 1.3.5、geoserver-1.7.3、geowebcache 1.1.5 及以上版本；
手机端：windows mobile 或 Symbian 操作系统。

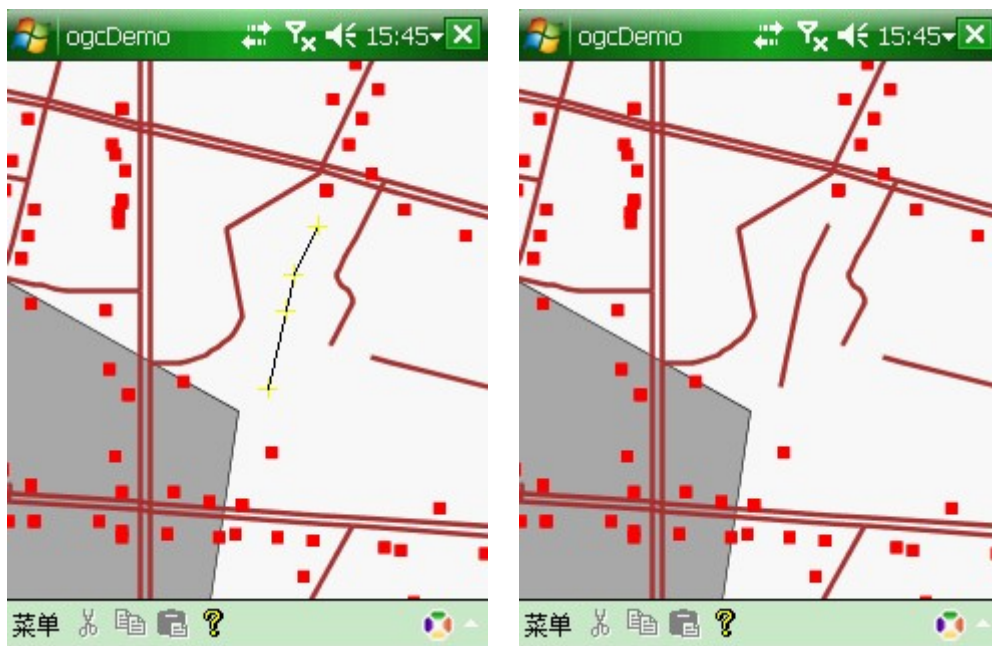
1.4 基本功能展示



地图浏览



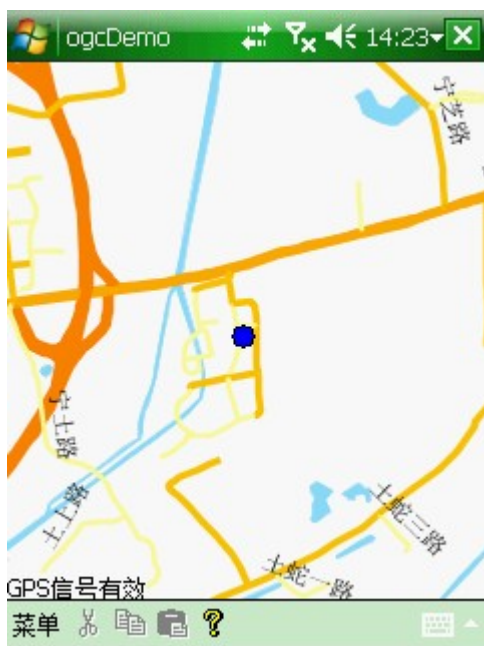
属性查询



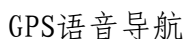
添加要素



影像浏览及要素编辑



GPS 监控



注：以上只是最基础的功能展示，由于 ucmap 是一款支持 OGC 标准移动 GIS 开发平台，开发者可以基于 ucmap 平台定制开发出符合 OGC 标准的、功能丰富的移动 GIS 应用，且可以将手机拍照、手机视频等多媒体元素集成进来，实现多功能集成的移动 GIS 应用。

南京跬步科技有限公司 (<http://www.creable.cn>), 是技术领先的移动 GIS 开发平台及解决方案提供商, 位于南京大学科技园。

跬步科技一直致力于自主知识产权的产品、专注于地理信息与移动互联网的应用与开发，积极推动移动 GIS 在各个行业的普及与应用。公司一直本着“以人为本，追求核心技术”的理念，与客户建立良好的合作关系，全程提供强有力的技术支持，为移动 GIS 在各行业的应用不懈努力。

二、服务器部分

2.1 服务器端软件安装与配置

首先需要先安装 JRE、Tomcat、postgresql(安装时必须输入一个密码,不能为空)、postgis。安装 geoserver 的步骤有点特殊,将从网络上下载的 geoserver.war 文件放到 Tomcat 的安装目录中的 webapps 文件夹下,如果您没有改变 Tomcat 的安装路径,那么这个文件夹就是 C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps。这时 Tomcat 会自动在

webapps 文件夹下生成 geoserver 文件夹。用写字板或者 word 打开 geoserver\WEB-INF 文件夹下的 web.xml 文件，将里面的所有“D:\gwc”替换成为您设置的瓦片文件夹名称。我公司网站上的 geoserver 已经集成了我们开发的组件，不用进行下面的配置，如果您使用官方的 geoserver，那么需要进行下面的配置工作。

从我公司网站下载 filter.rar 文件，解压缩到 geoserver\WEB-INF\classes 文件夹下。

用写字板或者 word 打开 geoserver\WEB-INF 文件夹下的 web.xml 文件，在第五行 <description>Web Mapping, Feature and Coverage Services</description>下面添加如下内容

```
<context-param>
  <param-name>GWC_METASTORE_DISABLED</param-name>
  <param-value>TRUE</param-value>
</context-param>
<context-param>
  <param-name>GEOWEBCACHE_CACHE_DIR</param-name>
  <param-value>D:\gwc</param-value>
</context-param>
<!-- The following only affects GeoServerConfiguration.class -->
<context-param>
  <param-name>GEOSERVER_WMS_URL</param-name>
  <param-value>http://localhost:8080/geoserver/wms?request=GetCapabilities</param-value>
</context-param>
```

其中 D:\gwc 是瓦片文件使用的文件夹，您可以根据你的需要，改变这个值。

在所有的<filter>之前添加如下内容

```
<filter>
  <filter-name>Creable Refresh Filter</filter-name>
  <filter-class>cn.creable.RefreshFilter</filter-class>
  <init-param>
    <param-name>GEOWEBCACHE_CACHE_DIR</param-name>
    <param-value>d:\gwc</param-value>
  </init-param>
</filter>
<filter>
  <filter-name>Creable WFS-T Filter</filter-name>
  <filter-class>cn.creable.WFSTFilter</filter-class>
  <init-param>
    <param-name>GEOWEBCACHE_CACHE_DIR</param-name>
    <param-value>d:\gwc</param-value>
  </init-param>
</filter>
```

其中 d:\gwc 是瓦片文件使用的文件夹，务必要与您上面设置的一致。（有两处 d:\gwc 需要修改）

在所有<filter-mapping>之前添加如下内容

```
<filter-mapping>
  <filter-name>Creable Refresh Filter</filter-name>
```

```
<url-pattern>/gwc/service/wms/*</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>Creable WFS-T Filter</filter-name>
  <url-pattern>/wfs/*</url-pattern>
</filter-mapping>
```

2.2 如何保护数据

geoserver 具有一个默认的管理员帐户，这个帐户用于管理您服务器上的数据，您可以修改 geoserver\data\security\users.properties 文件中的用户名和密码，用记事本打开这个文件，找到 admin=，修改为 admin=×××，其中×××为您设置的密码。

网络数据如果不加限制，会被其他非授权人员修改，geoserver 支持数据保护。首先您要新建一个帐户，用写字板打开 geoserver\data\security\users.properties 文件，在最后添加一行 username=password, ROLE_WFS_WRITE，这样就添加了一个名为 username，密码为 password，角色的 ROLE_WFS_WRITE 的用户。再用写字板打开 geoserver\data\security\service.properties 文件，在最后添加一行 wfs.Transaction=ROLE_WFS_WRITE，即指定只有角色为 ROLE_WFS_WRITE 的用户才能对数据进行添加、修改、删除的工作。用户想要修改数据，必须先输入用户名和密码，这样就达到保护数据的目的。

2.3 如何添加数据

geoserver 中一个 FeatureType 对应一个地图的图层，同一个 FeatureType 里的所有要素的几何类型都相同，即不允许点和线同层等类似情况出现。FeatureType 中规定着这个图层的样式范围等信息，创建它之前，首先要创建“数据库”，也就是数据源。geoserver 支持许多不同类型的数据，下面以 shapefile 和 postgis 空间数据库为例说明如何添加数据。

在添加数据之前，您需要新建一个名字空间，具体过程如下：

- ① 使用IE浏览器进入<http://localhost:8080/geoserver/>页面。
- ② 点击“配置”，并输入之前您设置的用户名和密码登陆。
- ③ 再点击“数据”、“名字空间”、“新建”。
- ④ 输入“前缀”，即名字空间的名字。
- ⑤ 输入 url，例如可以输入贵公司的网址。（前缀和 url 在今后的客户端 xml 配置中需要用到）。
- ⑥ 按下窗口左侧的“应用”、“保存”两个按钮。

如何添加 shapefile 类型数据到 geoserver？

- ① 首先将××.shp ××.shx ××.dbf 文件放入 geoserver\data\data 文件夹中。
- ② 使用IE浏览器进入<http://localhost:8080/geoserver/>页面。
- ③ 点击“配置”，并输入之前您设置的用户名和密码登陆。

- ④ 再点击“数据”、“数据库”、“新建”，Feature 数据描述选择 Shapefile，Feature 数据集 ID 输入这个 shapefile 数据源的名称。
- ⑤ 之后出现“Feature 数据集编辑器”界面，名字空间选择您之前设置的名字空间，url 中输入 file:data/×××.shp，charset 输入 GB2312。按下“提交”按钮。
- ⑥ 按下窗口左侧的“应用”、“保存”两个按钮。
- ⑦ 点击“配置”、“数据”、“FeatureTypes”、“新建”。FeatureType 名选择×××::×××，×××为您上面输入的 Feature 数据集 ID，按下“新建”按钮。
- ⑧ 之后出现“FeatureType 编辑器”界面，样式选择一个您设定的样式（样式如何制作在后面将详细介绍）。SRS（空间引用系统）输入数据的坐标系，如果为 WGS84 那么就输入 4326。按下边界盒状模型旁边的“生成”按钮，即可自动生成改 shapefile 的范围，您可以手动修改这些值。按下页面最下方的“提交”按钮。
备注：关于 SRC 空间引用系统的定义，您可以打开 <http://localhost:8080/geoserver/config/data/srsHelp.do> 查询您需要的编号。
- ⑨ 按下窗口左侧的“应用”、“保存”两个按钮。

在添加 postgis 数据到 geoserver 之前，您需要将数据导入到 postgis 数据库中，如何将 shapefile 数据导入 postgis 数据库？操作如下：

- ① 生成 sql 脚本。使用 shp2pgsql 程序生成 shp 文件对应的 sql 脚本。假定 PostgreSQL 安装路径为默认，那么单击 windows 开始菜单、运行、输入 cmd、回车。输入命令行"C:\Program Files\PostgreSQL\8.3\bin\shp2pgsql" -W "GBK" D:\×××.shp ××× > D:\×××.sql 其中×××为 shp 文件名，盘符根据你 shp 文件所在位置设置。
- ② 运行步骤 1 生成的 sql 脚本，导入数据到数据库。使用 psql 程序运行脚本。假定 PostgreSQL 安装路径为默认，那么单击 windows 开始菜单、运行、输入 cmd、回车。输入命令行"C:\Program Files\PostgreSQL\8.3\bin\psql" -d postgis -f D:\×××.sql postgres 其中×××为 sql 脚本文件名，postgis 为数据库名。之后提示您输入数据库密码，输入密码之后，按下回车，自动将数据导入数据库。

如何添加 postgis 数据到 geoserver？

- ① 使用IE浏览器进入<http://localhost:8080/geoserver/>页面。
- ② 点击“配置”，并输入之前您设置的用户名和密码登陆。
- ③ 再点击“数据”、“数据库”、“新建”，Feature 数据描述选择 Postgis，Feature 数据集 ID 输入这个 postgis 数据源的名称。
- ④ 之后出现“Feature 数据集编辑器”界面，名字空间选择您之前设置的名字空间，host 输入 Postgis 数据库所在服务器的 ip 地址，如果是本机，那就输入 localhost，port 输入 postgis 数据库的端口，默认为 5432，database 输入 postgis 数据库的名称，一般为 postgis，这取决与您安装 postgis 时设定的名字。user 输入默认的 postgres。passwd 输入密码。按下“提交”按钮。
- ⑤ 按下窗口左侧的“应用”、“保存”两个按钮。
- ⑥ 点击“配置”、“数据”、“FeatureTypes”、“新建”。FeatureType 名选择 AAA::BBB，其中 AAA 为您上面输入的 Feature 数据集 ID，BBB 为 postgis 数据库中的一个表的名字，这个表应该是您之前导入的。
- ⑦ 之后出现“FeatureType 编辑器”界面，样式选择一个您设定的样式（样式如何制作在后面将详细介绍）。SRS（空间引用系统）输入数据的坐标系，如果为 WGS84 那么就输入 4326。按下边界盒状模型旁边的“生成”按钮，即可自动生成改 shapefile 的范围，您可以手动修改这些值。按下页面最下方的“提交”按钮。

⑧ 按下窗口左侧的“应用”、“保存”两个按钮。

2.4 如何配置地图样式

地图如何绘制，由图层样式控制，在添加 FeatureType 的时候，提供您选择样式的机会，添加完毕之后，仍然可以通过编辑 FeatureType 的方式修改样式。样式其实都是一个个 xml 文件，如下所示一个典型的例子，这个是 geoserver 自动的 line 样式，指定采用 1 像素#0000FF 颜色来绘制线。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sld
http://schemas.opengis.net/sld/1.0.0/StyledLayerDescriptor.xsd">
  <NamedLayer>
    <Name>Default Line</Name>
    <UserStyle>
      <Title>1 px blue line</Title>
      <Abstract>Default line style, 1 pixel wide blue</Abstract>

      <FeatureTypeStyle>
        <!--FeatureTypeName>Feature</FeatureTypeName-->
        <Rule>
          <Title>Blue Line</Title>
          <Abstract>A 1 pixel wide blue line</Abstract>
          <LineSymbolizer>
            <Stroke>
              <CssParameter name="stroke">#0000FF</CssParameter>
            </Stroke>
          </LineSymbolizer>
        </Rule>

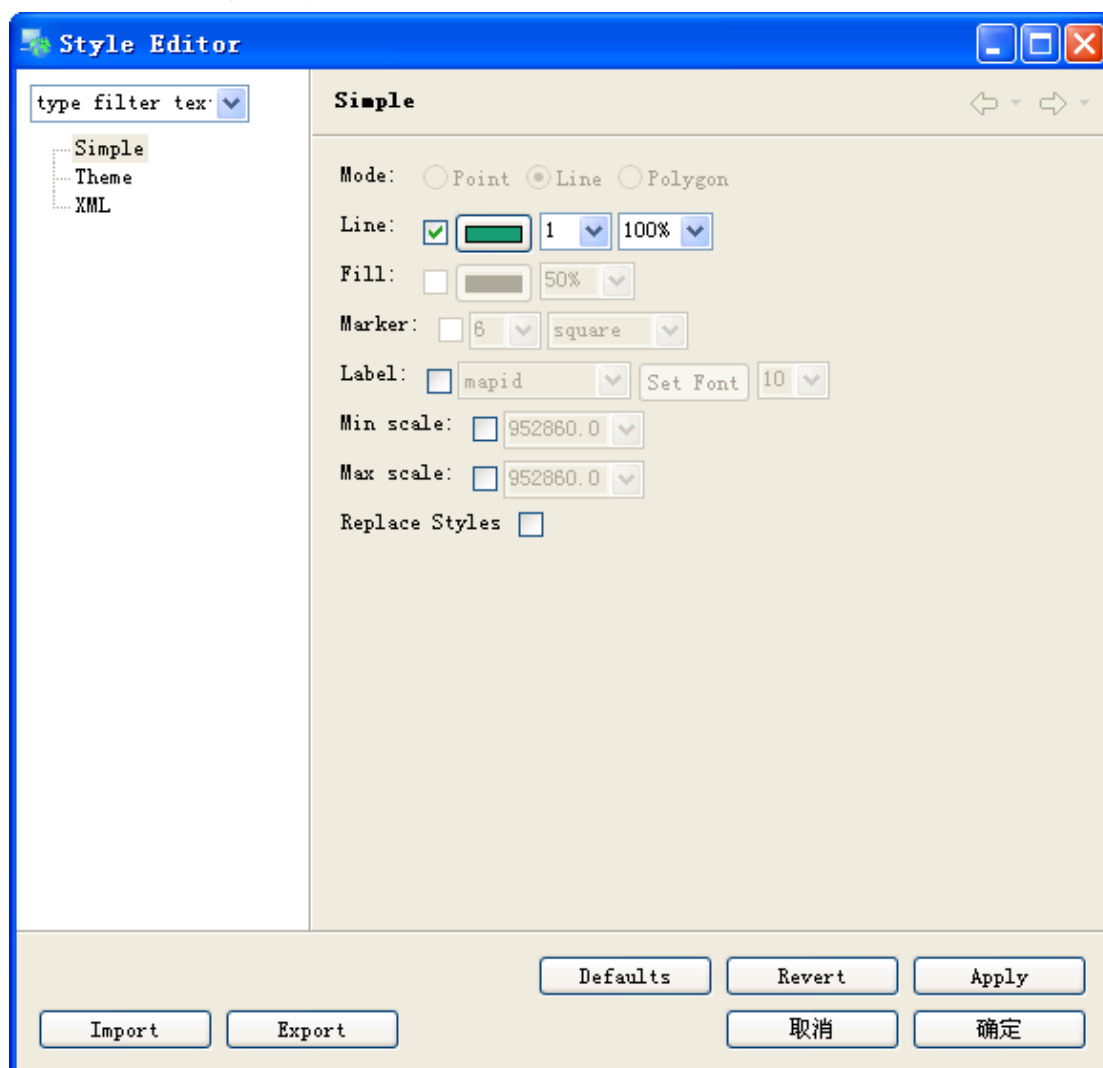
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

一般来说，我们不会直接去编辑上面那些难懂的文字。这里建议使用 uDig 来生成样式，样式制作好之后，再导入到 geoserver 中去应用，这样比较直观。下面简单介绍如何利用 uDig

来制作样式，并应用到 geoserver 中。

- 1) 运行 uDig，单击 Layer 菜单，选择 Add。
- 2) 出现 Add Data 对话框，选择 Web Feature Server，单击“下一步”。
- 3) 输入<http://localhost:8080/geoserver/wfs>，单击“下一步”。
- 4) 选择您要打开的层，单击“完成”。
- 5) uDig 窗口左侧用鼠标右键单击一个图层，选择 Change Style。
- 6) 配置完毕之后，在上一步打开的 Style Editor 对话框的左下角，单击“Export”按钮，将您建立好的样式导出为 sld 文件。
- 7) 用 IE 浏览器进入<http://localhost:8080/geoserver/>页面。
- 8) 点击“配置”，并输入之前您设置的用户名和密码登陆。
- 9) 再点击“数据”、“样式”、“新建”。
- 10) 输入样式 ID，即样式的名字。
- 11) 之后出现样式编辑器，单击窗口下方的“浏览”按钮，选择之前导出的 sld 文件。并单击“Upload”按钮，将文件提交。（如果内容中有中文，您或许需要将第一行改成“<?xml version='1.0' encoding='ISO-8859-1'?>”）
- 12) 单击窗口下面的“提交”按钮。单击窗口左侧的“应用”、“保存”按钮。

下面简单介绍 uDig 的 Style Editor 如何使用



上图所示样式编辑器具有三种方式定义样式，即 simple、Theme、XML。simple 方式提

供一种简单的途径实现样式，从上到下的功能项有如下作用：

- ① **Mode:** 这项是只读的，根据这个图层几何类型不同而不同，分为点、线、面三种。
- ② **Line:** 这项只有当 Mode 为 Line，即图层几何类型为线的时候有效，控制线的颜色、宽度以及透明度。
- ③ **Fill:** 这项只有当 Mode 为 Polygon，即图层几何类型为面的时候有效，控制多边形的颜色以及透明度。
- ④ **Marker:** 这项只有当 Mode 为 Point，即图层几何类型为点的时候有效，控制点的大小以及样式。
- ⑤ **Label:** 这项定义自动标注，控制自动标注所只用的字段、字体以及字体大小。
- ⑥ **Min Scale:** 这项定义该图层显示比例尺最小值，当地图当前显示比例尺小于这个值时，该图层不再绘制。
- ⑦ **Max Scale:** 这项定义该图层显示比例尺最大值，当地图当前显示比例尺大于这个值时，该图层不再绘制。
- ⑧ **Replace Styles:** 该项选中，则表示覆盖之前设置的其他样式，全部采用 simple 样式。

采用 Theme 方式设计样式，类似与 ArcMap 中惟一值方式渲染，这不提倡使用，如果想让一个图层上的地物按照种类绘图，即给出一个匹配条件来定义样式，而不是每个地物都指定一个样式，这需要直接修改 XML 代码，单击窗口左则的 XML 可以进入直接修改样式代码界面。

如果想用小图片来作为点符号绘制在地图上的话，您需要直接修改样式代码。找到 `<sld:Graphic></sld:Graphic>` 段落，替换为如下代码：

```
<sld:Graphic>
  <sld:ExternalGraphic>
    <sld:OnlineResource xlink:href="http://localhost:8080/geoserver/images/logo.png"/>
    <sld:Format>image/png</sld:Format>
  </sld:ExternalGraphic>
  <sld:Size>19</sld:Size>
</sld:Graphic>
```

之后还要找到 `<sld:StyledLayerDescriptor>`（第二行）后面添加一个属性 `xmllns:xlink="http://www.w3.org/1999/xlink"`

其中 `xlink:href` 表示图片的 url，`format` 表示图片格式（这里采用 png），`size` 表示符号尺寸。

2.5 如何发布 WMS 地图

地图由许多图层组成，WMS 地图也不例外，WMS 地图就是由多个 FeatureType 组成的一个 Layer Group。添加 WMS 地图的步骤如下：

- ① 使用 IE 浏览器进入 <http://localhost:8080/geoserver/> 页面。
- ② 点击“配置”，并输入之前您设置的用户名和密码登陆。
- ③ 再点击“WMS”、“内容”。
- ④ 出现“Web Map Server 内容”界面，滚动到页面最下方，单击“Add New Layer-Group”按钮。
- ⑤ 再次滚动到页面最下方，输入“图层组名称”，基础地图图层采用 AA:BB,AA:CC,.....

的形式输入，其中 AA 为名字空间，BB、CC、.....为 FeatureType 的 ID（名字），SRS（空间引用系统）输入采用的坐标系统，例如 EPSG:4326。单击“生成”按钮。

（此时，生成的坐标范围很可能是 -180,-90 到 180,90，这个需要您修改 geoserver\data\services.xml 文件中的值为正常的坐标范围，修改完毕之后重启 tomcat，这一步很重要，如果不修改的话，下面的瓦片生成会很多，因为瓦片从很遥远的地方开始生成）（如何知道正常的坐标范围？打开组成这个地图的每个 FeatureLayer 的坐标范围，取他们最大的一个）

- ⑥ 单击“提交”按钮，单击窗口左上方的“应用”，“保存”按钮。

2.6 如何提升访问速度

geoserver 中存储的矢量地图，绘制是比较慢的，现在流行采用瓦片缓冲的方式来提升访问速度。即事先将矢量地图绘制为光栅文件，这样客户访问地图是，服务器只需将这些已经绘制好的光栅文件发送到客户端。如果没有事先做好瓦片，UCMAP 访问地图之后，虽然 geowebcache 会自动生成之前访问的地图瓦片，但其他地方的瓦片还是没有，这不利于提升系统效率。对于较大的地图，我们建议要做下面的步骤来生成全部的瓦片文件。下面介绍 geowebcache 如何制作瓦片缓冲：

- ① 使用IE浏览器进入<http://localhost:8080/geoserver/>页面。
- ② 点击右上角的“登陆”输入用户名和密码登陆。
- ③ 使用IE浏览器进入<http://localhost:8080/geoserver/gwc/demo>页面。
- ④ 找到上面建立的 WMS 地图那一行，单击后面的“Seed this layer”
- ⑤ Zoom start 表示显示比例开始等级，0 就可以。Zoom stop 表示显示比例结束等级，推荐 16，超过这个数基本没有意义。按下“Submit”按钮。（注意：如果您上面没有修改 geoserver\data\services.xml 文件中对应的 WMS 地图的坐标范围，那么这里需要您输入正常的坐标范围）

三、客户端部分

3.1 配置 UCMAP 开发环境

首先安装 UCMAP 开发包，以下假定您安装的位置为默认，即 C:\Program Files\UCMAP SDK。本开发包支持 vs2005 与 vs2008 智能设备开发，步骤略有不同，如下所述：

VS2005 环境配置：

- ① 请确保 VS2005 安装了 Visual C++下的“智能设备可编程技术”。
- ② 单击菜单【工具】—》【选项】。选择“项目和解决方案”中的“VC++目录”，选择“Pocket PC 2003(ARMV4)”平台，将您安装的 SDK 的 include 文件夹加入包含文件列表，即 C:\Program Files\UCMAP SDK\include 文件夹。
- ③ 单击菜单【工具】—》【选项】。选择“项目和解决方案”中的“VC++目录”，选择“Pocket PC 2003(ARMV4)”平台，将您安装的 SDK 的 lib\VS2005 文件夹加入库文件列表，即 C:\Program Files\UCMAP SDK\lib\VS2005 文件夹。

VS2008 环境配置：

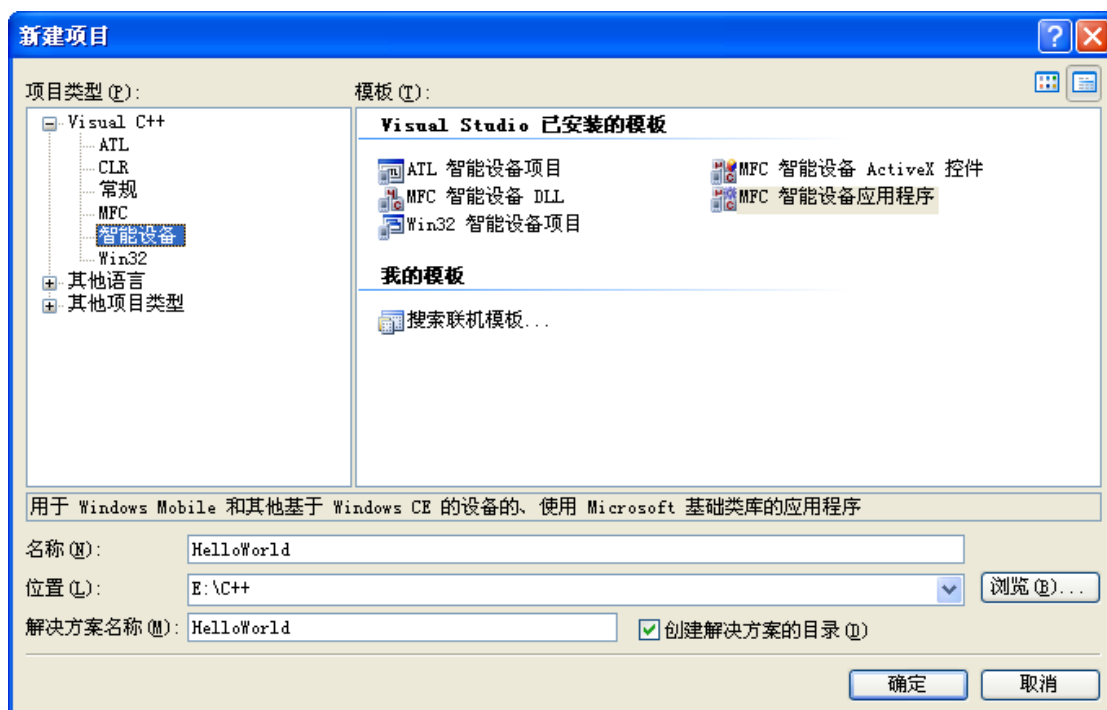
- ① 请确保 VS2008 安装了 Visual C++下的“智能设备可编程技术”。
- ② 单击菜单【工具】—》【选项】。选择“项目和解决方案”中的“VC++目录”，选择“Pocket PC 2003(ARMV4)”平台，将您安装的 SDK 的 include 文件夹加入包含文件列表，即 C:\Program Files\UCMAP SDK\include 文件夹。
- ③ 单击菜单【工具】—》【选项】。选择“项目和解决方案”中的“VC++目录”，选择“Pocket PC 2003(ARMV4)”平台，将您安装的 SDK 的 lib\VS2008 文件夹加入库文件列表，即 C:\Program Files\UCMAP SDK\lib\VS2008 文件夹。

由于 UCMAP 采用数据和程序分离的方式，您要打开地图，首先模拟器要能连上网络才行，下列步骤教您如何让 Pocket PC 2003 模拟器连上网络：

- ① 使用 regedit 打开注册表，找到 “[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows CE Services]” 添加如下字段和值“AllowLSP”=dword:00000000
- ② 打开 ActiveSync，点击“文件”——>“连接设置”，在“允许连接到以下其中一个端口”下选择“DMA”。
- ③ 打开 VS2005，点击菜单“工具”——>“设备仿真管理器”，选择一个 PocketPC SE 2003 模拟器。在右键菜单中点击“连接”。等模拟器启动后，再点击“插入底座”，此时，通过 ActiveSync 来连接到模拟器，并进行数据同步。电脑上弹出“新建合作关系”对话框，您可以直接点击“取消”，**一定要点，否则模拟器无法上网。**

3.2 利用 UCMAP 地图组件建立客户端应用程序

- ① 运行 Visual Studio 2005 或者 Visual Studio 2008。单击菜单【文件】—》【新建】—》【项目】，如下图所示，选择 Visual C++、智能设备、MFC 智能设备应用程序，并输入名称。单击“确定”。



② 选择下图的“下一步”



③ 添加您想加入的平台



④ 选择“在共享 DLL 中使用 MFC”，这一点很重要。下面的设置无关紧要，直接下一步即可。



- ⑤ 打开您视图类的头文件，这里是 HelloWorldView.h 文件。在 #pragma once 下面添加三行代码。

```
#include "MapControl.h"
```

```
using namespace ogc;
```

```
using namespace controls;
```

并给视图类添加一个成员 `MapControl* m_pMapControl;` 在类视图中添加 WM_ERASEBKGD、WM_KEYUP、WM_LBUTTONDOWN、WM_LBUTTONUP、WM_MOUSEMOVE 这五个消息的处理函数，并且在相应的消息处理函数中调用 MapControl 对象的同名函数。

- ⑥ 打开您视图类的源文件，这里是 HelloWorldView.cpp 文件。在视图类构造函数中键入 `m_pMapControl=NULL;`。析构函数中键入 `delete m_pMapControl;`。在 OnDraw 函数中键入 `m_pMapControl->OnDraw(pDC);`，并将 OnDraw 函数参数的注释去掉。在上述五个消息处理函数中调用 `m_pMapControl` 相应的函数。

- ⑦ 重载 OnInitialUpdate 函数。在 OnInitialUpdate 函数中键入如下三行代码。创建实际的 MapControl 对象。

```
CRect rect;
```

```
this->GetWindowRect(rect);
```

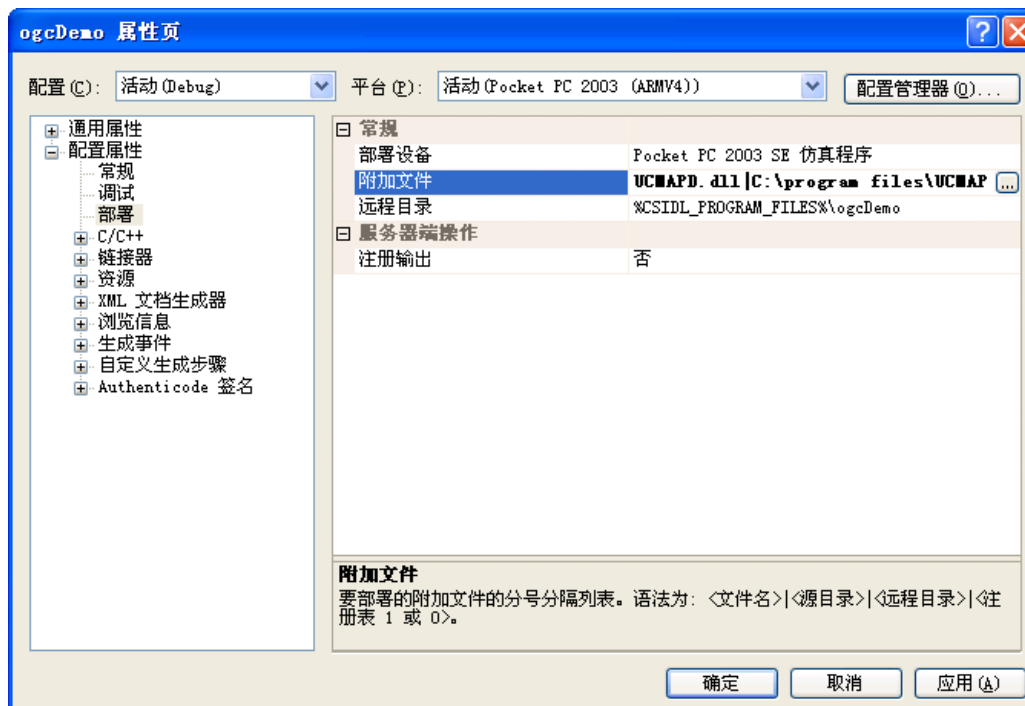
```
m_pMapControl=new MapControl(this,(short)rect.Width(),(short)rect.Height());
```

- ⑧ 这时候您可以使用 MapLoader 类的静态函数 loadMapXML 来加载地图。例如：`MapLoader::loadMapXML(m_pMapControl,L"c:\\test.xml");` 该代码读取 c 盘下的 test.xml 中的配置，加载并显示采用 wms 标准发布的地图，具体如何操作，请参考“2.4 客户端如何浏览服务器发布的地图”一章节。

3.3 如何部署应用程序

按照上面的步骤您建立起了属于自己的 UCMAP 程序，下面介绍如何部署您的应用程序。

- ① 运行 Visual Studio 2005 或者 Visual Studio 2008，打开 demo 这个解决方案。单击菜单【项目】—【属性】，选中“配置属性”的“部署”。左上角的配置选择“Debug”。



- ② 点击左侧“配置属性”列表中的“部署”，点击右侧“附加文件”后面的“...”按钮，添加 UCMAPD.dll|C:\program files\UCMAP SDK\lib\VS2005\|C:\CSIDL_PROGRAM_FILES%\\$(ProjectName)|0。这句话的意思是部署到模拟器或者真机的时候，自动将 C:\program files\UCMAP SDK\lib\VS2005\下的 UCMAPD.dll 文件拷贝到应用程序目录。您的程序需要通过加载 xml 文件来加载地图，那么也需要使用类似的方法添加上述一条语句到“附加文件”列表中。同理将 product.sn 文件也添加进入。（这个文件是许可文件，没有它程序无法运行）
- ③ 左上角的配置选择“Release”时，上述语句应该写成 UCMAP.dll.....，即 Release 版本应该加载 UCMAP.dll 动态链接库。VS2008 环境下只需注意将上述语句中的“VS2005”替换为“VS2008”。这样即可调试、运行您的应用程序了。

Pocket PC 2003 模拟器上的显示效果很差，文字有些糊，但是真机上或者以上版本的模拟器（例如：WM5 模拟器）显示的效果却很好。如果您想输出到真机上运行，您需要将上述对话框中所有的“附加文件”与 exe 文件一同放置在目标真机上，例如：UCMAP.dll、msvcr80.dll、atl80.dll、MFC80U.dll、product.sn 等等。对于 vs2008 环境，就是 UCMAP.dll、msvcr90.dll、atl90.dll、MFC90U.dll、product.sn 等文件。

3.4 客户端如何浏览服务器发布的地图

2.2 章节最后中提到您可以使用 MapLoader 类的静态函数 loadMapXML 来加载地图。这个函数接收两个参数，一个是 MapControl 对象的指针，一个是地图配置文件的文件标识符。现在就来说明这个地图配置文件（XML 格式）如何编写。下面是一个例子：

```
<?xml version="1.0" encoding="UTF-8"?>
<map maxExtent="-180,180,-90,90" numberOfLayers="1">
  <layer>
    <type>WMSWithCache</type>
    <name>nanjing</name>
    <url>http://192.168.1.10:8080/geoserver/gwc/service/wms</url>
    <WFSServerUrl>http://192.168.1.10:8080/geoserver/wfs</WFSServerUrl>
    <namespace>creable</namespace>
    <namespaceURL>http://www.creable.cn</namespaceURL>
    <maxExtent>-180,180,-90,90</maxExtent>

    <resolutions>1.71661376953125E-4,8.58306884765625E-5,4.291534423828125E-5,2.14576721
    19140625E-5,1.0728836059570312E-5</resolutions>
    <projection>EPSG:4326</projection>
    <format>image/png</format>
    <pixelWidth>256</pixelWidth>
    <pixelHeight>256</pixelHeight>
  </layer>
</map>
```

maxExtent: 表示 geowebcache 生成瓦片时采用的地图最大范围，坐标系统为 EPSG:4326 时，该值为 -180,180,-90,90

numberOfLayers: 表示图层个数（一个 <layer>.....</layer> 为一个图层）

type: 图层类型，WMSWithCache 表示带有瓦片缓冲机制的 WMS 标准图层。

name: 为图层名称，与 geoserver 中 WMS 中的 Layer Group 名字一致。

url: 为服务器 WMS 的 url，应该使用真实 ip 地址替换上面的例子。

WFSServerUrl: 为服务器 WFS 的 url，应该使用真机 ip 地址替换上面的例子。

namespace: 为该地图所属的名字空间，与 geoserver 中设置一致。

namespaceURL: 为名字空间对应的 url，与 geoserver 中设置一致。

maxExtent: 为地图最大范围，与上面那个 maxExtent 相同即可。

resolutions: 为显示比例尺列表，用逗号分割，（显示比例尺 = 瓦片实际坐标宽度 ÷ 瓦片像素宽度 = 当前窗口实际坐标宽度 ÷ 窗口像素宽度。）坐标系统为 EPSG:4326 时，geowebcache 裁切瓦片时将 -180,180,-90,90 范围的地图裁切为 2 块，即每块实际坐标尺寸为 180×180，其像素尺寸为 256×256， $180 \div 256 = 0.703125$ ，故第一个显示比例尺为 0.703125，之后的显示比例尺为前一个比例尺的一半，即 $0.703125 \div 2 = 0.3515625$ ，所以显示比例尺列表为 0.703125,0.3515625,0.17578125,0.087890625,0.0439453125,0.02197265625,0.010986328125,0.0054931640625,0.00274658203125,0.001373291015625,6.866455078125E-4,3.4332275390625E-4,1.71661376953125E-4,8.58306884765625E-5,4.291534423828125E-5,2.1457672119140625E-5,1.0728836059570312E-5,.....。作为一个城市的地图来说，前面几个比例尺显然太大，

没有意义，所以上面的例子中比例尺直接从 1.71661376953125E-4 开始，省去了前面的比例尺，即实现了打开地图时显示合适的比例尺，而不允许用户缩小地图到不合适的显示比例尺。

projection: 为坐标系统。EPSG:4326 即 WGS84

format: 图像格式

pixelWidth: 瓦片像素宽度，一般为 256

pixelHeight: 瓦片像素高度，一般为 256

编写的这些 xml 地图配置文件也要根据需要，按照 2.3 章节介绍的方法，一起部署到模拟器或真机上供使用。

3.5 如何进行空间查询以及属性查询

UCMAP 使用采用 WFS 标准，支持客户程序的空间查询操作。首先介绍空间关系谓词，包括：Disjoint, Touches, Crosses, Within, Overlaps, Contains, Intersects, Equals, DWithin, Beyond, BBOX。前面八个是 OGC 定义的空间关系谓词，其解释如下：

(1) Disjoint

任意给两个空间实体 a 和 b(拓扑闭合的): $a.disjoint(b) \iff I(a) \cap I(b) = \emptyset$

(2) Touches

touch 关系适合于面/面、线/线、线/面、点/面、点/线间的相互关系,但点/点间没有定义这种关系。对于满足要求的空间实体 a 和 b:

$a.touch(b) \iff (I(a) \cap I(b) = \emptyset) \wedge (a \cap b \neq \emptyset)$

touch 关系的几个例子如图 2 所示:

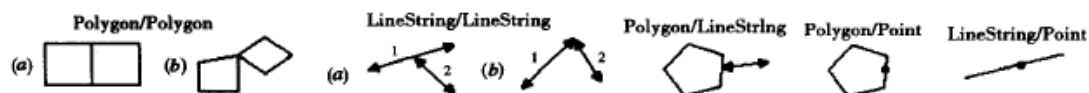


图 2 Touch 关系的例子

(3) Crosses

crosses 关系适用于点/线、点/面、线/线及线/面间关系。令 $\dim(a)$ 表示空间对象 a 的维度,则 crosses 关系定义为:

$a.crosses(b) \iff (\dim(I(a) \cap I(b)) < \max(\dim(I(a)), \dim(I(b)))) \wedge (a \cap b \neq a \wedge (a \cap b) \neq b)$

crosses 关系的几个例子如图 3 所示:

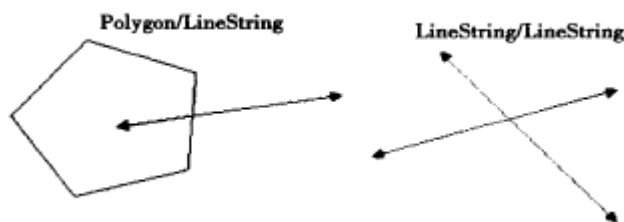


图 3 Cross 关系的例子

(4) Within

within 关系定义为:

$a.within(b) \iff (a \cap b) = a \wedge (I(a) \cap I(b)) \neq \emptyset$

overlaps 的几个例子如图 4 所示:



图 4 Within 关系的例子

(5) Overlaps

overlaps 关系适用于面/面、线/线和点/点关系,定义如下:

$a.overlaps(b) \iff \dim(I(a)) = \dim(I(b)) = \dim(I(a) \cap I(b)) \wedge (a \cap b \neq a) \wedge (a \cap b \neq b)$

overlaps 的几个例子如图 5 所示:

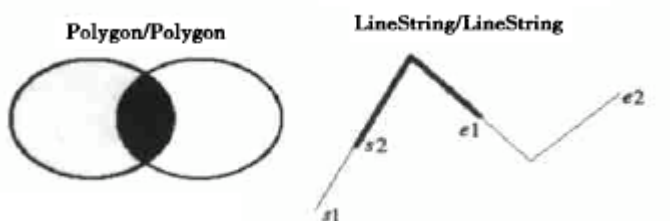


图 5 Overlaps 关系的例子

除了上述 5 个关系,为了使用的方便,还定义了以下几个关系:

(6) Contains:

$a.contains(b) \iff b.within(a)$

(7) Intersects:

$a.intersects(b) \iff \neg a.disjoints(b)$

(8) Equals:

$a.equals(b) \iff a.within(b) \wedge b.within(a)$

后面三个是 WFS 定义的空间关系谓词。DWithin 可以查询一个圆形范围内的要素,需要给一个距离参数。BBOX 查询矩形范围内的要素。

属性查询类似于数据库 SQL 语句,提供如下属性比较谓词: Equal, NotEqual, Less, Greater, LessOrEqual, GreaterOrEqual, Like, IsNull, Between。

- ① Equal: 等于。
- ② NotEqual: 不等于。
- ③ Less: 小于。
- ④ Greater: 大于。
- ⑤ LessOrEqual: 小于等于。
- ⑥ GreaterOrEqual: 大于等于。
- ⑦ Like: 相似。使用通配符 “*” 表示任意数量的字符,“?” 表示一个字符。
- ⑧ IsNull: 为空, 字段值为空。
- ⑨ Between: 介于两者之间。

要执行上述查询,有三种方法:

- ① 调用一个函数帮您生成常用的 WFS 查询条件,并执行查询。

WMSLayer 类中有一组函数能够提供一些缺省的查询。(WMSLayer 对象的指针如

何获得？您的程序调用 `MapLoader::loadMapXML` 函数之后，`map` 对象中的第一个图层便是 `WMSLayer`，所以可以通过这样的语句获取该指针 `dynamic_cast<WMSLayer*>(m_pMapControl->getMap()->getLayer(0))`

`WMSLayer:: getFeatureByCircle`: 在指定的圆范围内查询某(几)个图层的空间地物信息。

`WMSLayer:: getFeatureByRect`: 在指定的矩形范围内查询某(几)个图层的空间地物信息。

`WMSLayer:: getFeatureByPoint`: 点选查询。指定地图的某一点的位置，查询该处某(几)个图层的空间地物信息。

`WMSLayer:: getFeatureByPolygon`: 在指定的多边形范围内查询某(几)个图层的空间地物信息。

`WMSLayer:: getFeatureByLine`: 查询与指定的线相交的某(几)个图层的空间地物信息。

以 `getFeatureByCircle` 为例，说明如何使用。

```
QueryParam params;
```

```
QueryLayerParam* param=params.createAndAddQueryLayerParam(L"layerName");
```

```
Point pt(118.780716f,32.047253f);
```

```
GetFeatureResult *result=layer->getFeatureByCircle(&pt,100, &params);
```

首先建立一个 `QueryParam` 对象（该对象有个十分重要的参数 `maxFeatures`，即此次查询最大要素个数，其默认值为 100，调用 `setMaxFeatures` 函数可以修改这个值），之后调用 `QueryParam` 类的 `createAndAddQueryLayerParam` 函数创建一个 `QueryLayerParam`，这个对象对应着服务器上的一个 `FeatureType`，即 WMS 地图中的一个图层，名字为 “`layerName`”，需要您自己修改。然后调用 `WMSLayer` 的 `getFeatureByCircle` 函数，以一个 `point` 对象、一个距离和之前建立的 `QueryParam` 对象作为参数。函数返回一个 `GetFeatureResult` 对象指针，这个对象包含您需要的查询结果，但这个指针不能删除。

② 调用一系列函数帮您生成您自定义的 WFS 查询条件后查询。

`WMSLayer` 类有个 `getFeature` 函数，这个函数是执行查询用的，在执行这个函数之前，您要建立一个 `QueryParam` 对象，并填充里面的查询条件。`QueryParam` 类具有如下的一组函数来帮助您编写查询条件，`spatialFilterBeginAnd`，`spatialFilterEndAnd`，`spatialFilterBeginOr`，`spatialFilterEndOr`，`spatialFilterBeginNot`，`spatialFilterEndNot()`，`spatialFilterInsert` 这些函数用于编写空间查询条件。`AttFilterBeginAnd`，`AttFilterEndAnd`，`AttFilterBeginOr`，`AttFilterEndOr`，`AttFilterBeginNot`，`AttFilterEndNot`，`AttFilterInsert` 这些函数用于编写属性查询条件。空间查询条件和属性查询条件会被系统自动合并。下面举个例子说明如何使用这些函数：

假定我们要查询 `pnanjing` 这个 `FeatureType` 上 `name` 字段值类似于“新百商场*”（即字段值由“新百商场”开头，并且后面字符数量随意）、并且 `kind` 字段等于“2200”、并且位于指定的矩形范围内。这个查询中，我们将属性查询和空间查询区分开来处理。具体代码如下所示（`layer` 为 `WMSLayer` 对象指针）：

```
QueryParam params;
```

```
QueryLayerParam* param=params.createAndAddQueryLayerParam(L"pnanjing");
```

```
param->AttFilterBeginAnd();
```

```
param->AttFilterInsert(L"name",Like,L"新百商场*",NULL);
```

```
param->AttFilterInsert(L"kind",Equal,L"2200",NULL);
```

```
param->AttFilterEndAnd();  
Envelope env(0,120,0,120);  
param->SpatialFilterInsert(BBOX,NULL,&env,0,layer->getShapeFieldName(),layer->getProjection(),layer->getGeometryType());  
GetFeatureResult *result =layer->getFeature(&params);
```

首先建立 QueryParam 对象，在建立名字为“pnanjing”的 QueryLayerParam 参数，之后调用 AttFilterBeginAnd 添加<And>标记，在调用两次 AttFilterInsert 添加两个不同的条件，调用 AttFilterEndAnd 添加</And>标记，到此属性查询条件建立完毕。后面调用 SpatialFilterInsert 直接写了一个空间查询条件。最后调用 WMSLayer 的 getFeature 函数得到查询结果 result，但这个指针不能删除。

您如果想在一次查询中同时查询多个图层，那么需要多次调用 QueryParam 类的 createAndAddQueryLayerParam 方法创建 QueryLayerParam 对象，并使用上述方法编写查询条件。

使用上述函数建立的查询条件可以通过调用 QueryLayerParam 类的 getAttFilter 函数得到生成的属性查询条件文本，调用 getSpatialFilter 得到生成的空间查询条件文本。

③ 您直接写出查询语句后查询。

直接写查询语句，您需要知道 WFS 查询语句如何编写。

首先介绍属性查询语句如何编写。这里可以解释上面的例子中两个属性查询条件为什么不是 条件1 And 条件2，而是 <And> 条件1 条件2 </And>。

WFS 查询条件中 And、Or、Not 这三个逻辑判断谓词需要使用<××></××>的形式将条件包含其中，这一点和 SQL 差异比较大。

以下说明每种属性比较谓词如何写成 WFS 查询条件：

- 1) Equal: 第一个%s 填写字段名称，第二个%s 填写字段值
<PropertyIsEqualTo><PropertyName>%s</PropertyName><Literal>%s</Literal></PropertyIsEqualTo>
- 2) NotEqual: 第一个%s 填写字段名称，第二个%s 填写字段值
<PropertyIsNotEqualTo><PropertyName>%s</PropertyName><Literal>%s</Literal></PropertyIsNotEqualTo>
- 3) Less: 第一个%s 填写字段名称，第二个%s 填写字段值
<PropertyIsLessThan><PropertyName>%s</PropertyName><Literal>%s</Literal></PropertyIsLessThan>
- 4) Greater: 第一个%s 填写字段名称，第二个%s 填写字段值
<PropertyIsGreaterThan><PropertyName>%s</PropertyName><Literal>%s</Literal></PropertyIsGreaterThan>
- 5) LessOrEqual: 第一个%s 填写字段名称，第二个%s 填写字段值
<PropertyIsLessThanOrEqualTo><PropertyName>%s</PropertyName><Literal>%s</Literal></PropertyIsLessThanOrEqualTo>
- 6) GreaterOrEqual: 第一个%s 填写字段名称，第二个%s 填写字段值
<PropertyIsGreaterThanOrEqualTo><PropertyName>%s</PropertyName><Literal>%s</Literal></PropertyIsGreaterThanOrEqualTo>
- 7) Like: 第一个%s 填写字段名称，第二个%s 填写字段值
<PropertyIsLike
wildCard="*" singleChar="?"
escapeChar="\"><PropertyName>%s</PropertyName><Literal>%s</Literal></PropertyIsLike>

- 8) IsNull: 第一个%s 填写字段名称
<PropertyIsNull><PropertyName>%s</PropertyName></PropertyIsNull>
- 9) Between: 第一个%s 填写字段名称, 第二个%s 填写字段值下限, 第三个%s 填写字段值上限
<PropertyIsBetween><PropertyName>%s</PropertyName><LowerBoundary>%s</LowerBoundary><UpperBoundary>%s</UpperBoundary></PropertyIsBetween>

综上所述, ②的例子中属性查询语句部分可以直接写成:

```
<And><PropertyIsLike                wildCard="*"                singleChar="?"  
escapeChar="\"><PropertyName>name</PropertyName><Literal> 新 百 商 场  
*</Literal></PropertyIsLike><PropertyIsEqualTo><PropertyName>kind</PropertyName><  
Literal>2200</Literal></PropertyIsEqualTo></And>
```

调用 QueryLayerParam 的 setAttFilter 函数, 将上面的字符串传递进去即可 (②的例子中使用的 AttFilterBeginAnd、AttFilterInsert 等函数就是完成了上面那串字符而已)。

下面介绍空间查询语句如何编写。以下说明每种空间比较谓词如何写成 WFS 查询条件:

- 1) Equals: 第一个%s 填写图形字段名称 (WMSLayer 类的 getShapeFieldName 函数可以得到), 第二个%s 填写 GML3 格式的图形
<Equals><PropertyName>%s</PropertyName>%s</Equals>
- 2) Disjoint: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形
<Disjoint><PropertyName>%s</PropertyName>%s</Disjoint>
- 3) Touches: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形
<Touches><PropertyName>%s</PropertyName>%s</Touches>
- 4) Within: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形
<Within><PropertyName>%s</PropertyName>%s</Within>
- 5) Overlaps: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形
<Overlaps><PropertyName>%s</PropertyName>%s</Overlaps>
- 6) Crosses: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形
<Crosses><PropertyName>%s</PropertyName>%s</Crosses>
- 7) Intersects: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形
<Intersects><PropertyName>%s</PropertyName>%s</Intersects>
- 8) Contains: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形
- 9) DWithin: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形, 第三个%f 填写一个距离 (实际坐标距离)
<DWithin><PropertyName>%s</PropertyName>%s<Distance
units="m">%f</Distance></DWithin>
- 10) Beyond: 第一个%s 填写图形字段名称, 第二个%s 填写 GML3 格式的图形, 第三个%f 填写一个距离 (实际坐标距离)
<Beyond><PropertyName>%s</PropertyName>%s<Distance
units="m">%f</Distance></Beyond>
- 11) BBOX: 第一个%s 填写图形字段名称, 第二个%s 填写坐标系统, 后面四个%f 填写一个矩形范围 (顺序为 xmin, ymin, xmax, ymax)
<BBOX><PropertyName>%s</PropertyName><Box
srsName="%s"><coordinates>%f,%f %f,%f</coordinates></Box></BBOX>

多个空间查询条件也需要逻辑比较谓词来包含，一个空间查询条件可以看作是一个属性查询条件，这方面操作是一致的。

注意：如果服务器设置对于 `wfs.GetFeature` 的访问权限，那么客户端就必须调用 `WMSLayer` 对象的 `setUserName` 和 `setPassword` 函数来指定用户名和密码才能成功查询。

上面提到的 GML3 格式的图形，涉及到如何将图形对象转换为 GML3 格式的问题，详细的您可以参考有关书籍，这里介绍如何调用 `UCMAP` 的函数来达到转换的目的，`GMLWriter` 类有个静态函数 `toGML3`，调用方法如下：

```
CString geo=GMLWriter::toGML3(geometry,projection,geometryType);
```

其中 `geometry` 为图形对象指针，`projection` 由 `WMSLayer` 的 `getProjection` 函数得到，`geometryType` 由 `WMSLayer` 的 `getGeometryType` 函数得到。`geo` 就是 GML3 格式的图形描述字符串。

3.6 编辑工具如何使用

注意：如果服务器设置对于 `wfs.GetFeature` 的访问权限，那么客户端就必须调用 `WMSLayer` 对象的 `setUserName` 和 `setPassword` 函数来指定用户名和密码才能成功查询。

`ogc` 命名空间包含四个工具：`AddFeatureTool` 用于添加要素（点、线、面都可以）、`DeleteFeatureTool` 用于删除要素、`EditFeatureTool` 用于修改要素图形的节点、`EditFeatureAttTool` 用于修改要素的属性。这些工具的使用方法等同于 `GridGIS` 中的其他 `MapTool`，即 `new` 一个工具对象，将指针传递给 `MapControl` 的 `setCurrentTool` 函数。在使用上述编辑工具之前，需要先指定当前编辑层（`FeatureType`）。可以调用 `WMSLayer` 类的 `setEditLayer` 函数指定当前层。

使用 `AddFeatureTool` 添加点要素操作步骤（包含代码编写和用户操作）：

- ① 调用 `WMSLayer` 类的 `setEditLayer` 函数指定一个容纳点要素的 `FeatureType`
- ② `new` 一个 `AddFeatureTool` 工具对象，将指针传递给 `MapControl` 的 `setCurrentTool` 函数。
- ③ 自 `IEditFeatureFilter` 接口派生一个类，并将指针传递给 `AddFeatureTool` 类的 `setFilter` 函数。（来过滤用户添加数据事件，用户添加数据之前，`UCMAP` 会调用这个接口的 `doFilter` 函数，您的程序可以在这个函数里显示输入属性对话框来供用户输入属性，`doFilter` 返回 `true` 则继续添加数据，否则不执行添加）
- ④ 用户在屏幕上一个位置单击，屏幕出现一个红十字，这时候您可以使用手机键盘上的“上下左右”键精确移动这个红十字，或者再次点击屏幕快速移动到一个新的位置。
- ⑤ 用户按下“确定”键，`UCMAP` 调用 `IEditFeatureFilter` 的派生类中的 `doFilter` 函数。
- ⑥ 您的程序在 `doFilter` 函数中显示输入属性对话框，让用户输入属性。
- ⑦ 用户点击对话框“确定”按钮，您的 `doFilter` 函数即返回 `true`，于是 `UCMAP` 完成后面的添加数据工作。

使用 `AddFeatureTool` 添加线的操作步骤与添加点类似，但需要选择一个容纳线要素的 `FeatureType`，并确保用户单击“确定”按钮，不是执行添加要素，而是添加了线的一个节点，

重复输入多个节点之后，用户长时间使用触摸笔按住屏幕 3 秒以上时间再松开，才会调用 doFilter 函数，并且添加数据。

使用 AddFeatureTool 添加面的操作步骤与添加线类似，不同的是添加面最后会自动封闭。

使用 DeleteFeatureTool 删除要素操作步骤（包含代码编写和用户操作）：

- ① 调用 WMSLayer 类的 setEditLayer 函数指定一个 FeatureType
- ② new 一个 DeleteFeatureTool 工具对象，将指针传递给 MapControl 的 setCurrentTool 函数。
- ③ 用户在屏幕上点选一个对象，该对象会闪烁表示被您选中，然后单击键盘上的“确定”键，即可删除该对象。

使用 EditFeatureTool 移动点或者线面上的节点的操作步骤（包含代码编写和用户操作）：

- ① 调用 WMSLayer 类的 setEditLayer 函数指定一个 FeatureType
- ② new 一个 EditFeatureTool 工具对象，将指针传递给 MapControl 的 setCurrentTool 函数。
- ③ 用户在一个要素上点击，被选中的要素使用绿色十字显示出所有的节点（包括点、线节点、面节点），您使用触摸笔选中一个节点，屏幕会将该节点显示为红色，使用红色虚线表示您更改之后的效果，这之后，您可以使用触摸笔单击一个位置，指定该节点移动到那里，也可以使用键盘的“上下左右”键精确移动。然后按下键盘的“确定”键，结束修改。

使用 EditFeatureAttTool 修改要素属性的操作步骤：

- ① 调用 WMSLayer 类的 setEditLayer 函数指定一个容纳点要素的 FeatureType
- ② new 一个 EditFeatureAttTool 工具对象，将指针传递给 MapControl 的 setCurrentTool 函数。
- ③ 自 IEditFeatureFilter 接口派生一个类，并将指针传递给 EditFeatureAttTool 类的 setFilter 函数。
- ④ 用户在一个要素上点击，UCMAP 会自动调用您的 doFilter 函数，您可以在这个函数中通过传递给您的 WMSLayer 对象指针来读取用户点击对象的属性，并显示出修改属性对话框供用户修改，用户修改并按下“确定”键之后，doFilter 返回 true，UCMAP 继续完成下面的修改要素属性工作。

四、GPS 读取、解析、显示

4.1 GPS 读取

UCMAP 的 lbs 命令空间含有一个名为 GPS 的类，首先调用这个类的 open 方法，将 com 号和比特率传递进去即可打开 gps 模块，您需要自行实现 IGPSUpdate 接口，也将实现该接口的类的指针传递给 open 方法，这样 GPS 类在收到新的 gps 信息是，会自动调用 IGPSUpdate 接口中的 GPSUpdate 函数，您可以在这个函数中处理收到的数据。

4.2 GPS 协议解析

GPSUpdate 函数有个 GPS 对象指针作为参数，通过这个参数，你可以获取到 gps 信息中的所有信息，包含时间日期、经纬度、方向、速度等等信息。GPS 类中的 reader 成员包含了这些所有信息，例如：假设 gps 为这个 GPS 对象指针，那么 `gps->reader->valid` 就是当前 gps 信息是否有效，`gps->reader->m_iLonDegree` 就是经度中的度部分，以此类推。

4.3 GPS 显示

GPS 点的显示和其他没有什么区别。需要您实现 ICustomDraw 接口，并将实现这个接口的对象指针传递给 MapControl 的 setCustomDraw 函数，并且在这个接口的 draw 函数中将 gps 点绘制出来，UCMAP 便会自动调用您写的这段代码。代码详见案例代码中的 PositionCustomDraw 类。

五、license 授权

5.1 免费授权

您在我公司网站下载的试用版中含有一个免费授权，包含一个期限为 1 个月的试用时间，如果时间超过授权的时间，那么您在使用 UCMAP 打开地图时，会提示一个信息。显示的信息如下：

Time expired, the following time frame to be able to try

××××-××-×× ××××-××-××。

其中×××为年月日。您可以将电脑或手机的日期修改在这个以内，便可以再次试用。也可以向我公司申请新的免费授权。

5.2 部署授权

您也可以购买我们的部署授权，需要您提供对方服务器名、服务器 ip 地址以及项目名称信息，之后接收到我公司提供的部署授权之后，复制到您客户的每个设备的程序文件夹里，（与 UCMAP.dll 在同一文件夹），该授权没有时间限制，没有对设备的数量限制。

以上具体如何使用，可以参考我公司提供的案例代码。