

引路蜂 电子地图 API 开发人员手册



James Shen

www.guidebee.biz

Guidebee Pty Ltd.



文档版本

版本	日期	描述	作者
1.0	2011 年 1 月 8 日	中文版初稿	James Shen jing.shen@guidebee.biz



目录

文档版本	2
1.0 引路蜂地图 API 概述.....	5
2.0 地图开发包类定义	7
2.1 开发包中核心类定义	7
2.2 GIS 包定义	10
2.3 GIS.GEOMETRY 包定义	11
2.4 GIS.DRAWING 包定义	11
2.5 GIS.RASTER 包定义	12
2.6 GIS.SERVICE 包定义	14
2.7 GIS.VECTOR 包定义	15
2.8 GIS.LOCATION 包定义	16
2.9 GIS.NAVIGATION 包定义	16
2.10 DRAWING 包定义	18
2.11 DRAWING.GEOMETRY 包定义	19
2.12 典型软件结构图	20
3.0 一些地图开发包基本知识	21
3.1 地图对象类层次关系	21
3.2 地图图片读取类层次关系	21
3.3 地图服务提供商	22
3.4 图形子系统	22
4.0 准备工作	27
4.1 引路蜂地图开发包使用许可	27
4.2 GOOGLE, CLOUDMADE, MAPABC 地图 MAP KEY	27
4.3 地图配置	27
4.3.1 工作线程数	27
4.3.2 使用缓存	28
4.3.3 查询配置	28
4.4 第一个地图应用	28
5.0 地图基本操作	34
5.1 设置地图类型	34
5.2 放大、缩小	35
5.3 地图平移	35
5.4 地图缓存	36
6.0 使用地图服务	37
6.1 地址查询	37
6.2 路径查询	38
6.3 本地查询	39
6.4 IP 地址查询	40
6.5 地址反编码	41
6.6 选择地图服务	42
6.7 叠加自定义图层	44

7.0	离线地图	47
7.1	离线地图生成工具结构图	47
7.2	离线地图生成工具使用方法	48
7.3	离线地图示例	50
8.0	GPS 定位.....	52
9.0	矢量地图和导航 API	53

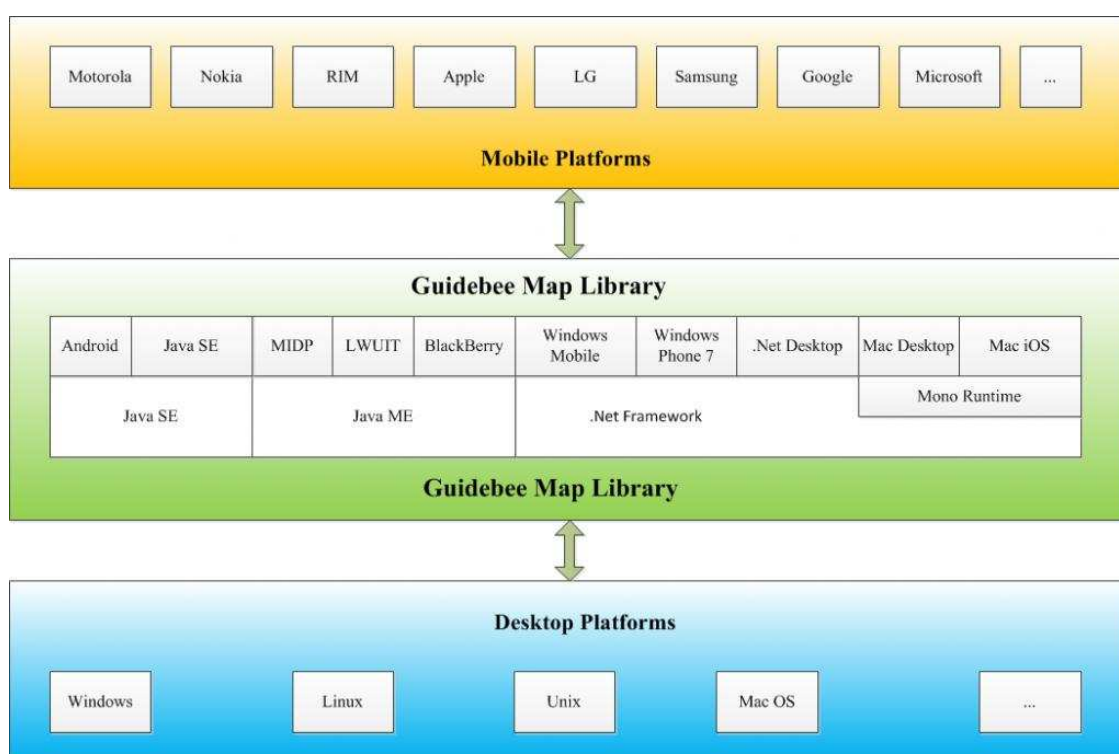
1.0 引路蜂地图 API 概述

提起地图开发包，Google 地图和 Bing 地图主要应用于 Web 应用，在离线方式和移动平台上它们并没有提供足够的支持，比如在 Java ME 平台上使用 Google 地图 API 或是在 Android，iPhone 平台上使用 Bing 地图。如在中国，想在 Windows Mobile 平台上使用 MapABC 地图 API 都不是很方便。

引路蜂地图开发包提供了对几乎所有软件平台（桌面和移动平台）的支持，采用统一的接口使用包括 Google 地图, Bing 地图，MapABC 地图在内多达二十多种地图类型，并在同一开发包中支持离线地图，矢量地图显示。

下图显示了引路蜂地图开发包支持的开发平台，软件包以三个不同的开发包提供对各个平台的支持：

- Java ME 地图开发包 MIDP/LWUIT/Blackberry
- Java SE 地图开发包 Android/Java SE
- .Net Framework 地图开发包 Windows Mobile/Windows Phone 7/.Net desktop/Mac OS/iPhone/iPad



引路蜂地图开发包支持的主要功能如下表：

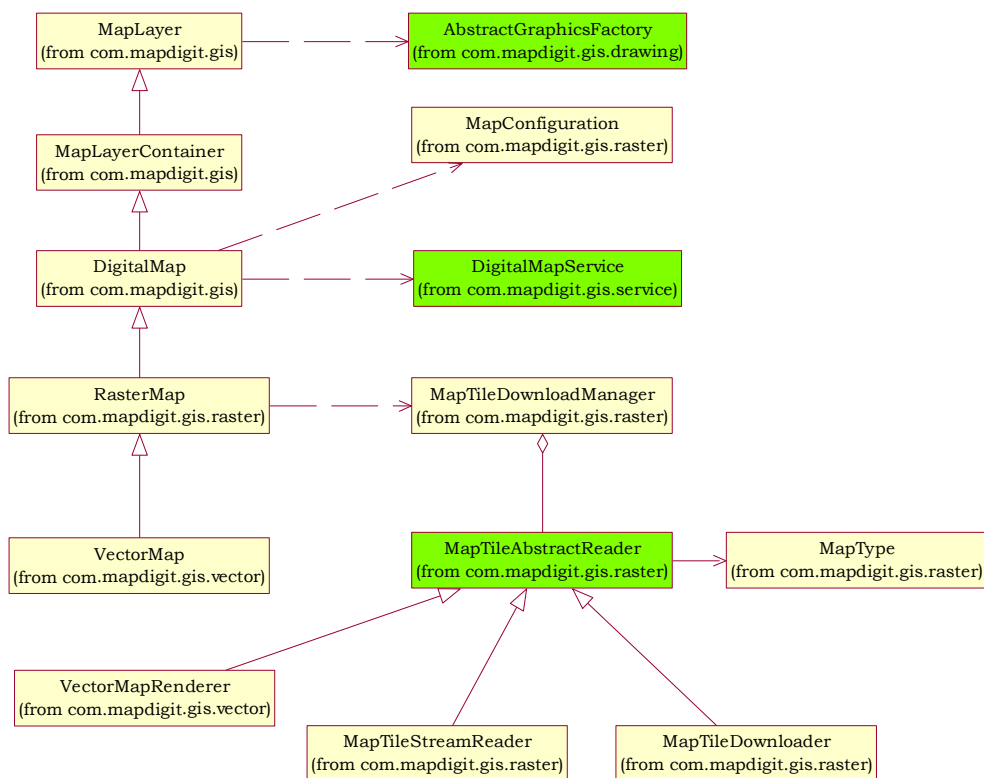
引路蜂地图开发包功能集	Java ME 平台			Java SE 平台		.Net Framework 平台	
	MIDP√	LWUIT√	Blackberry √	Android √	Java SE √	Mobile √	Desktop√
平台 UI 界面支持	√			√		√	
地图平移	√			√		√	
地图缩放	√			√		√	
地址编码	√			√		√	
地址逆编码	√			√		√	
路径查询	√			√		√	
本地搜索	√			√		√	
IP 地址查询	√			√		√	
经纬度查询	√			√		√	
GPS 位置查询	√			√		√	
小区位置查询	√			√		√	
Google 地图服务	√			√		√	
MapAbc 地图服务	√			√		√	
CloudMade 地图服务	√			√		√	
其它地图服务扩展	√			√		√	
矢量地图	√			√		√	
离线地图	√			√		√	
多种地图类型	√			√		√	
自定义地图类型	√			√		√	
其它 UI 界面扩展	√			√		√	
二维图形库	√			√		√	
示例代码	√			√		√	
开发文档	√			√		√	
试用许可	√			√		√	
商用许可	√			√		√	

注：对于 Mac OS, iPhone,iPad 的支持是通过 Mono C#支持，.Net Mobile 包括 Windows Mobile 6.0-6.5 和 Windows Phone 7. 这些版本目前正在测试中，将会在不久的将来发布。

2.0 地图开发包类定义

2.1 开发包中核心类定义

下图为引路蜂地图开发包中关键的几个类定义及其关系图：

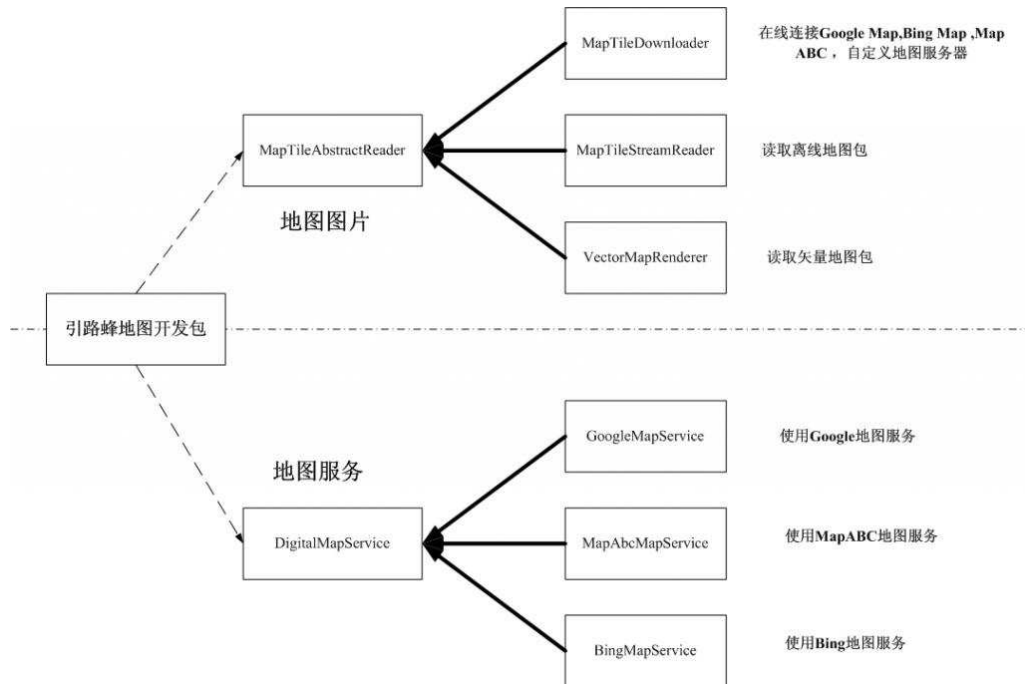


上图中显示为绿色的三个类分别用于对不同平台 UI,不同地图服务, 不同地图图片的支持, 引路蜂地图开发包设计时对地图图片和地图服务两部分进行了抽象, 其中地图图片主要用于显示地图图片, 地图图片可以来源于三个方面

- 在线地图, 可以连接包括 Google Map, Bing Map 在内的二十多种地图, 并支持自定义地图。
- 离线地图, 可以读取本地地图包。
- 矢量地图, 可以读取 MapInfo 格式 (需使用工具转换) 矢量地图。

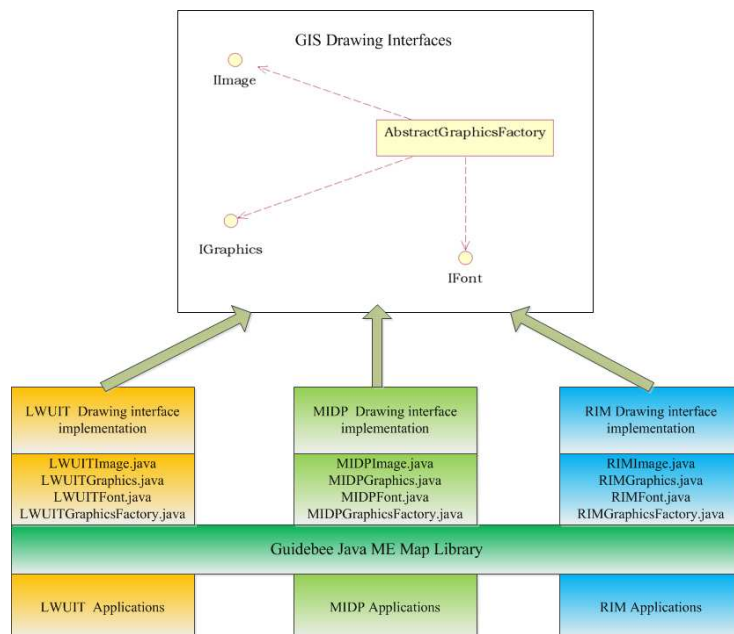
而地图服务则指地址查询, 路径查询, 本地查询等。引路蜂地图开发包支持谷歌地图服务, 在中国也同时支持 MapABC, CloudMade, Bing 地图 (地图服务也可以扩展)

两个部分相对独立, 也就是说你可以实现在显示 Bing Map 的同时, 使用谷歌地图服务, 而在显示 Bing Map 中国地图时, 使用 MapABC 地图服务。给开发人员以很大的自由选择余地。



引路蜂开发包之所以能够实现跨平台是因为对不同平台的图形系统进行了抽象。在 GIS.Drawing 包中定义了 IImage, IGraphics, IFont, AbstractGraphicsFactory 抽象类和接口。这样对于不同的平台只要实现相应接口，就可以使用同一个开发包。

比如在 java ME 平台上，程序员可以使用 midp.lcdui 或是 Lwuit 或是 Blackberry UI framework.



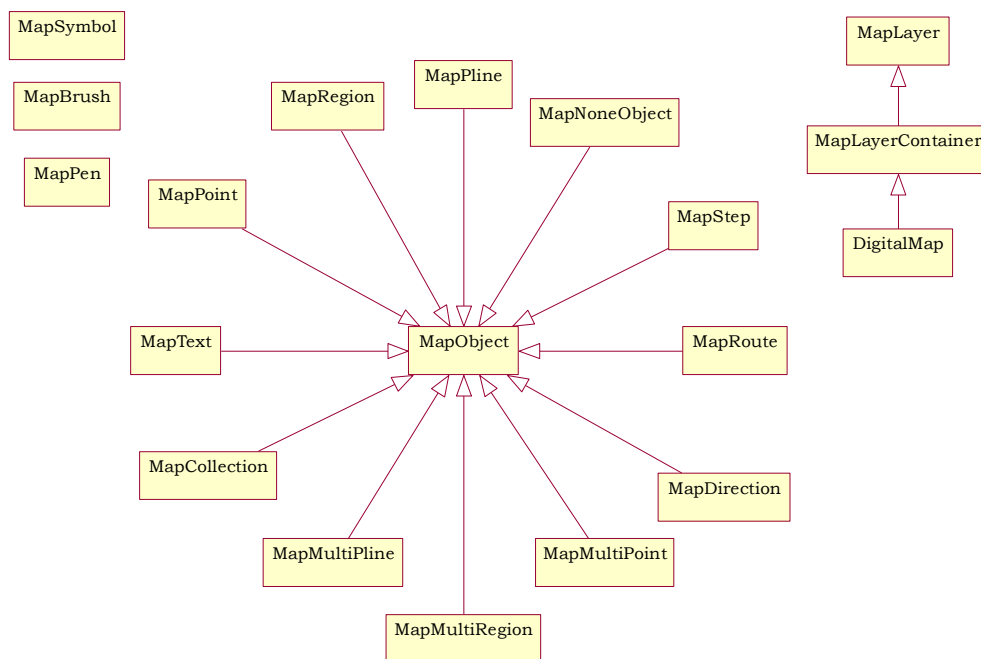
引路蜂地图开发包主要以 Java ,C# (.Net) 两种语言提供，尽管 Java 和 C#是两种不同的语言，但开发包提供的两种语言的开发接口却非常类似，熟悉一种就很快会知道如何在另一个平台上开发。

下表列表出了引路蜂地图开发包所定义类包。

包名称	Java 包	.Net 包
Gis	com.mapdigit.gis	Mapdigit.Gis
Gis.Geometry	com.mapdigit.gis.geometry	Mapdigit.Gis.Geometry
Gis.Drawing	com.mapdigit.gis.drawing	Mapdigit.Gis.Drawing
Gis.Raster	com.mapdigit.gis.raster	Mapdigit.Gis.Raster
Gis.Service	com.mapdigit.gis.service	Mapdigit.Gis.Service
Gis.Vector	com.mapdigit.gis.vector	Mapdigit.Gis.Vector
Gis.Location	com.mapdigit.gis.location	Mapdigit.Gis.Location
Gis.Navigation	com.mapdigit.gis.navigation	Mapdigit.Gis.Navigation
Drawing	com.mapdigit.drawing	Mapdigit.Drawing
Drawing.Geometry	com.mapdigit.drawing.geometry	Mapdigit.Drawing.Geometry
Util	com.mapdigit.util	Mapdigit.Util
Licence	com.mapdigit.licence	Mapdigit.Licence

2.2 Gis 包定义

Gis 包中定义了一些通用的地图对象类:



MapDirection	定义路径查询结果，路径查询结果可以由多条路径组成。
MapObject	所有地图对象的基类。
MapPoint	地理几何图形点对象。
MapRoute	定义了单个路径对象。
MapStep	定义路径上单个步骤的详细信息，比如向左，向右等。
DigitalMap	所有地图对象的基类。
MapSymbol	地图点对象的类型。
MapPen	用于来绘制地图对象边框的笔型定义。
MapBrush	用于来绘制地图对象内部的画笔定义
MapPline	定义地图对象中一条多义线。
MapRegion	定义地图对象中一条多边形。
MapMultiPoint	地理地图对象点对象集合。
MapMultiPline	定义地图对象中多义线集合。

MapMultiRegion	定义地图对象中多边形集合.
MapCollection	定义地图对象集合.
MapNoneObject	定义地图对象中没有几何信息的对象。
MapText	定义地图对象中图形文字对象。
MapLayer	定义地图层.
MapLayerContainer	定义地图层的集合.

2.3 Gis.Geometry 包定义

该包定义了地图对象中各种地理几何图形.

GeoPoint

GeoBounds

GeoLatLng

GeoLatLngBounds

GeoPolyline

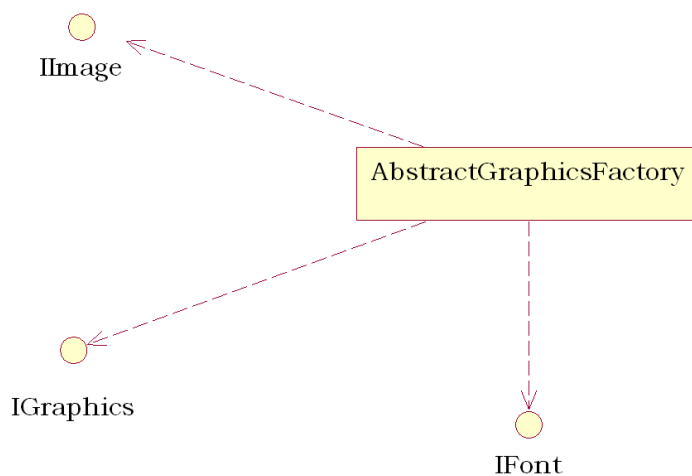
GeoSize

GeoPolygon

GeoBounds	定义地图对象的边界矩形。
GeoLatLng	点义一个经纬度坐标.以 latitude,longitude 表示。
GeoLatLngBounds	点义一个经纬度为单位的地图对象的边界矩形。
GeoPoint	定义平面坐标系中一个点，以 x,y 表示。
GeoPolygon	定义地理几何图形中一个多边形.
GeoPolyline	定义地理几何图形中一个多义线.
GeoSize	定义一个几何图形矩形的大小.

2.4 Gis.Drawing 包定义

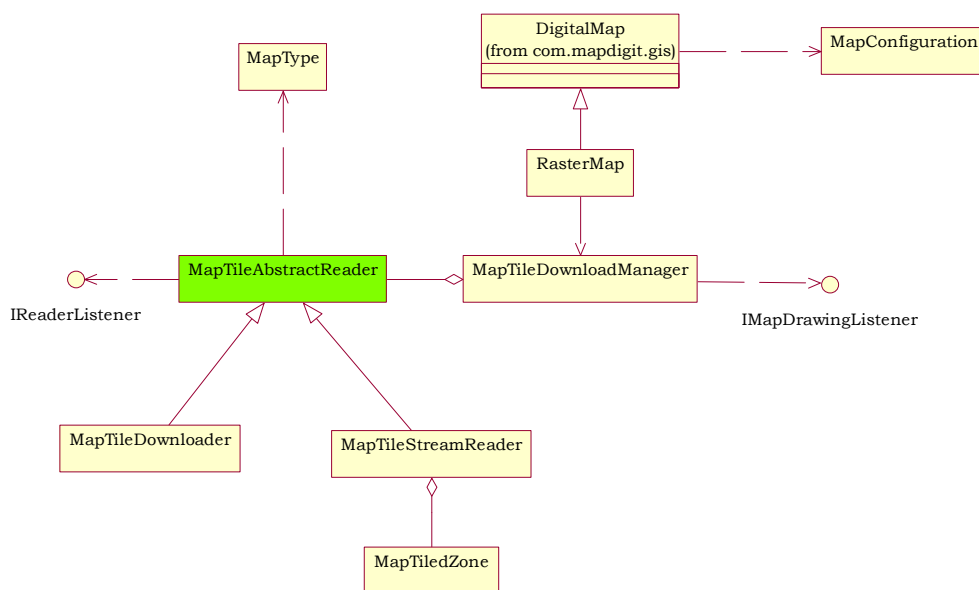
如前文所述，该包对不同平台 UI 部分进行了抽象，定了图形系统的接口，在不同平台上使用地图名，只要实现这些接口，就可以在该平台上使用地图 API，在发布的地图名包含了对这些接口的缺省实现，不同平台，这些接口的实现方法不尽相同。



AbstractGraphcsFactory	抽象图形类工厂，主要用于创建新的图象对象.
IGraphics	Graphics 接口定义，定义了用于图形绘制的一些方法。
IFont	Font 定义，如果不使用矢量地图，可以不实现该类。
IImage	Image 图象接口，定义了取得图象大小等方法。

2.5 Gis.Raster 包定义

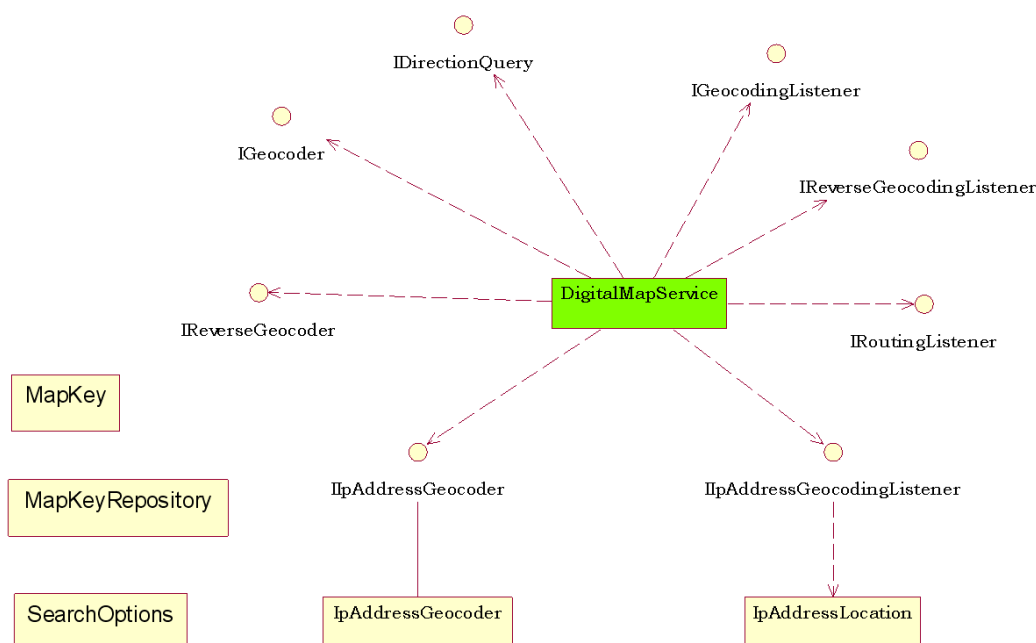
该包定义了在线和离线地图对象相关的一些类.



Map Type	定义各种地图类型，如 Google 地图，Google 卫星图，Bing 中国地图等。
MapConfiguration	定义地图的一些配置信息，如是否使用缓存，同时启动最大工作线程数目等。
RasterMap	显示在线地图和离线地图一个关键类.它是地图开发包最核心的一个类，实现地图平移，缩放，路径，查询等各种功能。
MapTileAbstractReader	一个抽象类定义了如何取得地图图片的方法.从它可以派生出对离线地图，矢量地图以及在线从地图服务器上读取地图图片以用于显示地图。
MapTileDownloader	实现从服务器（如 Google Map,Bing Map 以及自定义服务器）在线下载地图图片功能。是 MapTileAbstractReader 一个子类。
MapTileStreamReader	实现离线读取本地地图包中地图图片，也是 MapTileAbstractReader 的一个子类。
MapTileZone	定义离线地图中一个区域，这是对离线地图包一个缺省实现，开发人员也可以自定义离线地图包格式。
IReaderListener	一个用于获取下载进度的回调函数接口。
IMapDrawingListener	当在线下载，离线读取或矢量地图图片生成器完成时的的回调函数接口。

2.6 Gis.Service 包定义

这个包中定义了地址查询，地理反编码，路径查询，本地查询等地图服务的接口。

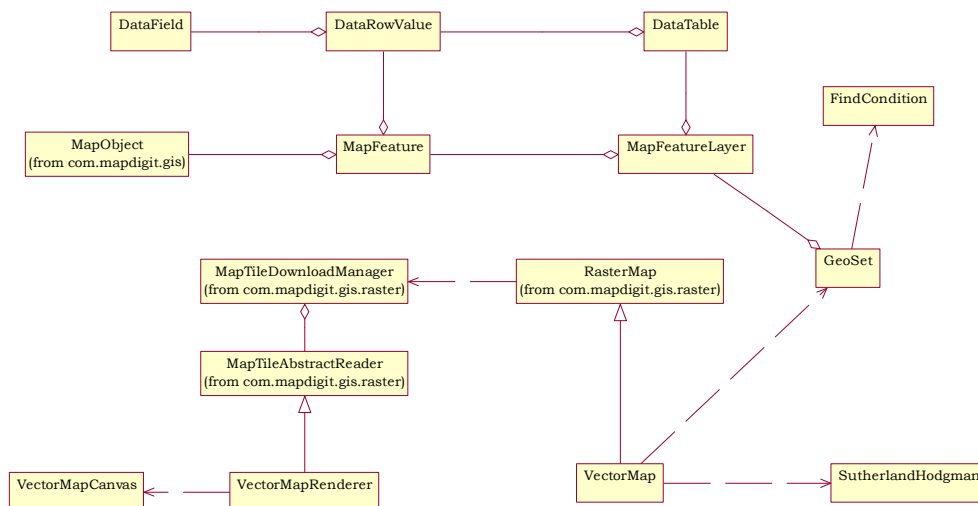


DigitalMapService	各种地图服务（地址查询，路径）的实现。开发人员可以选择使用地图服务的类型，如使用 Google 地图或是 CloudMade 地图服务，在中国的用户还可以选择使用 MapAbc 地图服务。
IDirectionQuery	路径查询接口定义.
IGeocoder	地址查询接口定义.
IReverseGeocoder	根据经纬度查询地名的地理逆编码查询接口定义.
IIpAddressGeocoder	根据 IP 地址查询地名的查询接口定义.
IGeocodingListener	地址查询服务回调函数接口.
IReverseGeocodingListener	地理逆编码查询服务回调函数接口.
IRoutingListener	路径查询服务回调函数接口.
IIpAddressGeocodingListener	IP 地址查询服务回调函数接口.
IPAddressGeocoder	该类定义 IP 地址查询服务.
SearchOptions	使用查询服务一些配置，如查询结果返回数目，路径长度单位为公里或是英里，语言配置等。
MapKey	不同的地图服务可能使用不同的 Map Key,如果商用，需要向不同公司申请地图 Key,如 Google Map Key, MapAbc

	Key,CloudMade Map Key.
MapKeyRepository	用于管理各种 Map Key 并为地图服务选择合适的 Map Key。

2.7 Gis.Vector 包定义

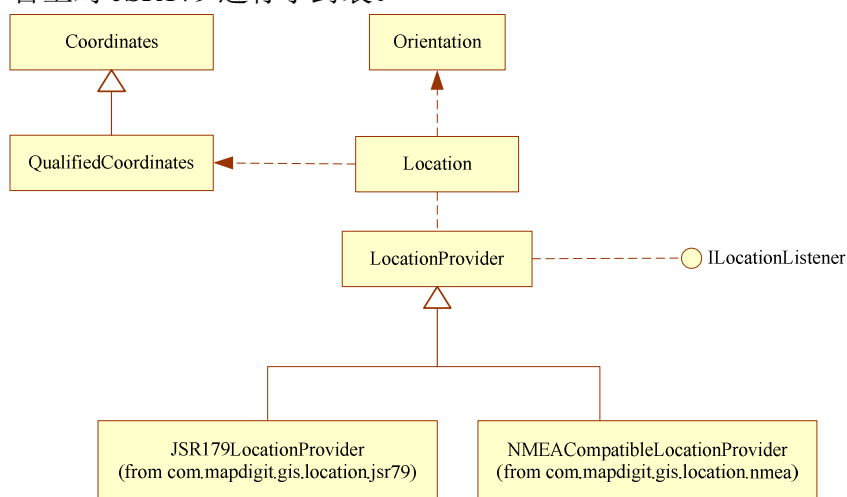
该包定义了矢量地图所用一些类.



SutherlandHodgman	实现 Sutherland-Hodgman 剪裁算法.
VectorMap	矢量地图类，它是 RasterMap 的一个子类。矢量地图可以由多个图层组成。
GeoSet	多个地图图层的集合，并定义了图层之间的层次关系。
MapFeatureLayer	单个地图图层，一个图层可由地图点对象，或是多义线，或是多边形组成。
MapFeature	一个地图对象。
DataTable	定义一个数据表。
DataField	定义数据表格的一个字段。
DataRowValue	定义数据表一个行数据。
FindCondition	查询条件。
FindConditions	查询条件的集合。

2.8 Gis.Location 包定义

本包定义了 GPS 接收器一个通用接口，并提供对 NMEA 2.0 数据的解码方法。在 Java ME 平台上对 JSR179 进行了封装。



Coordinates	定义地址经纬度坐标。
Location	定义地址信息，如名称等。
LocationProvider	地理信息提供者，是所有 GPS 设备的基类。
Orientation	提供当前 GPS 坐标中的方向信息。
QualifiedCoordnates	提供当前 GPS 坐标中的详细方向信息，包括精度。
JSR179LocationProvider	JSR179 API locationProvider 封装类，只在 Java ME 平台提供。
NMEACompatibleLocationProvider	NMEA 兼容设备的基类，提供对 NMEA 2.0 数据解码。

2.9 Gis.Navigation 包定义

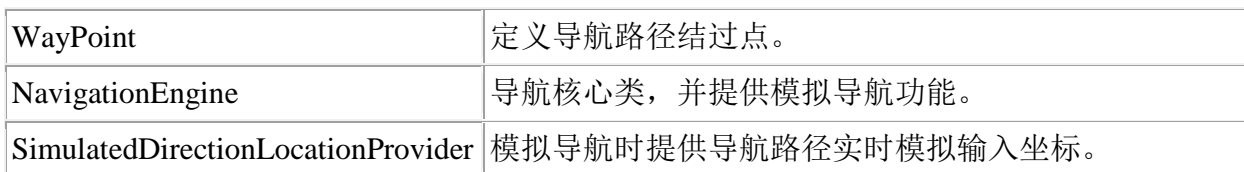
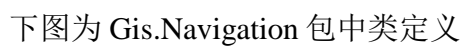
本包提供了路口到路口实时导航 API，从地图服务器返回的路径信息含有文字和路径的地理坐标信息，类 NavigationEngine 根据路径和当前坐标实现实时导航。它内部含用三个工作线程：

Location Monitor 实时监控当前位置坐标是否偏离路径，如果偏移，决定是否需要重新计算路径。

语音命令生成 根据当前路径和位置预先生成语音命令列表。

语音命令处理 从预先生成语音命令列表选取一个命令并通知回调函数。

下图是 Navigation Engine 的一个 简明状态机转换图



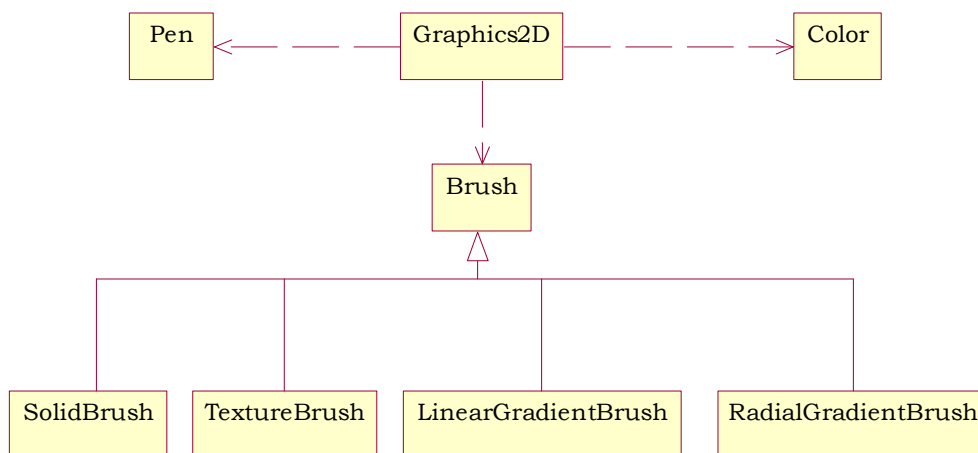
VoiceCommandArg	语音命令参数。
VoiceCommandSpeaker	播放语音。
VoiceCommandType	语音命令类型。
WalkOnRoute	当前在路径上的位置。
INavigationListener	导航回调函数接口。
IVoiceCommandListener	语音命令回调函数接口。

注：导航功能是否存在请参看开发包中功能定义。

2.10 Drawing 包定义

定义平台无关二维图形库类定义。其功能和 Java SE 中 Graphics2D 或是 .Net 中 System.Drawing 功能类似。

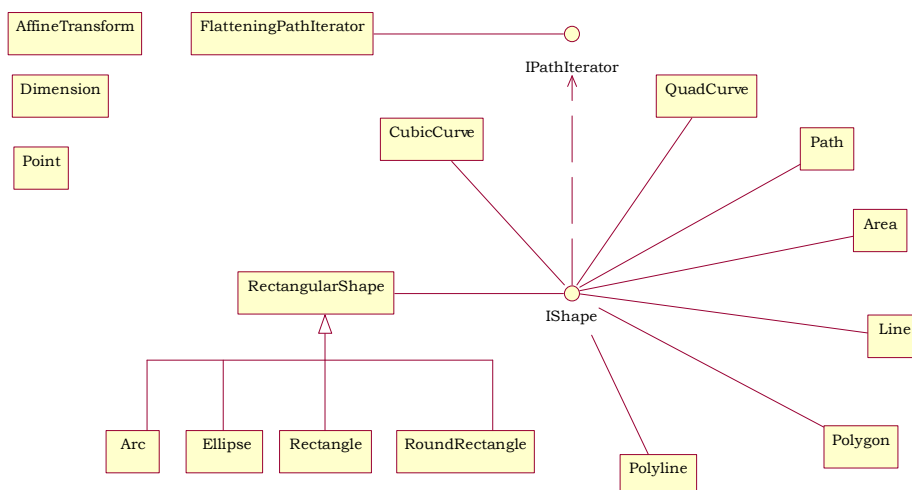
提供 Drawing 包的的主要原因是部分移动平台自带的二维图形库功能比较弱。比如 Java ME 平台上不支持线宽大于 1 的画笔等等。



Brush	画刷类。
Color	颜色类。
Graphics2D	Graphics2D 提供各种绘图功能。
LinearGradientBrush	线性渐变画刷类。
Pen	画笔类。
RadialGradientBrush	环状渐变画刷类。
SolidBrush	单色画刷类。
TextureBrush	材质画刷类。

2.11 Drawing.Geometry 包定义

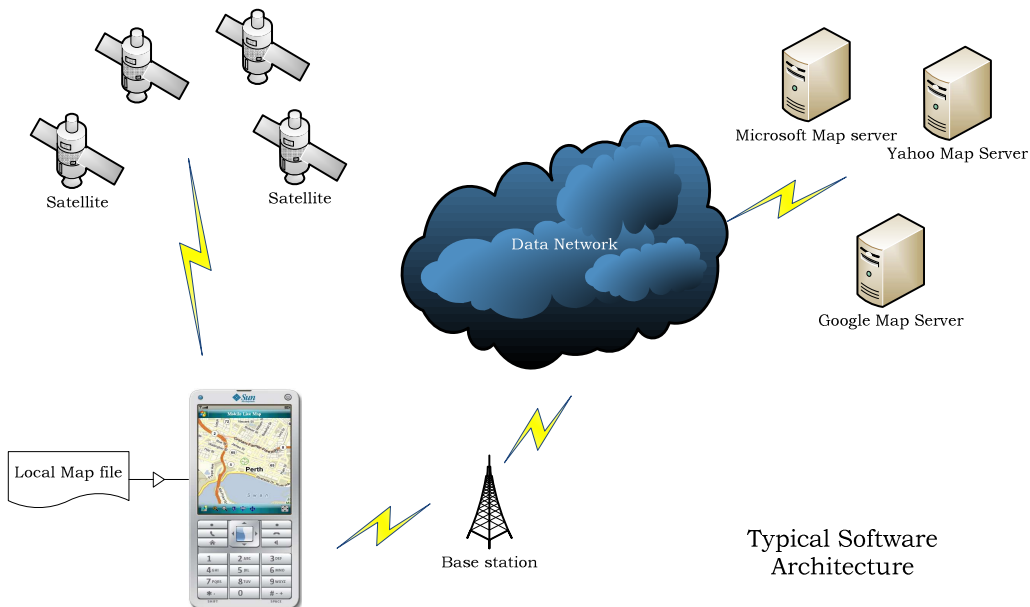
本包中定义了各种二维几何图形。基本上是参照 Java SE 中相应类来定义实现的。



PathIterator	路径元素枚举类。
IShape	Shape 接口定义了二维图形的一些基本方法。
AffineTransform	仿射变换类。
Arc	圆弧类。
Area	二维区域类。
CubicCurve	三次曲线类。
Dimension	二维图形大小类。
Ellipse	椭圆类。
FlatteningPathIterator	平滑变换路径枚举类。
Path	路径类。
Line	线段类。
Point	二维点。
Polygon	多边形。
Polyline	多义线。
QuadCurve	四次曲线类。
Rectangle	矩形类。
RectangularShape	具有矩形外框几何图形的基类。
RoundRectangle	圆角矩形类。

2.12 典型软件结构图

下图是基于地图的应用一个典型软件构架图。

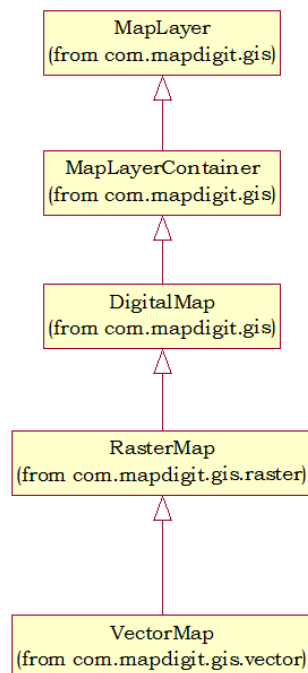


3.0 一些地图开发包基本知识

引路蜂地图开发包对在线，离线，矢量地图提供了统一的接口 API。

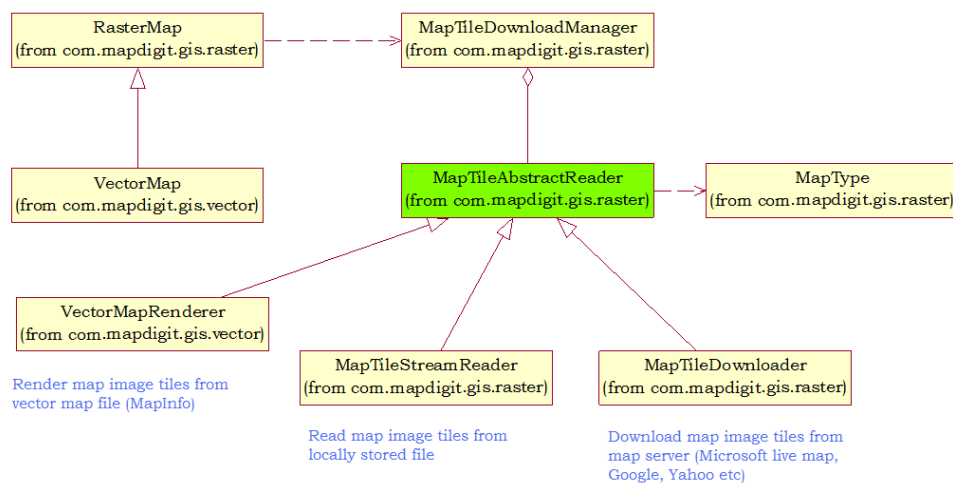
3.1 地图对象类层次关系

- MapLayer 定义了通用的地图操作，如设置地图中心点，放大，缩小，移动，以及坐标变换（从屏幕到地图坐标或反向），它也是所有地图类的基类。
- MapLayerContainer 为一个集合类，它定义了一组 MapLayer 的集合及它们的上下层关系，并且使这些 MapLayer 一起移动，放大或缩小等。
- DigitalMap 定义了地图服务，如地址查询，路径查询，它包含了对不地图服务的支持。
- RasterMap 定义了对在线地图，离线地图的支持，内部支持多线程获取地图图片并提供回调函数用于了解地图图片下载进度等。
- VectorMap 为 RasterMap 一个子类，主要提供类似 MapInfo 的矢量地图支持。矢量地图通过读取地图矢量数据并将其栅格化成图象显示。

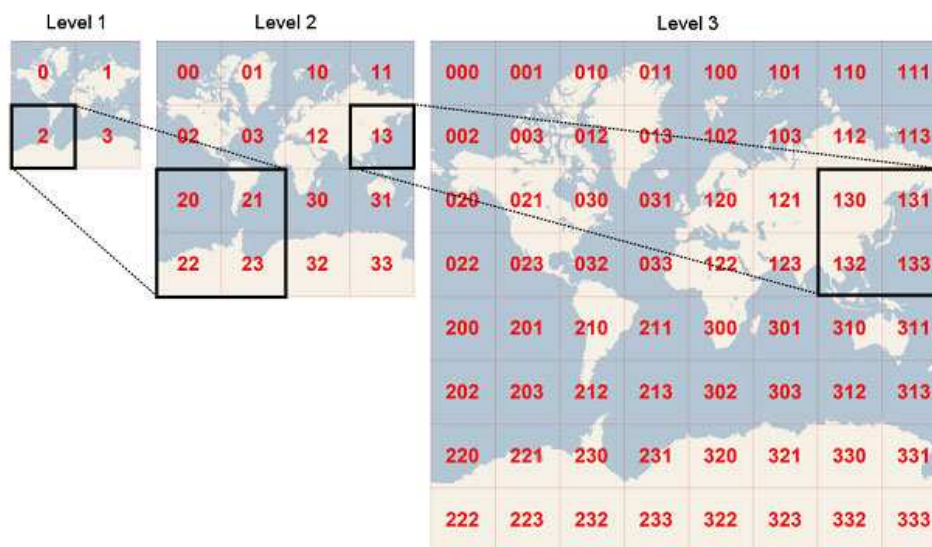


3.2 地图图片读取类层次关系

不管是在线地图，离线地图还是矢量地图，对于 RasterMap 来说最终都是一张张 256x256 大小的图象，在线地图是从地图服务器上下载图片，离线地图是从本地地图包中读取，而矢量地图是将矢量数据栅格化处理。



RasterMap 中含有一个 MapTileDownloadManager 实例，这个对象用于管理所用的地图图片下载，读取或是栅格化工作线程。这些线程都是 MapTileAbstractReader 的子类。MapTileDownloader 用于在线地图下载地图图片，MapTileStreamReader 从本地地图文件流中读取地图图片，而 VectorMapRenderer 则是栅格化矢量地图数据。大部分商用的地图服务器如 Google 地图，Bing 地图等都采用了类似的地图分片方法，将地图分成 0 - 17 级，每张地图图片都是大小为 256 个像素的正方形图片，下图显示 1 级到 3 级地图分级方法。



3.3 地图服务提供商

DigitalMap 类中带有对 DigitalMapService 类的一个引用, Digital Map Service 定义了地址编码, 地址反编码, 路径查询等地图服务。目前引路蜂地图包中实现了对 Google 地图服务, CloudMade 地图服务, MapAbc 地图服务的支持, 缺省采用 Google 地图服务, 对其它地图如 Bing 地图也可以扩展实现。

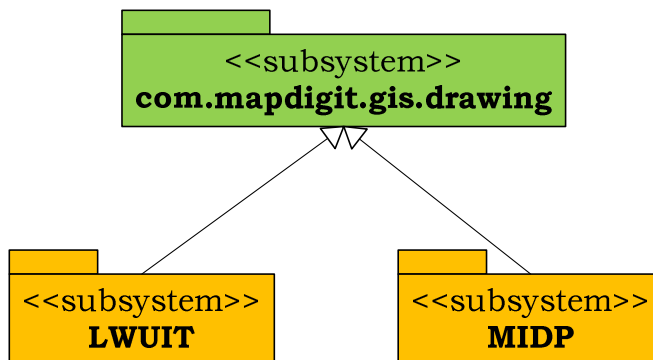


注：您需要从不同地图服务提供商申请 Map Key 来合法使用这些地图服务。引路蜂地图开发包中 MapKey, MapKeyRepository 类用来管理这些 Map Key。

3.4 图形子系统

如前文所述，引路蜂地图开发包中 Gis.Drawing 类定义了几个抽象类或接口，对不同平台的图形系统进行了抽象。以 Java ME 平台为例，可以使用 Sun 的 LWUIT 或是 MIDP 自带的 javax.microedition.lcudi 来绘制几何图形或是图象。这些绘图接口不同，如果在引路蜂地图开发包直接引用这些类方法，这样不同平台就会需要编译不同的开发包。通过抽象处理，引路蜂地图开发包并直接调用平台相关的图形类方

法。而是通过 开发包 + 平台相关图形系统实现的方法，就可以实现跨平台。下面是 Java ME 平台的一个例子。



通过对 LWUIT 和 MIDP 图形系统接口的实现，同一个库可以用于不同的平台。



LWUIT Hello world demo
Microsoft Live Map



MIDP Hello world demo
Google Map

实际上 Java ME 平台上同一个库也可以用于 Blackberry 应用，Blackberry 除了支持 MIDP 之外，还定义自己的 UI Framework，这是只要实现在 Blackberry 的 UI Framework 上的 Gis.Drawing 接口就可以开发 Blackberry 应用（非 Java ME Midlet 应用）。

Gis.Drawing 定义了地图开发包中所需要的最小功能集，如果你开发的应用中不需要使用矢量地图，则与矢量地图相关的一些类或方法则可以为空。

这里给出 Gis.Drawing 中定义的类或方法详细列表，是一个非常小的功能集合，也就是说如果想支持新的平台，实现这些方法是一个非常简单的任务。

在 2.4 中定义了

AbstractGraphicsFactory	抽象图形类工厂，主要用于创建新的图象对象。
IGraphics	Graphics 接口定义，定义了用于图形绘制的一些方法。
IFont	Font 定义，如果不使用矢量地图，可以不实现该类。
IImage	Image 图象接口，定义了取得图象大小等方法。

类 AbstractGraphicsFactory 定义

Java 定义

方法名	描述
abstract public IImage createImage(int[] rgb, int width, int height);	根据 int []数组创建 IImage 对象。 Rgb 为一个存取每个像素值的整数数组。 为 ARGB 格式。 Width,height 为要创建图形的大小。
abstract public IImage createImage(byte[] bytes, int offset, int len);	从一个字节数组中创建 IImage 对象。这引数组为一般为图象的 png 文件数据。
abstract public IImage createImage(int width, int height);	创建一个大小为 width ,height 的空 IImage 对象。

C# 定义

方法名	描述
public abstract IImage CreateImage(int[] rgb, int width, int height);	根据 int []数组创建 IImage 对象。 Rgb 为一个存取每个像素值的整数数组。 为 ARGB 格式。 Width,height 为要创建图形的大小。
public abstract IImage CreateImage(byte[] bytes, int offset, int len);	从一个字节数组中创建 IImage 对象。这引数组为一般为图象的 png 文件数据。
public abstract IImage CreateImage(int width, int height);	创建一个大小为 width ,height 的空 IImage 对象。

接口 IGraphics 定义

Java 定义

方法名	描述
public void setClip(int x, int y,int width,int height)	设置绘制剪裁区域，可以空实现。
public void drawImage(Image img, int x, int y);	在位置 (x,y) 处绘制 img。
public void drawLine(int x1, int y1, int x2, int y2)	绘制一条线段
public void setColor(int RGB);	设置颜色
public void fillRect(int x, int y, int width, int height);	填充矩形
public void drawRect(int x,int y,int width, int height);	绘制矩形
public void drawString(String str,int x,int y);	显示字符串，只在矢量地图时需要。
public void setFont(IFont font);	设置字体，只在矢量地图时需要。
public void fillTriangle(int x1, int y1, int x2, int y2, int x3, int y3);	填充三角形，只在矢量地图时需要。

C# 定义

方法名	描述
void SetClip(int x, int y, int width, int height);	设置绘制剪裁区域，可以空实现。
void DrawImage(Image img, int x, int y);	在位置 (x,y) 处绘制 img。
void DrawImage(Image img, int x, int y,int transparentColor);	在位置 (x,y) 处绘制 img，只在 windows mobile 时需要，transparentColor 为透明色。其它时候，调用 void DrawImage(Image img, int x, int y)即可。
void DrawLine(int x1, int y1, int x2, int y2);	绘制一条线段
void SetColor(int rgb);	设置颜色
void FillRect(int x, int y, int width, int height);	填充矩形
void DrawRect(int x, int y, int width, int height);	绘制矩形
void DrawString(string str, int x, int y);	显示字符串，只在矢量地图时需要。
void SetFont(IFont font);	设置字体，只在矢量地图时需要。

IFont 类定义，IFont 只在使用矢量地图时需要定义，其它情况可以使用空实现。

Java 定义

方法名	描述
public Object getNativeFont();	返回平台相关字体对象。
public int charsWidth(char[] ch, int offset, int length);	返回指定字符串宽度。

C#定义

方法名	描述
object GetNativeFont();	返回平台相关字体对象。
int CharsWidth(char[] ch, int offset, int length);	返回指定字符串宽度。

IImage 类定义

Java 定义

方法名	描述
public IGraphics getGraphics();	取得该图象对象的相关画板类，用于在该图象上绘制图形。
public int[] getRGB();	返回该图象的 int[] 数组，只在矢量地图时需要
public int getHeight()	返回图象高度
public int getWidth();	返回图象宽度
public Object getNativeImage();	返回平台相关图象对象

C#定义

方法名	描述
IGraphics GetGraphics();	取得该图象对象的相关画板类，用于在该图象上绘制图形。
int[] GetRGB();	返回该图象的 int[] 数组，只在矢量地图时需要
int GetHeight();	返回图象高度
int GetWidth();	返回图象宽度
object GetNativeImage();	返回平台相关图象对象

4.0 准备工作

引路蜂地图开发包提供对大多数平台的支持，在不同平台编写应用时所用的工具和设计语言不尽相同，引路蜂地图开发包主要以 Java 和 C# 支持各个平台，在下面的使用说明上主要以 Java 语言为例，但基本概念同样适合于 C# 支持的平台。如有不同点，将特别指出。另外可以直接参照各个平台下的代码示例。

4.1 引路蜂地图开发包使用许可

使用引路蜂地图开发包需要合法的使用许可，使用许可（Licence）分为试用和商用两种，试用许可可免费。如果您需要发布你的地图应用软件，则需要商用许可。

开发包使用试用许可时，在地图会显示“Guidebee”水印，有些功能可能不提供。从版本 2.1 开始，试用版取消了时间上的限制。

商用许可需要付费后使用，具体可以参见我们的网站。<http://www.guidebee.biz>或是发信到 james.shen@guidebee.biz 询问。

4.2 Google, CloudMade, MapAbc 地图 Map Key

引路蜂地图开发包提供了对 Google 地图，CloudMade 地图服务以及 MapAbc 地图服务的支持，这些地图服务提供商有自己的地图服务作用许可，您可能需要申请他们的 Map Key 来合法使用他们的服务。引路蜂地图开发包中的 MapKey 和 MapKeyRepository 用来管理这些 Map Key，但在开始时您可以使用开发包中自带的 Map Key 来开发调试您的地图应用。

例如：下面代码在 MapKeyRepository 增加你自己的 Google Map Key。

```
MapKeyRepository.addMapKey(MapKey.MAPKEY_TYPE_GOOGLE,yourgoogleKey);
```

4.3 地图配置

地图配置主要是通过类 MapConfiguration 和 SearchOptions 来设置。

这里给出最常用的几个配置。

4.3.1 工作线程数

引路蜂地图开发包允许用户配置同时工作的最大工作线程数，这些工作线程主要用来从地图服务器上下载地图图片。缺省设置为 4。

这是通过 MapConfiguration.WORKER_THREAD_NUMBER 来配置的。

例：下面将工作线程数设置 16。

```
MapConfiguration.setParameter(MapConfiguration.WORKER_THREAD_NUMBER, 16);
```

4.3.2 使用缓存

引路蜂地图开发包缺省是使用缓存来存放最近下载的图片以提高地图显示效率，但缓存的使用会占用一定的内存，大小可以通过配置，缺省大小为 2 5 6 K。如果设备内存比较小，也可以关闭缓存。

大小是通过 `MapConfiguration. MAP_CACHE_SIZE_IN_BYTES` 来配置，开启关闭缓存是通过 `MapConfiguration. IS_CACHE_ON` 来设置。

4.3.3 查询配置

在使用地图服务查询地址，路径时的一些配置主要通过 `SearchOptions` 在设置，可以设置结果返回的语言，每次返回结果最大数等。

指定语言时采用 语言编码来设置。

例如：下面指定语言为繁体中文。

```
DigitalMapService.getSearchOptions().LanguageID="zh-tw";
```

4.4 第一个地图应用

这里以 Java ME 平台为例，如果您不了解 Java ME 平台也没关系，只要注重蓝色部分代码，其它基本上是 Java ME 应用程序框架。

下面的代码显示中国南京地图。

```
//----- IMPORTS -----
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Canvas;
import javax.microedition.lcdui.Graphics;
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.Display;

import com.mapdigit.gis.MapLayer;
import com.mapdigit.gis.drawing.IGraphics;
import com.mapdigit.gis.drawing.IImage;
import com.mapdigit.gis.geometry.GeoLatLng;
import com.mapdigit.gis.raster.IMapDrawingListener;
import com.mapdigit.gis.raster.IReaderListener;
import com.mapdigit.gis.raster.MapTileDownloadManager;
import com.mapdigit.gis.raster.MapType;
import com.mapdigit.gis.raster.RasterMap;

import com.mapdigit.licence.LicenceManager;

import com.pstreets.gisengine.demo.midp.drawing.MIDPGraphicsFactory;

//[----- MAIN CLASS -----]
/**
 * Hello China demo.
 * <hr><b>&copy; Copyright 2011 Guidebee, Inc. All Rights Reserved.</b>
 * @version 1.00, 04/01/11
 * @author Guidebee Pty Ltd.
 */
public class HelloChina extends MIDlet implements IReaderListener,
```

```

    IMapDrawingListener {

        protected RasterMap map;
        protected MapTileDownloadManager mapTileDownloadManager;
        protected IImage mapImage;
        protected IGraphics mapGraphics;
        protected MapCanvas canvas;

        public HelloChina() {
            try {
                //setup the licence information
                (1) LicenceManager licenceManager = LicenceManager.getInstance();
                    long keys[] = {0x34ba283b8daeb659L, -0x53c811f9da86e998L,
                        -0x34ba25c3c581521eL, 0xf15df9fc7e45628L, 0x6a4ece44296c0287L,
                        0x4ab0cff532902blcL,};
                    licenceManager.addLicence("GuidebeeMap_JavaME", keys);
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }

        public void startApp() {
            canvas = new MapCanvas();
            //set the graphics factory
            (2) MapLayer.setAbstractGraphicsFactory(MIDPGraphicsFactory.getInstance());
                mapImage =
                MapLayer.getAbstractGraphicsFactory().createImage(canvas.getWidth(),
                    canvas.getHeight());
                mapGraphics = mapImage.getGraphics();
                //Create the Digital Map objects.
                (3) mapTileDownloadManager = new MapTileDownloadManager(this);
                    map = new RasterMap(1024, 1024, mapTileDownloadManager);
                    map.setScreenSize(canvas.getWidth(),
                        canvas.getHeight());
                (4) mapTileDownloadManager.start();
                    map.setMapDrawingListener(this);
                (5) GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
                    map.setCenter(center, 13, MapType.GOOGLECHINA);
                    Display.getDisplay(this).setCurrent(canvas);
        }

        public void pauseApp() {
        }

        public void destroyApp(boolean unconditional) {
            (6) mapTileDownloadManager.stop();
        }

        (9)public void readProgress(int downloaded, int total) {
            System.out.println(downloaded + "/" + total);
        }

        public void done() {
            if (canvas != null) {
                canvas.repaint();
            }
        }
    }

```

```

/**
 * Map canvas class, a subclass of Canvas.
 */
protected class MapCanvas extends Canvas {

    private void panMap(float x, float y) {
        float dx = x - oldX;
        float dy = y - oldY;
        if (!(dx == 0 && dy == 0)) {
            (7) map.panDirection((int) dx, (int) dy);
        }
    }

    boolean isPan = false;
    private float oldX = -1;
    private float oldY = -1;

    (8)protected void paint(Graphics g) {
        map.paint(mapGraphics);
        g.drawImage((Image) mapImage.getNativeImage(), 0, 0, 0);
    }

    public void pointerDragged(int x, int y) {
        if (isPan) {
            panMap(x, y);
            oldX = x;
            oldY = y;
        }
    }

    public void pointerPressed(int x, int y) {
        oldX = x;
        oldY = y;
        isPan = true;
    }

    public void pointerReleased(int x, int y) {
        oldX = x;
        oldY = y;
        isPan = false;
    }
}

```

1. 首先是设置使用许可（Licence），设置使用许可由二部分组成，一个是将 licence 文件(guidebee.lic)放在合适的目录下，然后在程序中输入正确的注册码。
2. 前文说过，引路蜂地图开发包不直接调用图形相关方法，而是定义了一些接口函数，在使用时，需实现这些平台相关的方法。例如在 Java ME 平台，MIDPFont，MIDPGraphics，MIDPGraphicsFactory，MIDPImage 实现了这些接口方法，在调用任务地图相关方法，需要告诉地图如何调用这些方法。
[MapLayer.setAbstractGraphicsFactory\(MIDPGraphicsFactory.getInstance\(\)\);](#)则是用来设置平台相关 GraphicsFactory 实现。

3. 创建 `MapTileDownloaderManager` 实例，`MapTileDownloaderManager` 是用来管理工作线程以下载地图图片或是从本地地图包中读取地图图片。
之后创建 `RasterMap` 地图实例，`RasterMap` 使用 `MapTileDownloaderManager` 来获取地图图片。
4. `MapTileDownloaderManager.Start` 启动工作线程。注意在程序结束时，需调用 `Stop` 方法来停止这些工作线程，如（6）
5. 设置地图中心点，这里设置是南京的经纬度坐标。地图类型选择是 **Google 中国** 地图，地图包中定义了多达 20 多种不同的地图可供选择。地图缩放级别为 1 3 级。1 级最小，17 级最大。
6. 程序结束时，需调用 `Stop` 方法来停止工作线程。
7. `panDirection(dx,dy)`用来平移地图，将在后面介绍。
8. 在屏幕上显示地图。在这个应用中 `MapCanvas` 最终用来在屏幕上显示地图，每个程序都有一个 UI 界面，地图最终总是要在屏幕上显示的。这两行代码是将地图复制到屏幕上。



9. 地图下载采取异步工作方式，也就是地图图片的读取是通过工作线程在后台工作，应用可以监视读取图片的进度 `readProgress`，每张地图图片下载完毕时，会使用消息 `Done` 通知应用，在 `Done` 消息处理中一般用来更新屏幕显示。

在后面的示例中，一些共同的代码不再重复，将设计一个基类 `MapDemoMIDP`，创建地图，设置使用许可等代码都放在基类中。

```
public abstract class MapDemoMIDP extends MIDlet implements IReaderListener,
    IMapDrawingListener {

    protected RasterMap map;
    protected MapTileDownloadManager mapTileDownloadManager;
    protected IImage mapImage;
    protected IGraphics mapGraphics;
    protected MapCanvas canvas;
```

```

public MapDemoMIDP() {
    try {
        LicenceManager licenceManager = LicenceManager.getInstance();
        long keys[] = {-0x8b9d5bd05f46354L, -0x4e41fd81b7d38cc3L, 0x5c7addc8022e908aL,
            -0x532dae1d56b062b2L, 0x3e18d8181cb217ecL, -0x39a3d8deb40bd92fL,};
        licenceManager.addLicence("GuidebeeMap_JavaME", keys);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

public void init() {
    canvas = new MapCanvas();
    //set the graphics factory
    MapLayer.setAbstractGraphicsFactory(MIDPGraphicsFactory.getInstance());
    mapImage = MapLayer.getAbstractGraphicsFactory().createImage(canvas.getWidth(),
        canvas.getHeight());
    mapGraphics = mapImage.getGraphics();
    //Create the Digital Map objects.
    mapTileDownloadManager = new MapTileDownloadManager(this);
    map = new RasterMap(1024, 1024, mapTileDownloadManager);
    map.setScreenSize(canvas.getWidth(),
        canvas.getHeight());
    mapTileDownloadManager.start();
    map.setMapDrawingListener(this);
}

public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
    mapTileDownloadManager.stop();
}

public void readProgress(int arg0, int arg1) {
    System.out.println(arg0 + "/" + arg1);
}

public void done() {
    if (canvas != null) {
        canvas.repaint();
    }
}

/**
 * Map canvas class, a subclass of Canvas.
 */
protected class MapCanvas extends Canvas {

    private void panMap(float x, float y) {
        float dx = x - oldX;
        float dy = y - oldY;
        if (!(dx == 0 && dy == 0)) {
            map.panDirection((int) dx, (int) dy);
        }
    }

    boolean isPan = false;
    private float oldX = -1;
    private float oldY = -1;
}

```



```
protected void paint(Graphics g) {
    map.paint(mapGraphics);
    g.drawImage((Image) mapImage.getNativeImage(), 0, 0, 0);
}

public void pointerDragged(int x, int y) {
    if (isPan) {
        panMap(x, y);
        oldX = x;
        oldY = y;
    }
}

public void pointerPressed(int x, int y) {
    oldX = x;
    oldY = y;
    isPan = true;
}

public void pointerReleased(int x, int y) {
    oldX = x;
    oldY = y;
    isPan = false;
}
}
```

5.0 地图基本操作

创建 RasterMap 实例之后，可以对其进行放大，缩小，平移，设置地图类型等操作。

5.1 设置地图类型

在第一个地图应用中，在调用 RasterMap.setCenter 时可以指定地图类型，另外也可以使用 RasterMap.setMapType 来更改地图类型，地图开发包中定义了 Google 地图，Google 中国地图，Bing 地图，Bing 卫星图等 20 多种地图类型，也就可设置自己定义地图类型。

下面的例子顺序显示 Google 中国地图，MapAbc 地图，Bing 中国地图。

```
public class MapTypeMIDP extends MapDemoMIDP implements CommandListener {

    private int mapType = 0;
    private static final int[] mapTypes = {MapType.GOOGLECHINA,
        MapType.MAPABCCHINA, MapType.MICROSOFTCHINA};
    private Command mapTypeCommand = new Command("MapType", Command.OK, 1);

    public void startApp() {

        init();
        canvas.addCommand(mapTypeCommand);
        canvas.setCommandListener(this);
        GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
        map.setCenter(center, 13, MapType.MICROSOFTCHINA);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if (c == mapTypeCommand) {
            map.setMapType(mapTypes[mapType]);
            mapType++;
            mapType %= mapTypes.length;
        }
    }
}
```



Bing 中国地图



Google 中国地图



MapAbc地图

5.2 放大、缩小

RasterMap 的 zoomIn, zoomOut 用来放大缩小地图。

```
public class MapZoomMIDP extends MapDemoMIDP implements
CommandListener{

    private Command mapZoomInCommand=new Command("Zoom
In",Command.OK,1);
    private Command mapZoomOutCommand=new Command("Zoom
Out",Command.CANCEL,1);

    public void startApp() {
        init();
        canvas.addCommand(mapZoomInCommand);
        canvas.addCommand(mapZoomOutCommand);
        canvas.setCommandListener(this);
        GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
        map.setCenter(center, 13, MapType.MICROSOFTCHINA);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if(c==mapZoomInCommand){
            map.zoomIn();

        }else if(c==mapZoomOutCommand){
            map.zoomOut();
        }
    }
}
```



5.3 地图平移

RasterMap 有两个方法可以用于平移地图，panTo 将地图移动到指定经纬度坐标，panDirection(dx,dy) 将地图从当前位置平移 dx,dy 个像素。

下列示例可以上，下，左，右平移地图。

```
public class MapPanMIDP extends MapDemoMIDP implements CommandListener {

    private Command mapUpCommand = new Command("Up", Command.OK, 1);
    private Command mapDownCommand = new Command("Down", Command.ITEM, 1);
    private Command mapLeftCommand = new Command("Left", Command.ITEM, 1);
    private Command mapRightCommand = new Command("Right", Command.ITEM, 1);

    public void startApp() {

        init();
        canvas.addCommand(mapUpCommand);
        canvas.addCommand(mapDownCommand);
        canvas.addCommand(mapLeftCommand);
        canvas.addCommand(mapRightCommand);
        canvas.setCommandListener(this);
        GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
        map.setCenter(center, 13, MapType.MICROSOFTCHINA);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if (c == mapUpCommand) {
            map.panDirection(0, -32);

        } else if (c == mapDownCommand) {
            map.panDirection(0, 32);
        } else if (c == mapLeftCommand) {
            map.panDirection(-32, 0);
        } else if (c == mapRightCommand) {
            map.panDirection(32, 0);
        }
    }
}
```

5.4 地图缓存

RasterMap 内部具有缓存功能，如果需要下载的图片已在缓存中，则直接从缓存中读取从而提高地图显示响应速度。缓存大小和是否使用缓存可以使用配置修改。要注意的是缓存只是临时的，程序退出后，缓存被清空，如果需要在程序退出时保存缓存中的内容以便下次启动应用时可以恢复上次的结果，可以使用 RasterMap 的 saveMapCache 和 restoreMapCache。

6.0 使用地图服务

引路蜂地图 API 中同样提供了地址查询，路径查询，本地搜索，IP 地址查询，地址反编码（通过经纬度查地名）等。

6.1 地址查询

地址查询（或称为地址编码）是将输入的地名（如南京林业大学）转换成对应的经纬度坐标然后将其显示在地图上。

```
public class MapGeocodingMIDP extends MapDemoMIDP implements CommandListener,
    IGeocodingListener {

    private Command mapFindAddressCommand = new Command("Find Address", Command.OK, 1);

    public void startApp() {

        init();
        canvas.addCommand(mapFindAddressCommand);
        map.setGeocodingListener(this);
        canvas.setCommandListener(this);
        GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
        map.setCenter(center, 15, MapType.MICROSOFTCHINA);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if (c == mapFindAddressCommand) {
            String name = "南京林业大学";
            map.getLocations(name);
        }
    }

    public void done(String query, MapPoint[] result) {
        if (result != null) {
            map.panTo(result[0].getPoint());
        }
    }
}
```

所有的地图服务都是采用异步方式调用，在调用 `RasterMap.getLocation(address)` 前，需要设置好返回结果时的回调函数 `RasterMap.setGeocodingListener`，回调函数接口定义为 `IGeocodingListener`。

回调方法为 `public void done(String query, MapPoint[] result)`，如果查询结果不为空，则 `result` 为查询结果的数组。示例中将地图转到第一个查询结果。



对于 MapAbc 地图服务，还可以指定城市编码，如南京编码为 2 5。

```
public void getLocation(int citycode,String query, IGeocodingListener listener);
```

6.2 路径查询

可以通过 RasterMap 的 `getDirection()`方法来查询路径，和查询地址类似，路径查询的结果也是通过回调函数的方式来通知应用程序的，下面的例子返回南京到北京的路径。

返回结果存放在 `MapDirection` 中，`MapDirection` 包含了路径的详细信息，包括路径的每个步骤，长度，时间，方向等。

```
public class MapRoutingMIDP extends MapDemoMIDP implements CommandListener,
    IRoutingListener {
    private Command mapGetDirectionCommand = new Command("Get Direction", Command.OK, 1);
    public void startApp() {
        init();
        canvas.addCommand(mapGetDirectionCommand);
        map.setRoutingListener(this);
        canvas.setCommandListener(this);
        GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
        map.setCenter(center, 15, MapType.MICROSOFTCHINA);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if (c == mapGetDirectionCommand) {
            String name1 = "南京";
            String name2 = "北京";
            map.getDirections("from: " + name1 + " to: " + name2);
        }
    }

    public void done(String query, MapDirection result) {
        if (result != null) {
```



```

        map.setMapDirection(result);
        map.resize(result.getBound());
        map.zoomOut();
    }
}

```



地图服务可以选择使用 Google 地图服务，CloudMade 地图服务，在中国还可能选择 MapAbc 地图服务，缺省使用 Google 地图服务。

getDirections()具有三个重载函数，例子中是采用的文字描述方式。上述示例采用了 from: address1 to: address2 的格式，CloudMade 地图服务和 MapAbc 地图服务则必需采用 经度 1，纬度 1，经度 2，纬度 2 和格式。

为避免混淆，可以使用下述格式。

```
public void getDirection(GeoLatLng[] waypoints, IRoutingListener listener);
```

其中 waypoints 为途径点坐标数组经纬值，可以支持多点路径查询。

此外对于 MapAbc 地图服务，还可以指定城市编码，如南京编码为 2 5。

```
public void getDirection(int citycode,String query, IRoutingListener listener);
```

6.3 本地查询

本地查询可以查询指定区域内诸如宾馆，邮局等用户感兴趣的地方。其使用方法和地址查询非常类似。

下面示例查询 32.0616667, 118.7777778 为中心点一定区域内宾馆的信息。

```

public class MapLocalSearchMIDP extends MapDemoMIDP implements CommandListener,
    IGeocodingListener {

    private Command mapFindAddressCommand = new Command("Find Address", Command.OK, 1);

    public void startApp() {

```

```

init();
canvas.addCommand(mapFindAddressCommand);
map.setGeocodingListener(this);
canvas.setCommandListener(this);
GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
map.setCenter(center, 15, MapType.MICROSOFTCHINA);
DigitalMapService.getSearchOptions().LanguageID = "zh-cn";
Display.getDisplay(this).setCurrent(canvas);
}

public void commandAction(Command c, Displayable d) {
    if (c == mapFindAddressCommand) {
        String name = "宾馆";
        GeoLatLng screenCenter = map.getScreenCenter();
        map.getLocations(name, 0, screenCenter, map.getScreenBounds(screenCenter));
    }
}

public void done(String query, MapPoint[] result) {
    if (result != null) {
        map.panTo(result[0].getPoint());
        for (int i = 0; i < result.length; i++) {
            System.out.println(result[i].objectNote);
        }
    }
}
}

```

本地查询方法

```
public void getLocations(String address,int start,GeoLatLng center,GeoBounds bound,
IGeocodingListener listener);
```

指定中心点和查询区域。本地查询可以多次返回结果，start 为查询结果起始顺序，每次返回结果在 SearchOptions 中定义，缺省每次返回 4 个。

上述结果返回内容为，具体结果可能不同。

"晶丽酒店 025-83310818 025-83310818212223 025-86636636 "

"城市名人酒店 025-83123333 025-83123888 "

"南京玄武饭店 025-83358888 "

"Sofitel Galaxy Nanjing - 南京索菲特银河大酒店 025-83718888 "

6.4 IP 地址查询

IP 地址查询，可以根据 IP 地址查询到该 IP 所在的地理经纬度坐标，比如下述例子查询 IP 地址 58.192.32.1，所在经纬度为 118.777802,32.061699，为南京大学所在地。

```

public class MapIpSearchMIDP extends MapDemoMIDP implements CommandListener,
    IIpAddressGeocodingListener {

    private Command mapFindAddressCommand = new Command("Find Address", Command.OK, 1);

```



```
public void startApp() {

    init();
    canvas.addCommand(mapFindAddressCommand);
    map.setIpAddressGeocodingListener(this);
    canvas.setCommandListener(this);
    GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
    map.setCenter(center, 15, MapType.MICROSOFTCHINA);
    Display.getDisplay(this).setCurrent(canvas);
}

public void commandAction(Command c, Displayable d) {
    if (c == mapFindAddressCommand) {
        map.getIpLocations("58.192.32.1");
    }
}

public void done(String query, IpAddressLocation result) {
    if (result != null && result.error.length() == 0 && result.longitude.length() > 0
        && result.latitude.length() > 0) {
        try {
            MapPoint mapPoint = new MapPoint();
            String latLng = "[" + result.longitude + "," + result.latitude + ",0]";
            mapPoint.point = DigitalMap.fromStringToLatLng(latLng);
            mapPoint.setName(result.organization);
            mapPoint.setNote(result.city + " " + result.country);
            map.panTo(mapPoint.point);
        } catch (Exception e) {
            result.error = "IP_NOT_FOUND";
        }
    }
}
}
```

注：目前 IP 查询结果总是以英文返回，如上述结果详细内容。

ISP: "China Education and Research Network"

Organization: "Nan Jing University"

Country: "CN"

City: "Nanjing"

若想知道该经纬度对应的地名，可以使用地址反编码服务。

你可以输入 127.0.0.1 查询本机地址。

6.5 地址反编码

地址反编码是通过经纬度查询对应的地名，下面示例是查询经纬度为 118.777802, 32.061699 对应的地名，结果为"中国江苏省南京市鼓楼区渊声巷 41 号"。

```
public class MapReverseGeocodingMIDP extends MapDemoMIDP implements CommandListener,
    IReverseGeocodingListener {
```

```
private Command mapFindAddressCommand = new Command("Find Address", Command.OK, 1);

public void startApp() {

    init();
    canvas.addCommand(mapFindAddressCommand);
    map.setReverseGeocodingListener(this);
    canvas.setCommandListener(this);
    GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
    map.setCenter(center, 15, MapType.MICROSOFTCHINA);
    Display.getDisplay(this).setCurrent(canvas);
}

public void commandAction(Command c, Displayable d) {
    if (c == mapFindAddressCommand) {
        map.getReverseLocations("32.061699,118.777802");
    }
}

public void done(String arg0, MapPoint[] result) {
    if (result != null) {
        map.panTo(result[0].getPoint());
    }
}
}
```

注意使用字符串经纬度格式时，纬度在前，经度在后，如果反了，则返回的地名或能为空或都跑到外国去了。结果也是一个数组，一般到第一个结果，后面结果是更大的区域或是距离相对较远的地名。

6.6 选择地图服务

引路蜂地图服务缺省使用 Google 地图服务，但你也可以选择其它地图服务，比方说当 Google 服务离线时，您可以选择 MapAbc 的地图服务，另外要注意的是中国地图是有偏移的。如果想使用无偏移的中国地图，一是采用地图偏移校正算法，另外一个使用 CloudMade 地图服务。

所前所述，引路蜂地图开发包在设计时将地图图片显示和地图服务两部分设计成相对独立的两部分，Google 中国地图图片，Bing 中国地图图片，MapAbc 中国地图图片是有偏移的地图图片，CloudMade（OpenStreet）中国地图图片是无偏移的。而 Google 中国地图服务，MapAbc 中国地图服务是有偏移，CloudMade 地图服务是无偏移的。所以在选择地图类型和地图服务类型时，要么都选择有偏移，要么都选择无偏移。否则地图在显示路径时或地址时就不匹配。

下面列表是合法的组合

地图类型 (MapType)	地图服务类型 (DigitalMapService)
GOOGLECHINA	GOOGLE_MAP_SERVICE
MICROSOFTCHINA	GOOGLE_MAP_SERVICE
MAPABCCHINA	GOOGLE_MAP_SERVICE
OPENSTREETMAP	CLOUDMADE_MAP_SERVICE
GOOGLECHINA	MAPABC_MAP_SERVICE
MICROSOFTCHINA	MAPABC_MAP_SERVICE
MAPABCCHINA	MAPABC_MAP_SERVICE

下述示例地图类型使用 MICROSOFTCHINA，而使用不同的地图服务时路径查询的情况。（南京到天津的路径）

```
public class MapRoutingMIDP extends MapDemoMIDP implements CommandListener,
    IRoutingListener {
    private Command mapGetDirectionCommand = new Command("Get Direction", Command.OK, 1);
    public void startApp() {
        init();
        map.setCurrentMapService(DigitalMapService.MAPABC_MAP_SERVICE);
        //map.setCurrentMapService(DigitalMapService.GOOGLE_MAP_SERVICE);
        //map.setCurrentMapService(DigitalMapService.CLOUDMADE_MAP_SERVICE);
        canvas.addCommand(mapGetDirectionCommand);
        map.setRoutingListener(this);
        canvas.setCommandListener(this);
        GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
        map.setCenter(center, 15, MapType.MICROSOFTCHINA);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if (c == mapGetDirectionCommand) {
            GeoLatLng latLng1=new GeoLatLng(32.0418381,118.7788905);
            GeoLatLng latLng2=new GeoLatLng(39.11643,117.180908);
            map.getDirections(new GeoLatLng[] {latLng1,latLng2});
        }
    }

    public void done(String query, MapDirection result) {
        if (result != null) {
            map.setMapDirection(result);
            map.setZoom(13);
            map.panTo(new GeoLatLng(32.0418381,118.7788905));
        }
    }
}
```



Google地图服务

MapABC地图服务

CloudMade地图服务

上图分别为三种地图服务返回的路径，Google 地图服务和 MapABC 地图服务返回的结果基本相同，实际上 Google 地图服务在中国的地图是来自于 MapABC。而 CloudMade 地图返回的结果似乎偏移了道路。这是因为 Bing 中国地图是被偏移过的，而 CloudMade 返回的无偏移的结果。Google 地图服务和 MapABC 地图服务返回的结果也是偏移过的，所以和偏移过的地图很“匹配”。

6.7 叠加自定义图层

在开发应用的过程中，常常需要在地图上需绘制一点自定义的兴趣点或是自定义的几何图形。引路蜂地图包中 RasterMap 最终提供的基实就是一张图片。

```
protected void paint(Graphics g) {  
    map.paint(mapGraphics);  
    g.drawImage((Image) mapImage.getNativeImage(), 0, 0, 0);  
    //start drawing your own sharps or images.  
    ...  
}
```

所以一个简单的方法是绘制完地图后，你可以使用任何绘图方法在地图绘制任何自定义的图形或是图象。

这里需要注意的是坐标变换，RasterMap 采用的是经纬度坐标，而屏幕显示采用的屏幕坐标，RasterMap 提供了坐标转换的方法：fromScreenPixelToLatLng 坐屏幕坐标

转换成地图经纬度坐标。`fromLatLngToScreenPixel` 从经纬度坐标转换成屏幕坐标。

下面给出的例子是采用派生 `MapLayer` 子类的方法，`RasterMap` 是 `MapLayerContainer` 的子类，可以用来管理多个地图层。这些图层从下到上相当于透明纸一层一层叠加形成最终显示的地图。

例子中显示几个自定义兴趣点，和一个三角形，并在地图中心显示十字标。

```
public class MapOverlayMIDP extends MapDemoMIDP {

    OverLayMapLayer mapLayer;

    public void startApp() {
        init();
        GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
        map.setCenter(center, 9, MapType.GOOGLECHINA);
        Display.getDisplay(this).setCurrent(canvas);
        mapLayer = new OverLayMapLayer(canvas.getWidth(),
            canvas.getHeight());
        map.addMapLayer(mapLayer);
    }

    class OverLayMapLayer extends MapLayer {

        GeoLatLng pt1 = new GeoLatLng(32.345281, 118.84261);
        GeoLatLng pt2 = new GeoLatLng(32.05899, 118.62789);
        GeoLatLng pt3 = new GeoLatLng(32.011811, 118.798656);

        public OverLayMapLayer(int width, int height) {
            super(width, height);
        }

        public void paint(IGraphics graphics, int offsetX, int offsetY) {
            drawCursor(graphics);
            drawTriangle(graphics);
            drawPoint(graphics, pt1);
            drawPoint(graphics, pt2);
            drawPoint(graphics, pt3);
        }

        public void drawTriangle(IGraphics g) {
            GeoPoint ptOnScreen1 = map.fromLatLngToScreenPixel(pt1);
            GeoPoint ptOnScreen2 = map.fromLatLngToScreenPixel(pt2);
            GeoPoint ptOnScreen3 = map.fromLatLngToScreenPixel(pt3);
            g.setColor(0x0000FF);

            g.drawLine((int) ptOnScreen1.x, (int) ptOnScreen1.y,
                (int) ptOnScreen2.x, (int) ptOnScreen2.y);
            g.drawLine((int) ptOnScreen2.x, (int) ptOnScreen2.y,
                (int) ptOnScreen3.x, (int) ptOnScreen3.y);
            g.drawLine((int) ptOnScreen1.x, (int) ptOnScreen1.y,
                (int) ptOnScreen3.x, (int) ptOnScreen3.y);
        }

        public void drawPoint(IGraphics g, GeoLatLng pt) {
            GeoPoint ptOnScreen = map.fromLatLngToScreenPixel(pt);
            int x = (int) ptOnScreen.x;
            int y = (int) ptOnScreen.y;
        }
    }
}
```

```
        g.setColor(0x00FF00);
        g.fillRect(x - 4, y - 4, 8, 8);

    }

    private void drawCursor(IGraphics g) {
        int x = getScreenWidth() / 2;
        int y = getScreenHeight() / 2;
        g.setColor(0x205020);
        g.drawRect(x - 4, y - 4, 8, 8);
        g.drawLine(x, y - 6, x, y - 2);
        g.drawLine(x, y + 6, x, y + 2);
        g.drawLine(x - 6, y, x - 2, y);
        g.drawLine(x + 6, y, x + 2, y);
    }
}
```



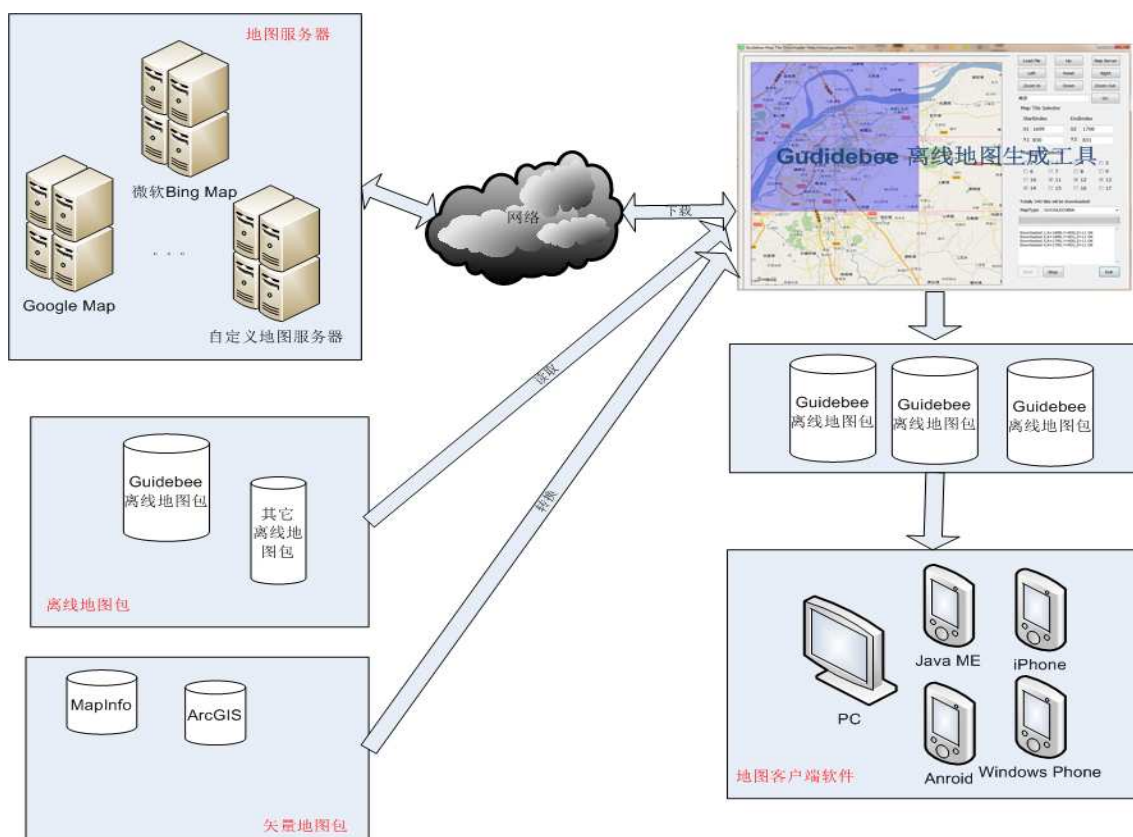
7.0 离线地图

7.1 离线地图生成工具结构图

前面描述的地图应用基本上都需要网络连接，每次更新地图都需要连接地图服务器。在编写地图客户端软件时，常常会想使用离线地图包，这样一来可以节省流量，同时也可以提高客户端响应能力。引路蜂软件在提供地图开发包的同时也提供了离线地图生成工具，可以用于地图开发包，可以做到显示地图时，先从本地地图包取出相应图片，如果本地没有，再从相应地图服务器上取图片。

Guidebee 离线地图生成器的数据来源可以有三种。

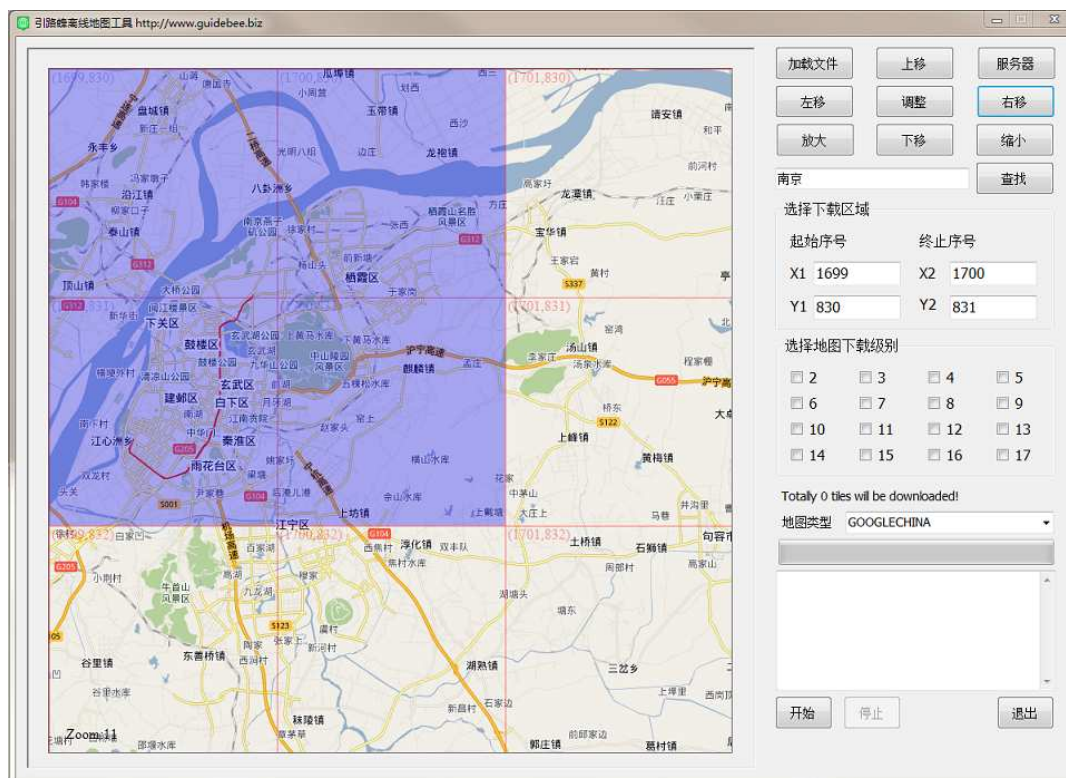
1. 各种地图服务器如微软 Bing map, google map 或者是用户自定义的服务器。只要符合通用的分片原则 <http://msdn.microsoft.com/en-us/library/bb259689.aspx>
2. 已生成的离线地图包，包括 guidebee 离线地图包以及其它格式离线地图包（如 mgmap）。
3. 来自矢量地图数据（如 MapInfo, ArgGIS 地图数据）



对于不同来源的数据，Guidebee 离线地图生成工具生成最终带索引的离线地图包.map，可以通过 Mapdigit 地图开发包用于各种平台的客户端。

7.2 离线地图生成工具使用方法

引路蜂离线地图生成工具用于生成离线地图包，运行环境需要安装 .net framework 3.5。



移动地图



加载文件 --> 读取离线文件，可以通过离线工具查看已生成的离线地图包，或者读取 MapInfo 矢量地图(不在免费版本中)

上移，左移，右移，下移-->移动地图，也可以通过鼠标来拖动地图

放大，缩小-->放大及缩小地图

服务器 -->在读取离线文件后，当时显示为离线地图包内容，若想重新连接相应地图服务器，则按些钮。

调整-->将当前地图对齐到最近的网格中。

查找-->为方便移动到地图到指定地方，可以通过查询地图来移动。

选择下载区域



选择下载区域	
起始序号	终止序号
X1 1699	X2 1700
Y1 830	Y2 831

可以通过输入起始及终止网格的序号来指定下载区域，一个简单的方法是通过按住右键拖动鼠标来选择区域，选择的区域以蓝色显示。

每个网格左上角显示当前网络的序号。

选择地图下载级别



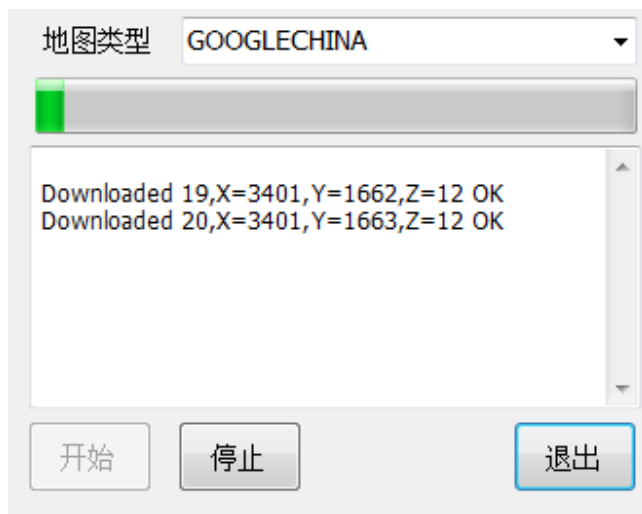
选择地图下载级别			
<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5
<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> 10	<input type="checkbox"/> 11	<input type="checkbox"/> 12	<input type="checkbox"/> 13
<input type="checkbox"/> 14	<input type="checkbox"/> 15	<input type="checkbox"/> 16	<input type="checkbox"/> 17

Totally 0 tiles will be downloaded!

在生成离线地图时，同样需要指定需下载的地图级别，比如当前放大级别为 10（显示在地图左下角）。你不能选择小于 10 的级别，下载总是会从当前级别开始。这时可以选择 11，12，一直到 17 级。total xxx，这里 xxx 给出了将有多少片图片会被下载到离线地图包中。

级别越大，下载图片越多，成指数级别增长。

地图类型选择



最后选择地图类型，地图类型只对服务器有效，可以从下拉框中选择各种地图类型，点击开始，则开始下载图片，下载结束时，显示“finished writing”。

地图文件名为 StartX_StartY_EndX_EndY_ZoomLevel_MapType.map

7.3 离线地图示例

在手机上，离线地图包一般放在 SD 卡上，然后通过文件读取。这里为简单起见，将离线放在例子的资源目录中和应用编译在一起。实际应用中请单独存放这些地图包。引路蜂地图开发包支持同时读取多个地图包，这是通过 `MapTiledZone` 类和 `MapTileStreamReader` 类来完成的。每个地图包对应于一个 `MapTiledZone`，地图包对每张地图建立的索引以加速图片检索将记录地图包存放的区域和缩放级别。而类 `MapTileStreamReader` 提供了管理这些地图包的方法。

```
public class StoredMapMIDP extends MapDemoMIDP {  
  
    MapTileStreamReader localMapTileFileReader;  
  
    public void startApp() {  
  
        init();  
  
        InputStream is = this.getClass().getResourceAsStream("/world03.map");  
        byte[] buffer = null;  
        try {  
            buffer = new byte[is.available()];  
            is.read(buffer);  
            is.close();  
        } catch (IOException ex) {  
            ex.printStackTrace();  
        }  
    }  
}
```

```
ByteArrayInputStream bais = new ByteArrayInputStream(buffer);
MapTiledZone mapTiledZone = new MapTiledZone(new DataInputStream(bais));

localMapTileFileReader = mapTileDownloadManager.getInteralMapTileStreamReader();
localMapTileFileReader.addZone(mapTiledZone);
//you can add more store map package here use addZone

localMapTileFileReader.open();
GeoLatLng center = new GeoLatLng(32.0616667, 118.7777778);
map.setCenter(center, 3, MapType.MICROSOFTCHINA);
Display.getDisplay(this).setCurrent(canvas);
}

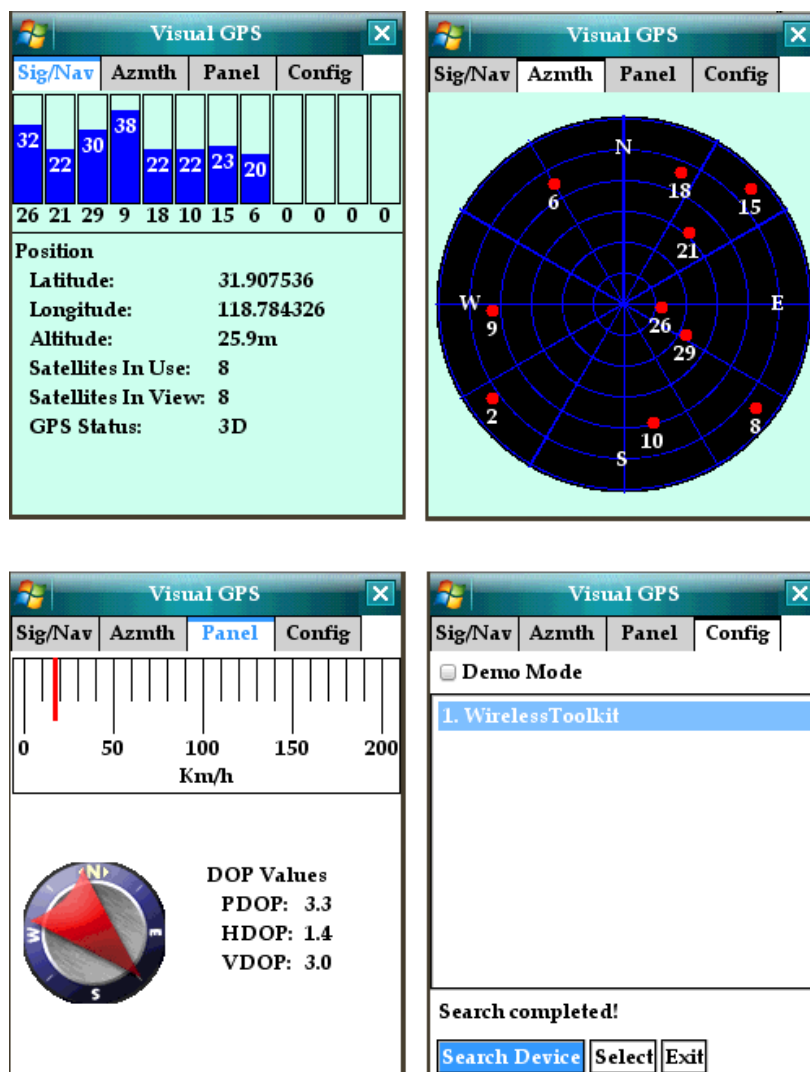
public void destroyApp(boolean unconditional) {
    localMapTileFileReader.close();
}
}
```



World03.map 存放了 1 - 4 级世界地图。MapTileManager 内部定义一个 MapTileStreamReader，如果设置了本地地图，MapTileManager 会先从本地读取，如果本地没有需要图片，然后再从地图服务器上下载。可以使用 MapTileStreamReader 的 addZone 方法添加多个地图包，地图包可以有重叠的区域，这时添加的先后顺序对结果就会影响，如果有多个，则取第一个。

8.0 GPS 定位

引路蜂地图开包中也提供了对 G P S 定位信息处理的支持，由于各种移动设备上对 GPS 支持的不同，比如在 Java ME 上有 J S R 1 7 9，iphone 上有 Core Location。您可以使用平台自带的定位 A P I 配合引路蜂地图开包使用。也可以使用引路蜂地图开包中定位 A P I。引路蜂地图开包定位 A P I 提供了对 N M E A 2.0 数据的处理，可以获取当前经纬度，卫星位置，方向，速度等。



这是使用引路蜂定位 A P I 的一个 G P S 应用，可以从网站下载源码，这个应用显示了当前 G P S 设备的各种信息。

9.0 矢量地图和导航 API

矢量地图和导航 API 属于地图高级应用，将在专门的文档中描述。其中矢量地图主要包含在开发包 `Gis.Vector`，而导航 API 则在 `Gis.Navigation` 包中。有兴趣的可以参见网站 www.guidebee.biz 论坛，或是引路蜂软件开发

Blog <http://blog.csdn.net/guidebee>