GUIDEBEE

Digital Map API Developer Guide

James Shen

www.guidebee.biz

Guidebee Pty Ltd.

ISSUE/AMENDMENT STATUS

Issue	Date	Description	Author
1.0	24th Jan 2009	First Version	James Shen james.shen@guidebee.biz
2.0	19th Sep 2009	Added Navigation API	James Shen james.shen@guidebee.biz
2.1	8th Jan 2011	Update for all platforms	James Shen james.shen@guidebee.biz

TABLE OF CONTENTS

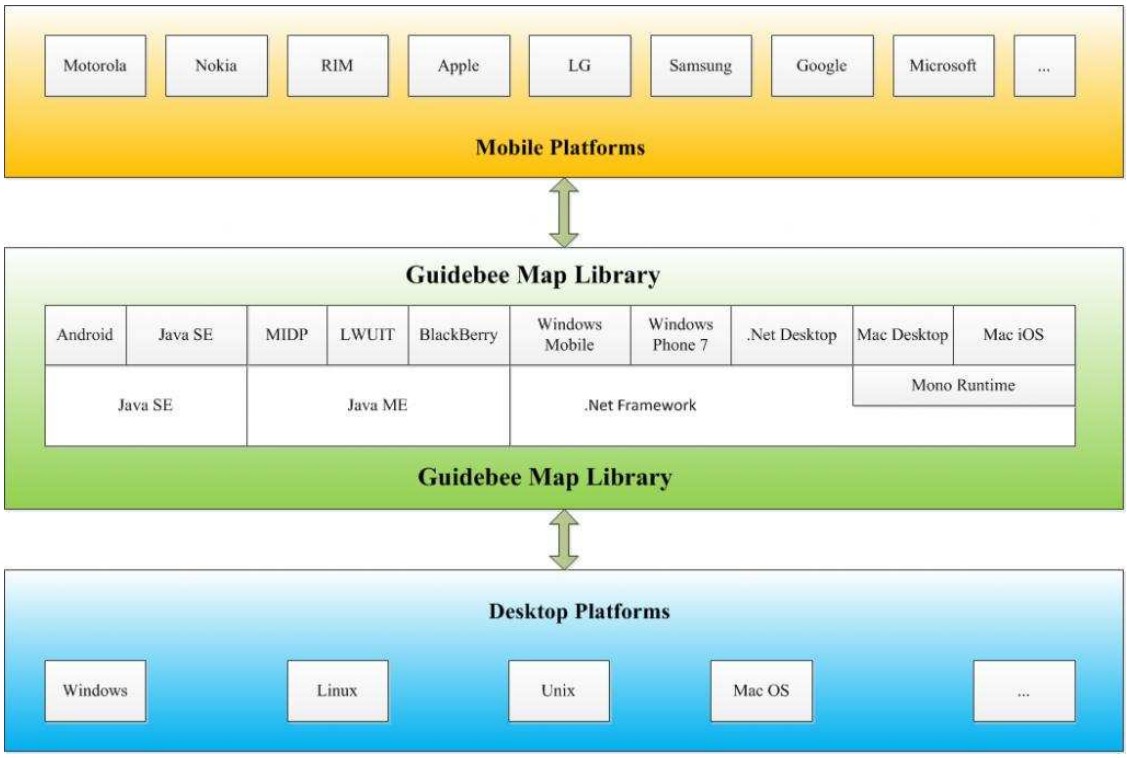
1.0 Overview of Guidebee Digital Map Java ME API

Speaking of Map API, the Google map API and Bing Map API probably are the most popular ones available to developers, but they mainly work with web applications requires online network and for example it's not very convenient to write applications on Java ME platform with Google Map and it'd be difficult to work with Bing Map on Android or iPhone devices.

Guidebee Map API supports all most every platform on the market, either desktop or mobile platforms and with consistent APIs; you can use more than 20 different map types includes Google Map, Bing Map, and Yahoo Map ,offline map and vector map with the same API package.

The following diagram shows the platforms Guidebee Map API supports. The API is provided as three different binaries to target all platforms.

- Java ME library MIDP/LWUIT/Blackberry
- Java SE library Android/Java SE
- .Net framework library Windows Mobile/Desktop/Windows Phone 7
  Mac OS/iPhone/iPad (with Mono, MonoTouch)



Major features defined in Guidebee Map API are shown in following table:

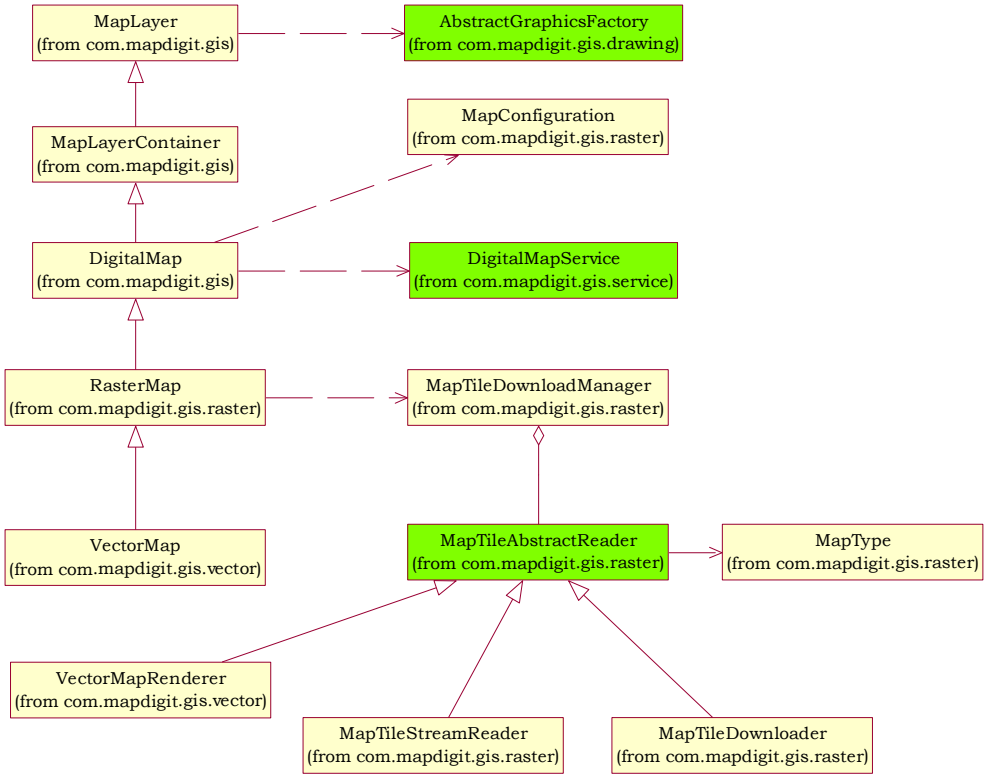
Guidebee Map API Features	Java ME (J2ME) & Blackberry (RIM)			Android& Java SE		.Net Framework	
Native UI Framework support	**MIDP√**	**LWUIT√**	**Blackberry UI√**	**Android √**	**Java SE √**	**Mobile √**	**Desktop√**
Map Pan	√			√		√	
Map Zoom In/Zoom Out	√			√		√	
Geocoding	√			√		√	
Reverse Geocoding	√			√		√	
Routing	√			√		√	
Local search	√			√		√	
IP Search	√			√		√	
Latitude, Longitude search	√			√		√	
Location API	√			√		√	
Cell Location search	√			√		√	
Google Map Service	√			√		√	
MapAbc Map Service	√			√		√	
CloudMade Map Service	√			√		√	
Vector Map support	√			√		√	
Stored Map support	√			√		√	
Multiple Map types support	√			√		√	
Multiple Language support	√			√		√	
Customized Map Type support	√			√		√	
Other UI Framework	√			√		√	
Graphics 2D API	√			√		√	
Sample code	√			√		√	
Detailed document	√			√		√	
Trial license (Free)	√			√		√	
Commercial License	√			√		√	

Note: support for Mac OS, iPhone, iPad requires Mono, MonoTouch.

2.0 Guidebee Map API packages

2.1 Core classes

Below is the core classes defined in the Map API.



The classes in green in above diagram define base classes to support different UI frameworks on different platforms, different map type support and to support different map services.

The Guidebee Map API design made two abstractions for the map type (the map images) and map services (Geocoding, routing etc).

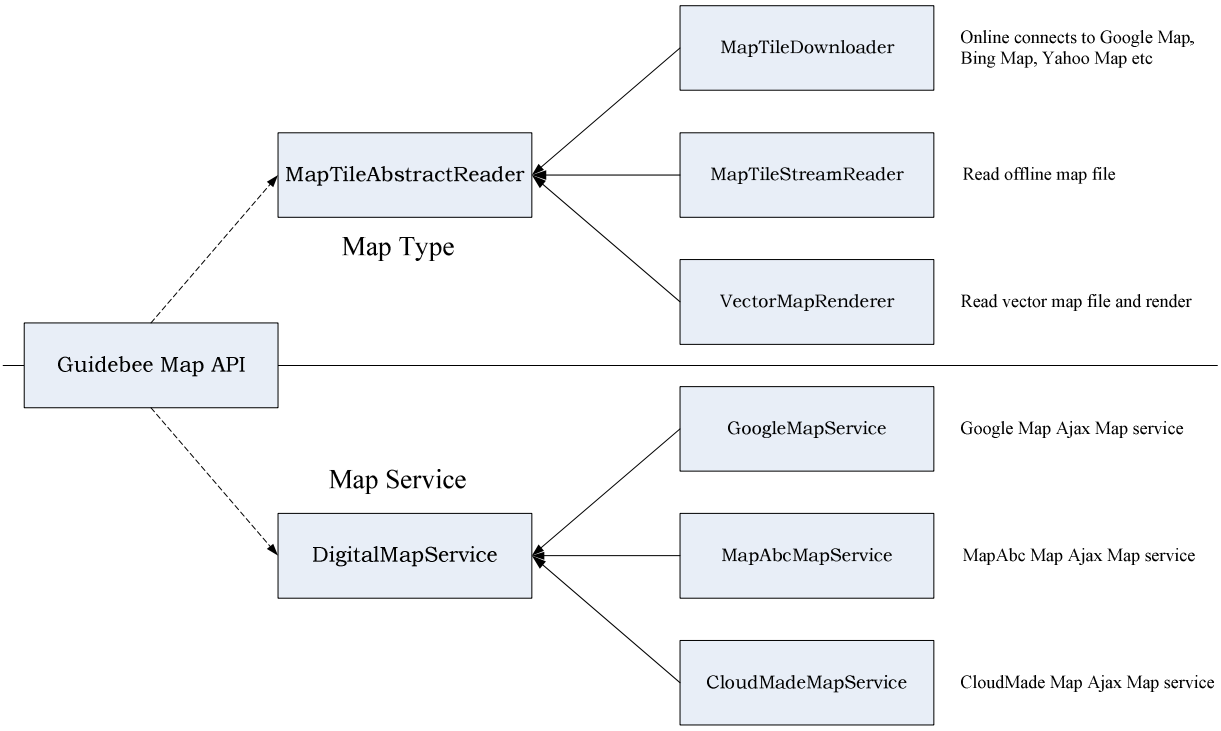
The map type is the map images; it comes from 3 different sources:

- Online map, it connects to map servers (like Yahoo map, Microsoft Live Map etc) and display map image.
- Offline map (or stored map), allow access self defined map image file locally to avoid data traffic to server.
- Vector map, it supports MapInfo compatible map data and provide similar functionalities as MapInfo map product.

While the Map services refer to Geocoding, Reverse Geocoding, Routing, Local Search etc, Guidebee Map API internally support Google Map service, CloudMade Map Service, if in

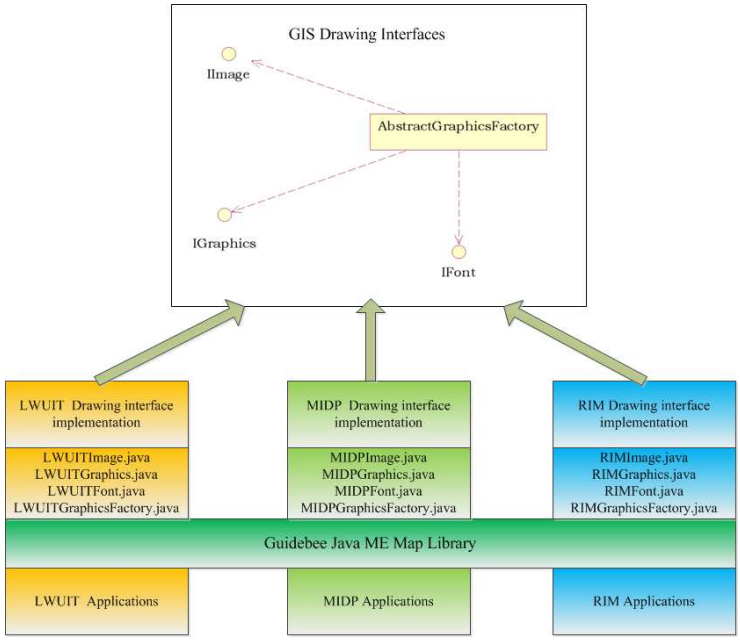
China also support MapAbc Map services. The other map services like Bing Map or Yahoo Map may be included in future releases.

Since Map Type (map images) and Map Services (Routing, Geocoding etc) are relatively independent with each other, you can display Google Map and using CloudMade Map services or show Bing Map China and use MapAbc map services. It provides such flexibility for developers to cater for different circumstances.



An abstract Graphics system is defined in Guidebee Map API and it's used to support different Graphics systems on different platforms. In package Gis.Drawing, IImage, IGraphics, IFont, AbstractGraphicsFactory are defined as interface or abstract class. In this way, the same binary with platform dependant implementation works on different platforms.

For instance, On Java ME platform, developer can choose MIDP's LCDUI or Sun's LWUIT or choose Blackberry UI framework for RIM devices.



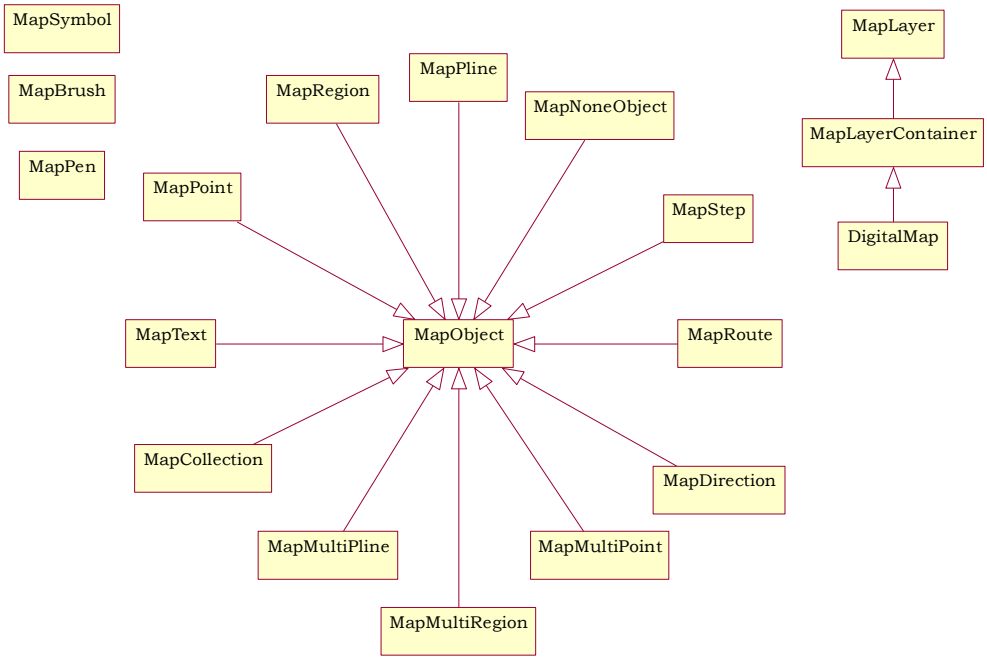
Java and C# (.Net Framework) are two major languages supported by Guidebee Map API. Though Java and C# are different, but there's nearly one to one mapping between two languages for the Map API, if you know one, you'll be quite familiar with the other.

Here's the list of the major packages defined within Guidebee Map API.

Package Name	Java Package	.Net Package
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
License	com.mapdigit.licence	Mapdigit.Licence

2.2 Package Gis

Package Gis define some common Map Objects:

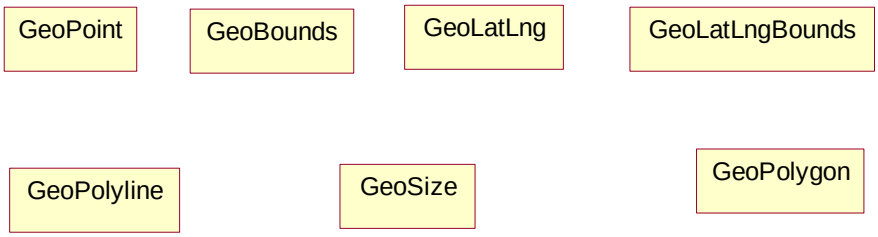


MapDirection	This class is used to store driving directions results
MapObject	Base class of all map objects.
MapPoint	Class MapPoint stands for a point map object.
MapRoute	Objects of this class store information about a single route in a directions result.
MapStep	Objects of this class store information about a single step within a route in a directions result.
DigitalMap	DigitalMap is the base class for Raster Map and Vector Map..
MapSymbol	Map symbol used to display a point.
MapPen	Map pen used to draw a map object.
MapBrush	Map brush used to paint a map object.
MapPoint	Class MapPoint stands for a point map object.
MapPline	Class MapPline stands for a map pline object.
MapRegion	Class MapRegion stands for a map region object.
MapMultiPoint	Class MapMultiPoint stands for map points' collection.
MapMultiPline	Class MapMultiPline stands for map plines' collection.
MapMultiRegion	Class MapMultiRegion stands for map regions' collection.

MapCollection	Class MapCollection stands for a collection of map objects.
MapNoneObject	Class MapNoneObject stands for a map object without geo info.
MapText	Class MapText stands for a text map object.
MapLayer	Base class for a map layer.
MapLayerContainer	A container for map layers, support add, move, delete map layers.

2.3 Package Gis.Geometry

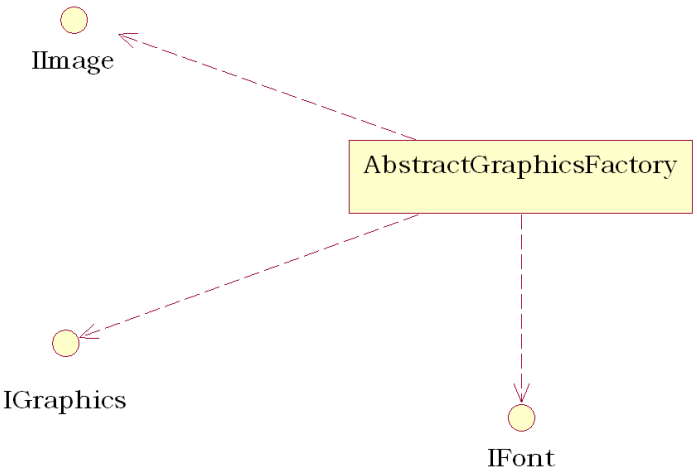
This package defines geographical geometry objects, like polygon, point, polyline etc.



GeoBounds	GeoBounds is a rectangular area of the map in pixel coordinates
GeoLatLng	GeoLatLng is a point in geographical coordinate's longitude and latitude.
GeoLatLngBounds	GeoLatLng is a bound in geographical coordinate's longitude and latitude.
GeoPoint	A point representing a location in (x, y) coordinates space, specified in integer precision.
GeoPolygon	Polygon on map.
GeoPolyline	Polyline on map.
GeoSize	The GeoSize class encapsulates the width and height of a component (in integer precision) in a single object.

2.4 Package Gis.Drawing

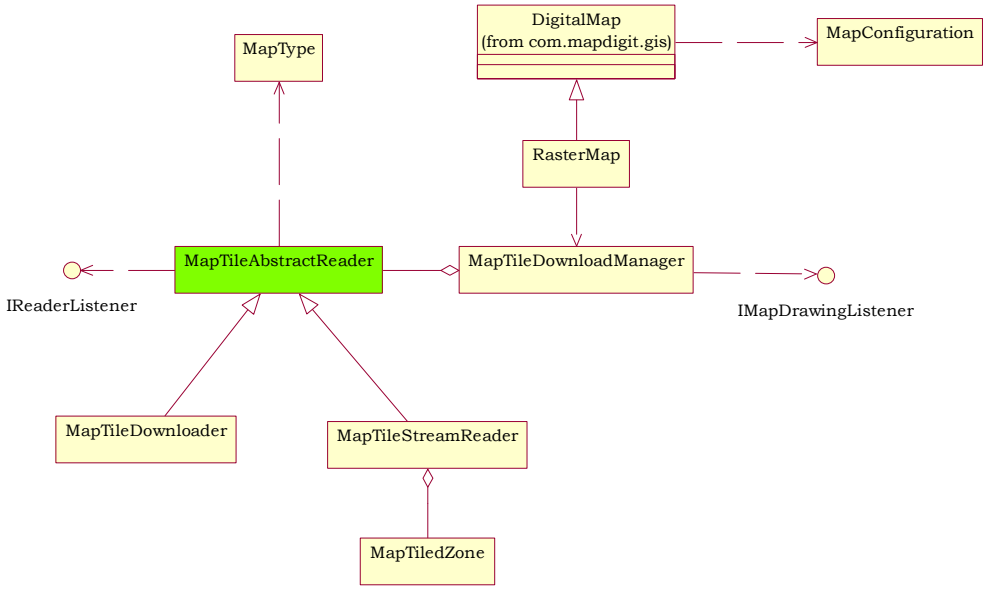
This package define a common interface for different Graphics System, such as MIDP Graphics or LWUIT Graphics classes, with these interfaces ,Guidebee Digital Map can be used with different graphics systems.



AbstractGraphcsFactory	Factor class, used to create other graphics objects like font, graphics ,images etc.
IGraphics	Interface for Graphics object used to draw font, images etc, .
IFont	Font interface.
IImage	Image interface.

2.5 Package Gis.Raster

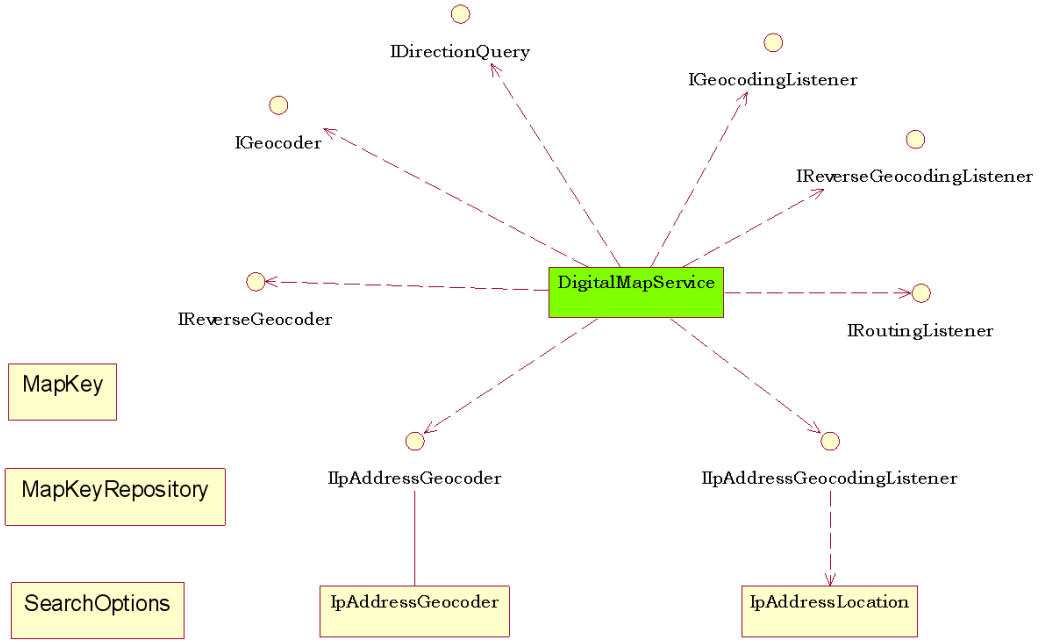
The raster package defines object used to raster maps (online or stored map).



Map Type	Define different map types (Google, Microsoft Live, Yahoo etc).
MapConfiguration	To fit for different device capability (memory size, screen size), the Map API can be configured to turn on/off cache, drawing routing polyline or not etc).these configuration will effect memory usage and performance of the Map API.
RasterMap	A core map class used to display map from server or from stored map file, provide panning, zooming etc.
MapTileAbstractReader	An abstract class defined how map tiles is obtained, connecting to map server, reading from local map files or rendering the vector map.
MapTileDownloader	MapTileDownloader download map image tiles from server (msn, yahoo, etc).
MapTileStreamReader	Read map image tiles from a input stream (mostly from local map file)
MapTileZone	A predefined stored map file. Developer can define their own stored map file format, if only it derives from MapTileAbstractReader.
IReaderListener	A listener used to monitor the progress of the reading
IMapDrawingListener	When a map reader finish downloading/reading/rendering a map tile, it trigs the listener to notify a map tile is available.

2.6 Package Gis.Service

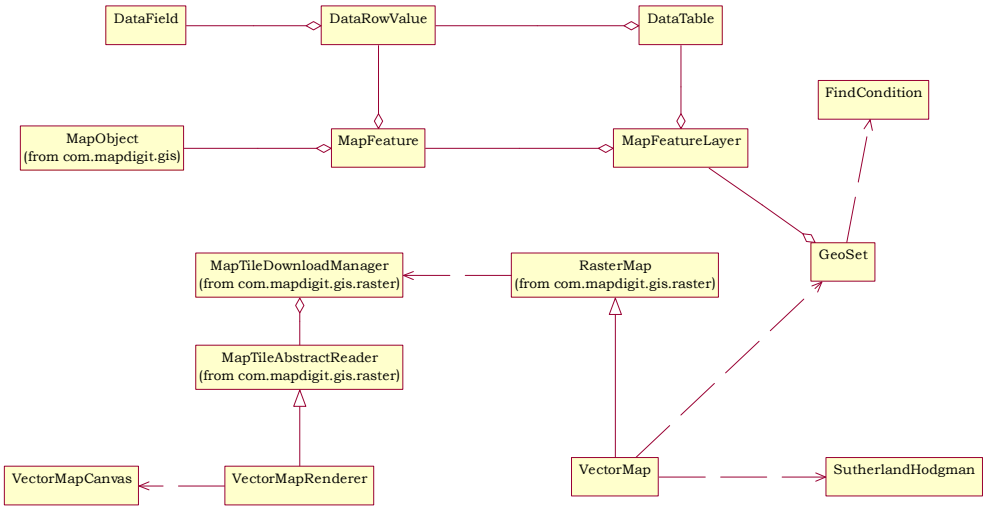
The service defines interface for Digital map services (Geocoding, reverse Geocoding and routing etc).



DigitalMapServer	Default implementation for Map Service (routing, Geocoding ,reverse Geocoding)
IDirectionQuery	Interface to query routing information..
IGeocoder	Interface to find address.
IReverseGeocoder	Interface to find address based on its latitude and longitude.
IIpaddressGeocoder	Interface to find address based on its ip address (may not included in this release).
IGeocodingListener	Call back when Geocoding is done.
IReverseGeocodingListener	Callback when reverse Geocoding is done.
IRoutingListener	Callback when routing is done.
IIpaddressGeocodingListener	Callback when IP Geocoding is done.
IpAddressGeocoder	Default implementation of IP Geocoding service.
MapKey	Define map key used by map services
MapKeyRepository	Manage map keys
SearchOption	Define options for using map services.

2.7 Package Gis.Vector

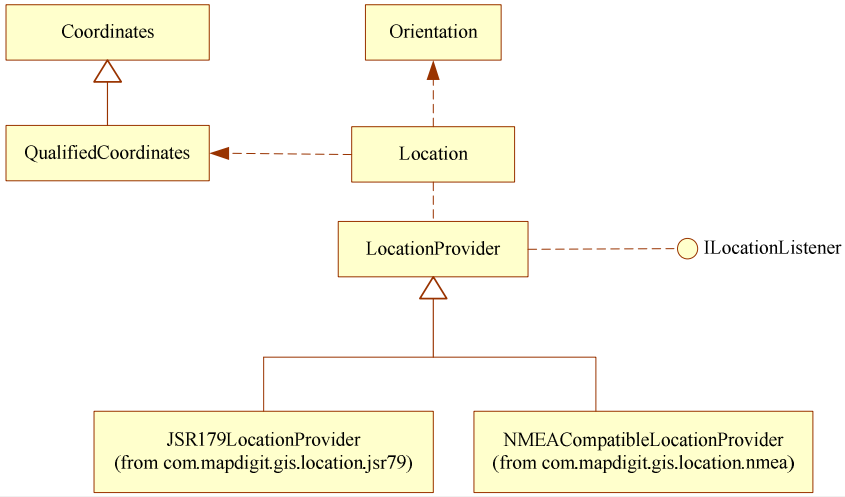
Vector package define objects for vector map.



SutherlandHodgman	Implements Sutherland-Hodgman clip algorithm.
VectorMap	VectorMap is the basic building blocks for Guidebee vector map. Each map is consists of multiple map Layers.
GeoSet	The Geoset object is built off the Map object and allows you to define a geoset. A Geoset is a collection of map layers and their settings.
MapFeatureLayer	MapLayer defines a map layer.Computer maps are organized into layers.
MapFeature	MapFeature defines a map feature in a map layer.
DataTable	Defines one tabular database table
DataField	Defines a field of a database table.
DataRowValue	Defines a row of a database table.
FindCondition	Defines a find condition when search for records.
FindConditions	Defines a find condition collection when search for records.

2.8 Package Gis.Location

The location packages defines API provide a common interface for GPS devices. The implementations provide a wrapper classes for JSR179 API, and also support NMEA compatible devices.



Coordinates	The Coordinates class represents coordinates as latitude-longitude-altitude values.
Location	The Location class represents the standard set of basic location information.
LocationProvider	This is the starting point for applications using location information in this API and represents a source of the location information.
Orientation	The Orientation class represents the physical orientation of the terminal.
QualifiedCoordnates	The QualifiedCoordinates class represents coordinates as latitude-longitude-altitude values that are associated with an accuracy value
JSR179LocationProvider	Wrapper class for JSR179 API locationProvider
NMEACompatibleLocationProvider	Base class for NMEA compatible devices.

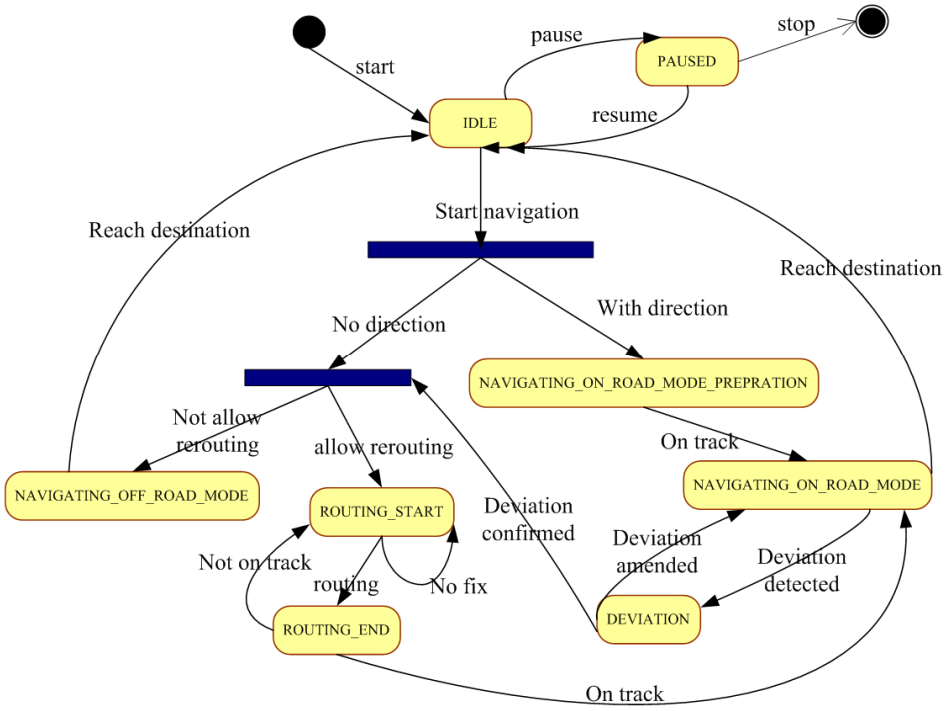
2.9 Package Gis.Navigation

The location packages defines API which provide turn by turn navigation instruction, the core class is NavigationEngine, it internally includes three working thread

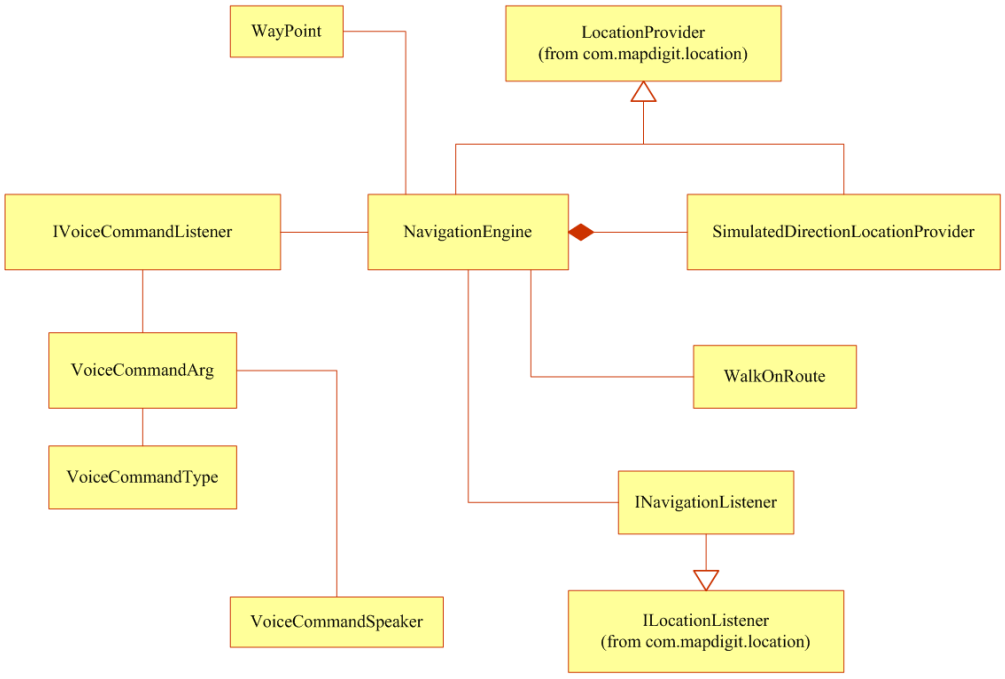
- Location monitor thread used to monitor current location against current navigation route if there's one and adjust raw location to the nearest position on the route.
- Voice command generator generate thread create upcoming voice command based on current step and location and put them to the voice command queue.

- Voice command processor pick up the voice command from the queue and notify the voice command listener

Here's state transition diagram of NavigationEngine:



Classes defined in this package:

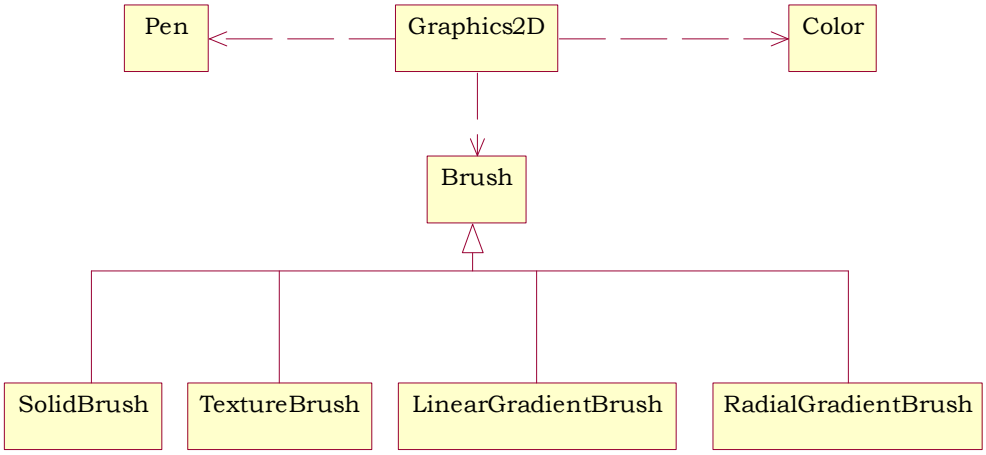


WayPoint	Define a navigation way point; it defines the name and

	Lat/Lng pair.
NavigationEngine	Navigation Engine, it internal has a navigation simulation thread ,it support turn by turn navigation , has on road and off road navigation mode, rerouting support etc.
SimulatedDirectionLocationProvider	Simulation location provider, it uses a MapDirection object as the simulation source.
VoiceCommandArg	Voice command argument class, which defines command type, command argument (which usually is the current road name)
VoiceCommandSpeaker	Default implementation for audio command player.
VoiceCommandType	Define all navigation command type, such as distance to a turn, turn left ,turn right etc.
WalkOnRoute	Current position on the route (nearest).
INavigationListener	Navigation listener, which monitor the status of the navigation engine, location update, rerouting status etc.
IVoiceCommandListener	Voice command listener.

2.10 Package drawing

The Drawing package provides access to basic 2D graphics functionality

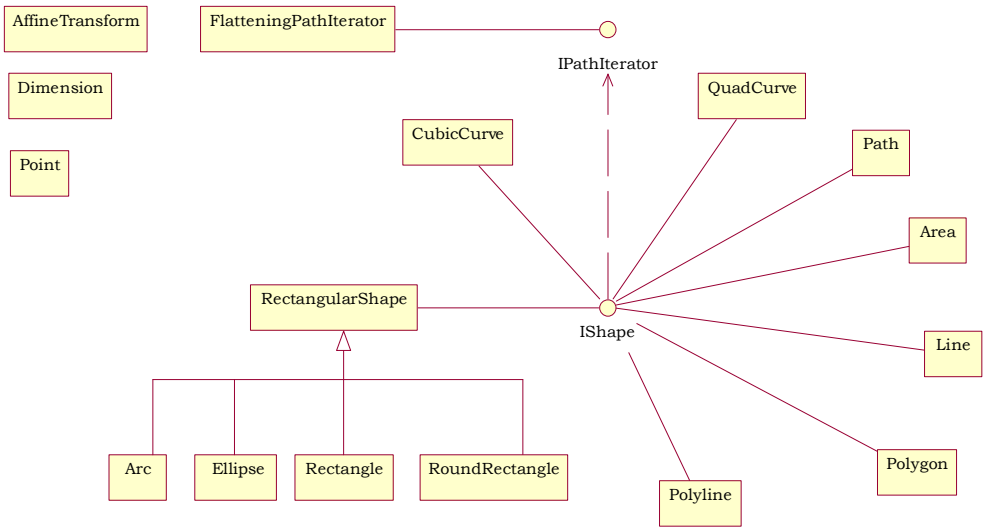


Brush	Classes derived from this abstract base class define objects used to fill the interiors of graphical shapes such as rectangles, ellipses, pies, polygons, and paths.
Color	The Color class is used to encapsulate colors in the default serge color space Every color has an implicit alpha value of 1.0 or an explicit one provided in the constructor.
Graphics2D	This Graphics2D class provides more sophisticated control over geometry, coordinate transformations, color management, and text layout.

LinearGradientBrush	The LinearGradientBrush class provides a way to fill a Shape with a linear color gradient pattern.
Pen	The Pen class defines a basic set of rendering attributes for the outlines of graphics primitives, which are rendered with a Graphics2D object that has its Stroke attribute set to this Pen.
RadialGradientBrush	The RadialGradientBrush class provides a way to fill a shape with a circular radial color gradient pattern.
SolidBrush	Defines a brush of a single color.
TextureBrush	The TextureBrush class provides a way to fill a Shape with a texture that is specified as an Image.

2.11 Package drawing.geometry

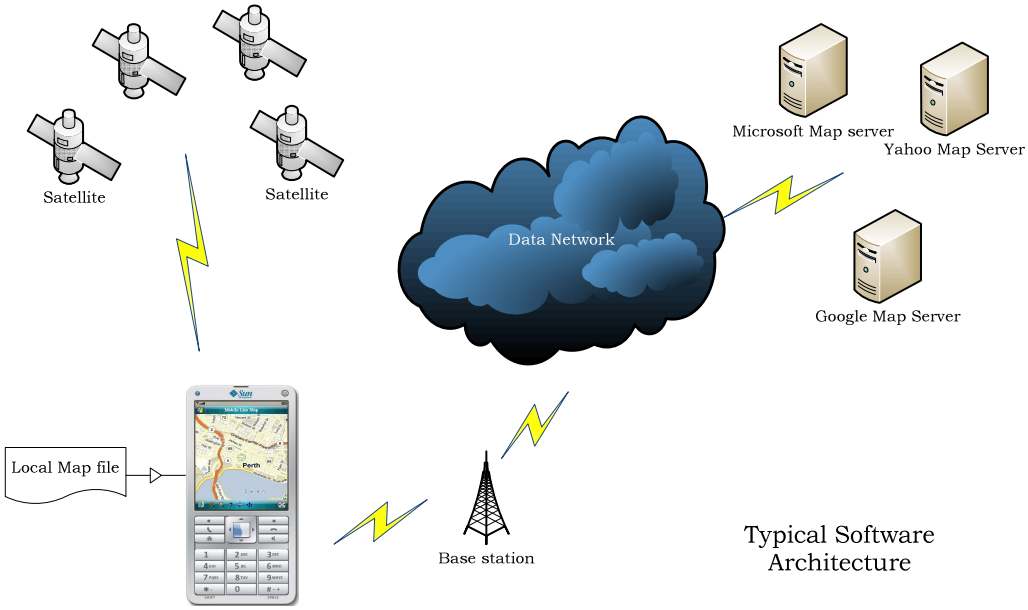
Package geometry provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.



PathIterator	The PathIterator interface provides the mechanism for objects that implement the Shape interface to return the geometry of their boundary by allowing a caller to retrieve the path of that boundary a segment at a time.
IShape	The Shape interface provides definitions for objects that represent some form of geometric shape.
AffineTransform	The AffineTransform class represents a 2D affine transform that performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the "straightness" and "parallelness" of lines.
Arc	Arc store a 2D arc defined by a framing rectangle, start angle, angular extent (length of the arc), and a closure type (OPEN, CHORD, or PIE).
Area	An Area object stores and manipulates a resolution-independent

	description of an enclosed area of 2-dimensional space.
CubicCurve	The CubicCurve class defines a cubic parametric curve segment in (x, y) coordinate space.
Dimension	The Dimension class encapsulates the width and height of a component (in integer precision) in a single object.
Ellipse	The Ellipse class describes an ellipse that is defined by a framing rectangle.
FlatteningPathIterator	The FlatteningPathIterator class returns a flattened view of another PathIterator object.
Path	The Path class represents a geometric path constructed from straight lines, and quadratic and cubic (Bezier) curves.
Line	This Line represents a line segment in (x, y) coordinate space.
Point	A point representing a location in (x, y) coordinates space, specified in integer precision.
Polygon	The Polygon class encapsulates a description of a closed, two-dimensional region within a coordinate space.
Polyline	The Polyline class encapsulates a description of a collection of line segments within a coordinate space.
QuadCurve	The QuadCurve class defines a quadratic parametric curve segment in (x, y) coordinate space.
Rectangle	A Rectangle specifies an area in a coordinate space that is enclosed by the Rectangle object's upper-left point (x, y) in the coordinate space, its width, and its height.
RectangularShape	RectangularShape is the base class for a number of Shape objects whose geometry is defined by a rectangular frame.
RoundRectangle	The RoundRectangle class defines a rectangle with rounded corners defined by a location (x, y), a dimension (w x h), and the width and height of an arc with which to round the corners.

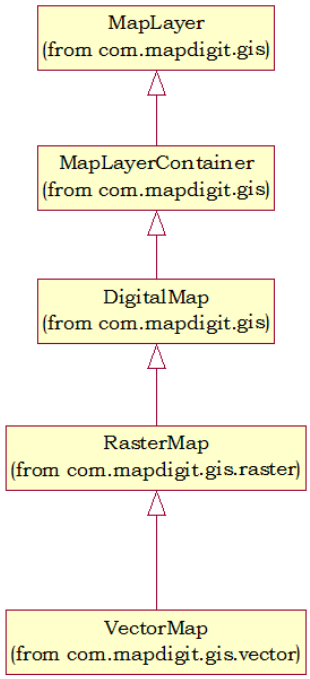
2.12 Typical Software architecture



3.0 Map Basic

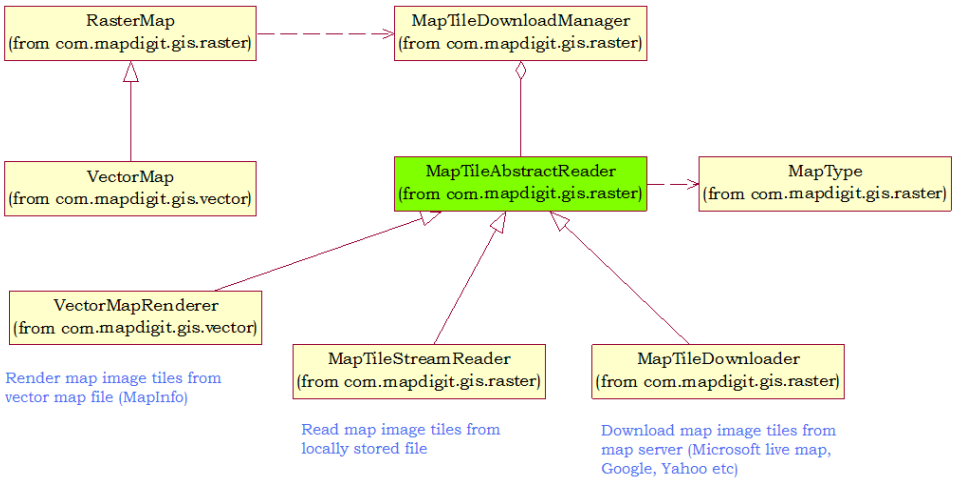
Guidebee Map API provide consistent interface for online, stored and vector map.

3.1 Map class hierarchy

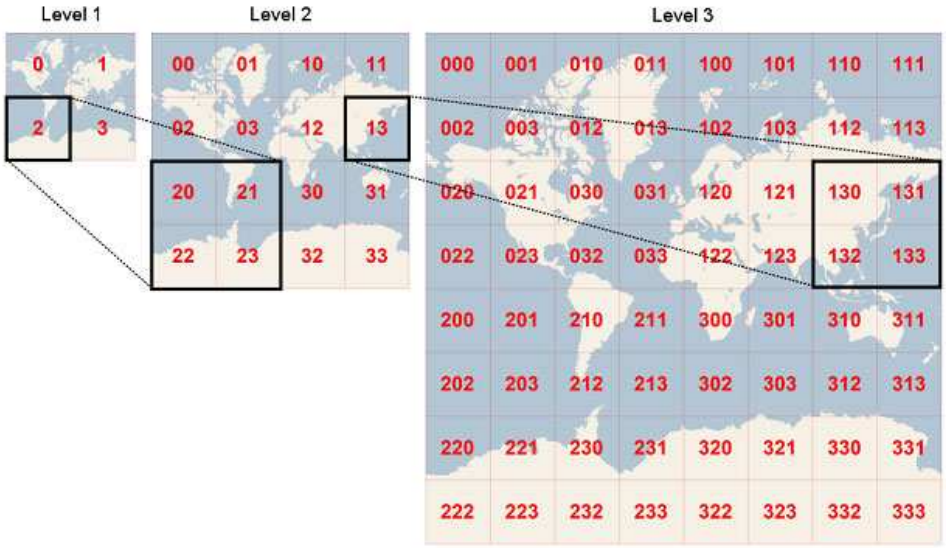


- MapLayer define common map operation, like setCenter, Zoom in, Zoom out, Pan Direction and coordinates conversion (from Map to screen and vice versa) and it's a common base class for all digital map.
- MapLayerContainer is container classes; manage a collection of Map layers allow them setCenter, Zoom In, Zoom out and Pan Direction together.
- DigitalMap introduces Map Services like Geocoding, Reverse Geocoding and Routing service. It supports incorporate different Map Services implementation.
- RasterMap deals with online, stored map images tiles, it support multi-threads and also provide listener to monitor the progress of downloading or reading map image tiles.
- VectorMap is a subclass of RasterMap, so it supports all functions provided by RasterMap, instead of downloading image tiles from map server or reading map image tiles from local files, Vector renders map image tile from MapInfo compatible vector map file also supports Geoset which managers multiple map feature layers.

3.2 Map tile reader hierarchy



RasterMap has a reference to MapTileDownloadManager, which manages all map tile reader work threads.
Most map servers like Google, Microsoft, Yahoo uses follow map tiles system, these map server divides the whole map into 256X256 map tiles.



- MapTileAbstractReader is the common base class for all map tile readers, from the RasterMap prospective, there's no difference for online map, stored map or vector map, they have each MapTileAbstractReader subclass to handle the differences for online map, stored map and vector map.
- VectorMapRender renders map image tiles from MapInfo compatible vector map data file.
- MapTileStreamReader reads map image tiles from an input stream (mostly from a local stored file).
- MapTileDownloader downloads map image tiles from map servers like Google Map, Microsoft Live Map, and Yahoo Map etc.
- MapDirectionRenderer renders routing direction map tile, Guidebee Map API internally use and is not exposed as public API.

Note: For developer want to support self-defined stored map or connects to different map server, the only things is to subclass MapTileAbstractReader and implements related interface and then Guidebee Map API will works smoothly with your self-defined stored map or your own map server.

3.3 Map service provider

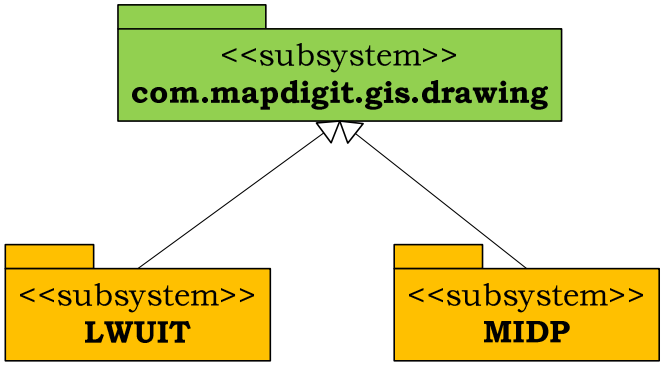
DigitalMap class has a reference to DigitalMapService, Digital Map Service provide Geocoding, reverse Geocoding and routing service. It's easy to switch map service providers for DigitalMap, for example, switch from Google map service to Microsoft map service or from local routing/Geocoding service.

The current default map services are Google map services.

Remember if you may need to apply for your own map key for different map service to legally use these map services or enhanced map services provided by Google, CloudMade or MapAbc. You can use MapKey and MapKeyRepository to manage these map keys.

3.4 Graphics Subsystem

As said previousely, an abstract graphics system is defined to let Guidebee Map API support multiple platforms with the same binary library. Here's an example on Java ME platform, you can use MIDP's LCDUI or Sun's LWUIT as the UI framework to write applications.



The demo code includes two sample concrete implementation for the com.mapdigit.gis.drawing package. One for standard MIDP library, one for Sun's LWUIT library.

LWUIT Hello world demo

Microsoft Live Map

MIDP Hello world demo

Google Map

As a matter of fact, the same Java ME library also works on Blackberry platform, developer for Blackberry can either write Java ME Midlet or write Blackberry application

using blackberry Java APIs. Guidebee also provides the Graphics system implementation for blackberry, so you can also use Guidebee Map API for blackberry applications.

The Gis.Drawing defines the minimum classes and methods necessary used by the API library. And if you don't want to use vector map, you can just provide empty implementation for some methods. And normally you don't need to implement these interfaces for your desired platform. Guidebee Map API package provide the default implementation for you already.

Here list the detailed classes and interfaces defined in Gis.Drawing package, it's quite a small function set so it's quite easy to support new graphics system if you want.

In 2.4 defines the following

AbstractGraphcsFactory	Factor class, used to create other graphics objects like font, graphics ,images etc.
IGraphics	Interface for Graphics object used to draw font, images etc,
IFont	Font interface.
IImage	Image interface.

Class AbstractGraphcsFactory Definition

Java Definition

Method	Description
abstract public IImage createImage (int[] rgb, int width, int height);	Create an image instance from rgb array. Rgb is the integer array, width, height are the size of the image. And the pixel format is ARGB array.
abstract public IImage createImage (byte[] bytes, int offset, int len);	Create an image instance for byte array. The byte array is the PNG image content.
abstract public IImage createImage(int width, int height);	Create an empty image instance with given width and height.

C# Definition

Method	Description
public abstract IImage CreateImage(int[] rgb, int width, int height);	Create an image instance from rgb array. Rgb is the integer array, width, height are the size of the image. And the pixel format is ARGB array.
public abstract IImage CreateImage(byte[] bytes, int offset, int len);	Create an image instance for byte array. The byte array is the PNG image content.
public abstract IImage CreateImage(int width, int height);	Create an empty image instance with given width and height.

Interface IGraphics Definition

Java Definition

Method	Description
public void setClip(int x, int y,int width,int height)	Set the clip region for this graphics. Can use empty implementation.
public void drawImage(IImage img, int x, int y);	Draw an image at given location.
public void drawLine(int x1, int y1, int x2, int y2)	Draw a line.
public void setColor(int RGB);	set the draw or fill color
public void fillRect(int x, int y, int width, int height);	Fill a rectangle.
public void drawRect(int x,int y,int width, int height);	Draw a rectangle.
public void drawString(String str,int x,int y);	Draw a string. Only needed for vector map.
public void setFont(IFont font);	Set font. Only needed for vector map.
public void fillTriangle(int x1, int y1, int x2, int y2, int x3, int y3);	Fill a triangle. Only needed for vector map.

C# Definition

Method	Description
void SetClip(int x, int y, int width, int height);	Set the clip region for this graphics. Can use empty implementation.
void DrawImage(IImage img, int x, int y);	Draw an image at given location.
void DrawImage(IImage img, int x, int y,int transparentColor);	Draw an image at given location.Only need for windows mobile. That's because windows mobile 6 doesn't support PNG transparency.
void DrawLine(int x1, int y1, int x2, int y2);	Draw a line.
void SetColor(int rgb);	set the draw or fill color
void FillRect(int x, int y, int width, int height);	Fill a rectangle.
void DrawRect(int x, int y, int width, int height);	Draw a rectangle.
void DrawString(string str, int x, int y);	Draw a string. Only needed for vector map.
void SetFont(IFont font);	Set font. Only needed for vector map.

Class IFont definition， IFont is only needed for vector map. For other circumstances it can have empty implementation.

Java Definition

Method	Description
public Object getNativeFont();	Get the native font object.
public int charsWidth(char[] ch, int offset, int length);	Calculate the chars width with this font.

C# Definition

Method	Description
object GetNativeFont();	Get the native font object.
int CharsWidth(char[] ch, int offset, int length);	Calculate the chars width with this font.

Class IImage Definition

Java Definition

Method	Description
public IGraphics getGraphics();	Get the graphics object associated with this image.
public int[] getRGB();	Get the RGB array of this image. Only needed for vector map.
public int getHeight()	Get the height of the image.
public int getWidth();	Get the width of the image.
public Object getNativeImage();	Get the native image assocated with this Image object.

C# Definition

Method	Description
IGraphics GetGraphics();	Get the graphcis object associated with this image.
int[] GetRGB();	Get the RGB array of this image. Only needed for vector map.
int GetHeight();	Get the height of the image.
int GetWidth();	Get the width of the image.
object GetNativeImage();	Get the native image assocated with this Image object.

4.0 Get Started

Guidebee Map API support most of platforms available on the market, different tools and different programming languages may be used for different platform. Guidebee Map API mainly provides Java and C# to support all platforms. The following examples use Java language, but the concepts apply for C# (.Net Framework) also. You can also check platform related examples for reference.

4.1 Guidebee Map API licenses

A valid developer license is needed for Guidebee Map API to work properly. The license can be a trial license or a commercial license. The trial license is free and you download it with sample code. But if you need to publish your application you need a commercial license.

The major difference between the two licenses, you get a "Guidebee" watermark on the map for the trial license and some functions may be disabled. From version 2.1, the trial time limitation is lifted. You can use the license as long as you want before you decide to get the commercial licenses.

You need to pay the license fee for the commercial license. For more information, please visit our website at http://www.guidebee.biz or send email to James.shen@guidebee.biz

4.2 Google ,CloudMade, MapAbc etc Map Key

Guidebee Map API internally supports Google Map Service, CloudMade Map Service and MapAbc Map Service. These map service provider have their own end user agreement. You may need to apply appropriate map key to use their map service legally or you want to use their enhanced map service.

Class MapKey and MapKeyRespository in Guidebee Map API are used to manage these map keys. Guidebee Map API choose proper map key for you to use their map services. But for development purpose, you can use the build-in map keys to run and testing your application.

The code below shows how you add a Google map key to the map key repository.

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

4.3 Map Configuration

There's some configuration may affect the performance and return result of some map services. Most map configuration is set via MapConfiguation or SearchOptions class.

Here lists some most often used map configurations.

4.3.1 The number of worker thread

You can configure the maximum number of worker thread used when download map tiles from map server. The default worker thread number is 4.

You may want to set the number of work thread to 16. Here's how:

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

4.3.2 Using Cache

With cache turned on, Guidebee Map API stores recently downloaded map tile in the cache to boost map display performance. Of course, cache requires extra memory. The default cache size is 256k. In some cases you may want to change the cache size or turn off the cache. MapConfiguration. MAP_CACHE_SIZE_IN_BYTES is used to set the cache size. And MapConfiguration. IS_CACHE_ON is used to turn on/off cache.

4.3.3 Search options

Search options are mainly used with map service, for instance, when you query an address, you may want to narrow down the return result to 10 or you want to change the return description to Chinese. SearchOptions is the class you can make such settings.

For instance, you can change the language to traditional Chinese using following

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

The default language is English: en-US.

4.3.4 Your first Map application

The following examples will use Java ME platform as the target platform, if you are not familiar with Java ME Midlet, only pay attention to the code in blue. The other parts are just the skeleton code for Java ME, just like Form_Load, Form_Close on .Net framework.

The following code display map of Perth, Australia.

```
//-------------------------------- 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,
            0x4ab0cff532902b1cL,};
            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);
   (10)   MapType.updateMapTileUrl();
        map = new RasterMap(1024, 1024, mapTileDownloadManager);
        map.setScreenSize(canvas.getWidth(),
                canvas.getHeight());
   (4)  mapTileDownloadManager.start();
        map.setMapDrawingListener(this);
   (5)  GeoLatLng center = new GeoLatLng(-31.948275, 115.857562);
        map.setCenter(center, 13, MapType.GOOGLEMAP);
       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. First is the setup the developer licence, the license information is consist of two part, one the license file (guidebee.lic), is normally put in the root directory. The other part is the license code corresponding to the license file.

2. As said before, Guidebee Map API internally doesn't call graphics related methods directly; it uses interfaces defined in Gis.Drawing package. Here's how you hook up platform dependent implementation to Guidebee Map library. Here for instance on Java ME platform, class MIDPFont， MIDPGraphics， MIDPGraphicsFactory， MIDPImage implement those interfaces. You need to tell the Map API where to look for these classes.

   `MapLayer.setAbstractGraphicsFactory (MIDPGraphicsFactory.getInstance ())` is used to setup the platform dependent GraphicsFactory.

3. Create an instance of MapTileDownloaderManager; this instance is used to manage all the worker thread. These work threads can download map tiles from server (online map), read map tile from local map file (offline map) or render vector map from vector map data (vector map). After the instance of MapTileDownloaderManager is created, an instance of RasterMap is created also.

   RasterMap uses MapTileDownloaderManager to get map tile images to draw on the internal map canvas.

4. MapTileDownloaderManager.Start starts all worker threads. And remember to stop the worker thread when application exit. Refer to step 6.

5. Set the center of the map. Here is the latitude, longitude of Perth, Australia. The map type used here is GoogleMap. There are more than 20 different map types you can choose, and you can also define your own map type if needed. The map zoom level here is 13. The maximum zoom level is 17 and the minimum zoom level is 1.

6. Stop all worker thread when application quits. Otherwise you may have hang threads preventing application exits.

7. PanDirection(dx,dy) moves map , it'll explained later.

8. Display the map on screen. In this application, MapCanvas is used to show the map on screen. These two lines display map on screen.

9. Asynchronized method is used when download map tiles from map server. i.e. There are worker threads which download map tiles in the background. In this way, the UI can keep responsive. The application can also monitor the download progress uses

listener method readProgress, and when each map tile finish downloading, done() is called by the Map library, done is where you update your map on screen.

10. This is optional. Map Servers like Google, Bing sometime update there map version. The URL may changes for these map server, if may get “image no available” exception if URL is not sync with latest map server. You can call updateMapTileUrl to get the newest URL from our server.

The examples in later chapter will not repeat common code, a base class MapDemoMIDP is used ,it has following definition.

```
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 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)) {
                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 Map Operation

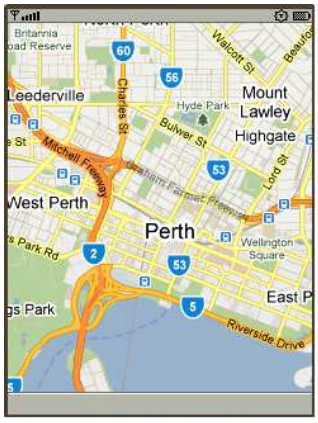
After create the instance of RasterMap, you can move, zoom in, zoom out or set different type of map.

5.1 Set Map Type

In the first example, When call RasterMap.setCenter, you can specify the map type. Besides RasterMap.setMapType can be used to change map type. There are more than 20 buildin map types, include Google map, Bing map, Bing Satellite map etc.

The following example change map type from Google Map, Bing Map to CloudMade Map (OpenStreet Map).

```
public class MapTypeMIDP extends MapDemoMIDP implements CommandListener {

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

    public void startApp() {

        init();
        canvas.addCommand(mapTypeCommand);
        canvas.setCommandListener(this);
        GeoLatLng center = new GeoLatLng(-31.948275, 115.857562);
        map.setCenter(center, 13, MapType.GOOGLEMAP);
         Display.getDisplay(this).setCurrent(canvas);
    }

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



Google Map

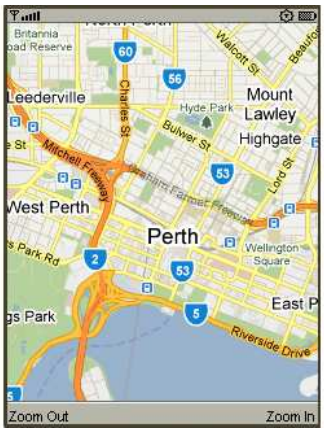
Bing Map

OpenStreet Map

5.2 Zoom In, Zoom Out

RasterMap's ZoomIn, ZoomOut are used to zoom in and zoom out the map.

```
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 Pan Map

There's two method normally ued to move map around, panTo move the center map to given latitude, longitude. panDirection (dx,dy) moves map to given dx,dy in pixel relative to current map location. panTo trigges the whole map refresh, so if you need constantly update map (like tracking current location) ,panDirection(dx,dy) provides a better update performance.

The following example can move map UP, DOWN, LEFT, RIGHT

```
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(-31.948275, 115.857562);
        map.setCenter(center, 13, MapType.GOOGLEMAP);
         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 Map Cache

As stated previously, RasterMap has an internal cache. If the map tile is in the cache already, RasterMap get the map tile from the cache directly, this will speed up map update.

The size of cache and turn on/turn off cache can be changed with class MapConfiguration.

Also remember the cache is tempory, the cache is cleaned after the application exits. If you need to store the cache persistently, so that next time user starts the application, you can display the map last time user visits. You can use saveMapCache and restoreMapCache methods to save and restore cache in a file.

6.0 Map Services

Guidebee Map API also provides methods to query address, get directions, local search, IP search, and search by latitude and longitude etc.

6.1 Geocoding

Address query or Geocoding, is to find an address's latitude and longitude according to its name (like James St, Perth).

The following examples show how to use the Geocoding services with ServerMap, it querys for 7 Fairway, Crawley, Australia. And then display the map in that area.

```
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(-31.948275, 115.857562);
        map.setCenter(center, 13, MapType.MICROSOFTMAP);
         Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if (c == mapFindAddressCommand) {
            String name = "7 Fairway, Crawley, Australia";
            map.getLocations(name);
        }
    }

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

All results returned by map service asynchronously.

Before call RasterMap.getLocation (address), a callback (listener) can be set with RasterMap.setGeocodingListener; the callback will be called when service returns.

The listener methods is defined as

public void done(String query,MapPoint[] result).

If the search has return , the result is the array of all returned result. the examples move the map to the first address found.

6.2 Get Directions

RasterMap's getDirections is used to get directions between two points or transit through a list of waypoints. The result is also returned with a callback function.

The following example returns the route between "Perth" and "Sydney".

The result is stored in an instance of MapDirection, which contains detail information like distance, driving instructions of each route, each steps.

```
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(-31.948275, 115.857562);
        map.setCenter(center, 13, MapType.GOOGLEMAP);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if (c == mapGetDirectionCommand) {
            String name1 = "perth";
            String name2 = "sydney";
            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();
        }
    }
}
```

For map service, there are Google Map Service, CloudMade Map Servcie and in China, MapAbc Map service to choose from. The default map service used is Google map service.

getDirections() has three overloaded methods. The example above uses the description format. If use string as the query input, CloudMade and MapAbc requires the following format “longitude1, latitude1, longitude2, latitude2”.

To avoid confusion, you can use following

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

Where waypoints are the transit points for the route. It may support multiple way points (>2) depends which service you use.

6.3 Local Search

You can use “Local search” to search business in given area, like search for café, restaurant etc. It has similar usage as Geocoding.

The following example searches for café around -31.948275, 115.857562.

```
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(-31.948275, 115.857562);
        map.setCenter(center, 13, MapType.GOOGLEMAP);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
        if (c == mapFindAddressCommand) {
            String name = "cafe";
            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);
            }
        }
    }
}
```

Local search method:

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

search based on the center and boundary. Local search can return mulptile results and can be called multiple time, start gives the start index of the result. you can specify the number of each return ,for google,each time is 4.

The above example's output is as following (maybe different in your case).

Cafe Cafe (08) 9388 9800 (08) 9388 7800

The Moon Cafe (08) 9328 7474

tiger, tiger coffee bar (08) 9322 8055

Monte Fiore Cafe Restaurant (08) 9227 9898

6.4 IP Search

You can also search address based on an IP address, Guidebee Map API also provide methods to query latitude, longitude base on a IP address.

For example, the following example search for IP 58.192.32.1, and the return latitude, longitude is 118.777802, 32.061699, which is Nanjing University in China.

```
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.longitude.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";
            }
        }
    }
}
```

Note: the result is always returned in English. A more detail information for above IP address is as below:

ISP： "China Education and Research Network"

Organization: "Nan Jing University"

Country: "CN"

City: "Nanjing"

You can also use “127.0.0.1” to query local address location.

6.5 Reverse Geocoding

Reverse Geocoding is to find a detail address name based on its latitude and longitude. The following example is to find what’s the address locate in -31.948275, 115.857562. which is "109-123 James St, Northbridge WA 6003, Australia"

```
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(-31.948275, 115.857562);
        map.setCenter(center, 13, MapType.GOOGLEMAP);
       Display.getDisplay(this).setCurrent(canvas);
    }

    public void commandAction(Command c, Displayable d) {
```

```
        if (c == mapFindAddressCommand) {
            map.getReverseLocations("-31.948275,115.857562");
        }
    }

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

Remember the string format, latitude comes first, and don't put space in between. The result is also returned as an array. Normaly only the first result is most appropriate, the rest is for bigger area, i.e the city, the state and then the country the address is in.

6.6 Set Map Service Type

Guidebee Map API default uses Google Map Service, but there's way you can choose which service you want to use. For example, if you applied CloudMade map key and you want to use CloudMade map service for your application. Or if you are in China, you want to use MapAbc map service instead of Google Map API.

Remember when use map in China, there's an "offset" for most maps (google map china, Bing map China, MapAbc Map) used in China. You need offset rectify data to adjust the location received from GPS device and then shown on these maps. CloudMade Map China map doesn't have such "offset".

The following example show how you choose map services in your application. It uses three different map services to get direction between Nanjing to Tianjing in China.

```
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 Map Servce

MapAbc Map Service

CloudMade Map Service

The above diagram display the route returned by Google Map Service, MapAbc Map Service and CloudMade MapService. You can notice there's "offset" for CloudMade sevice returned route.

This is because the Bing China Map itself is a map with offset, the returned result by CloudMade Map service is the actual coordinates. The google map china map service also returned the latitude, longitude with offset, that's why they are "matched" perfectly.

6.7 Add overlay on map

When developing Map based application, it's very likely you want to display some point of interests on top of the map or draw some shape on the map.

The RasterMap in the Guidebee Map API actually is an image.

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

So a simple way is to draw the POIs or shapes after you render the map on screen.

Remember to use proper coordinate system when draw on screen. The Map API uses latitude, longitude coordinate system while graphics system (screen) uses device coordinates. RasterMap has methods fromLatLngToScreenPixel to convert coordinate for latitude, longitude pair to x,y coordinate on screen. And method fromScreenPixelToLatLng to covert x,y coordinate on screen to latitude,longitude on map.

The following example uses a better solution to draw overlay on the map: define a subclass of MapLayer. RasterMap is also a map layer container, a sub class of MapLayerContainer, which can be used to manage mulptile map layers. Think of these layers as transparencies where each layer contains a different part of the map. The layers are stacked one on top of the other and allow you to see all aspects of the map at the same time.

The following examples display several POIs, a triangle and center cross on the map.

```
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);
        }
    }
}
```

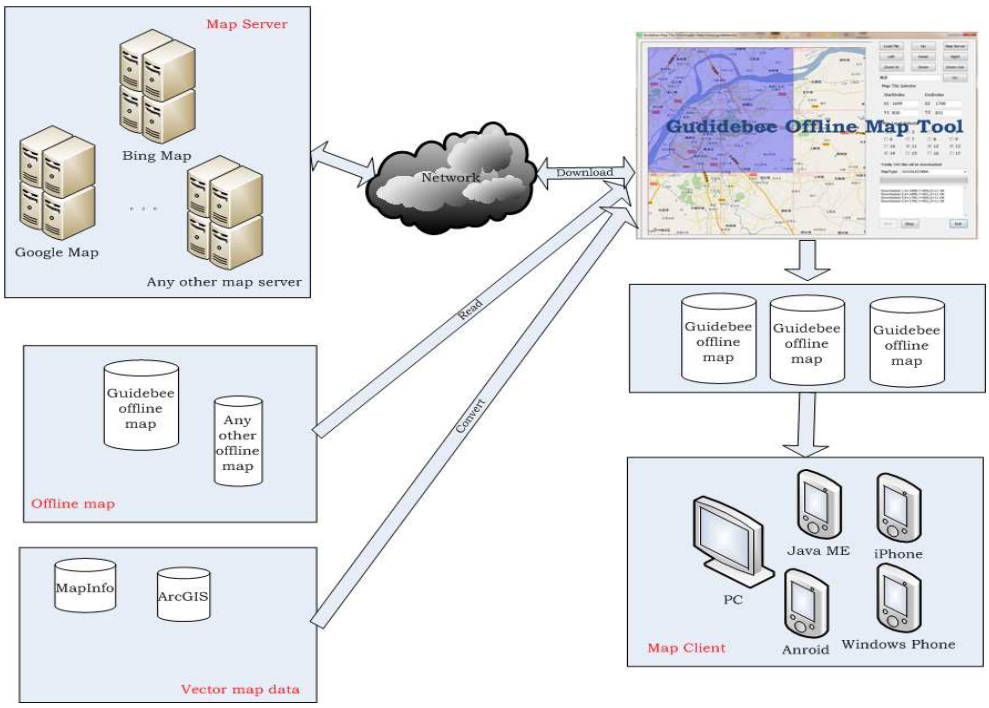
7.0 Offline Map

7.1 Offline Map Tool Diagram

All above examples require network connection, you need to work online. Some time offline map support is needed. With offline map support, mobile application can both save money for downloading and improve map response. An offline map tool is provided along with Guidebee Map API. It creates offline map file which works with Guidebee Map API. You application can be implemented like this, it tries to read map tile for locally stored map file ,if it finds requires map tiles it diplays the map from local map file, if the required map tile is not stored local then it try to download the map tile from server. Just remember the map service only works online.

The offline map tool can create offline map file from three different sources:

1. Map Servers like Bing Map, Google Map.
2. The offline map itself.
3. Vector Map Data (like MapInfo).

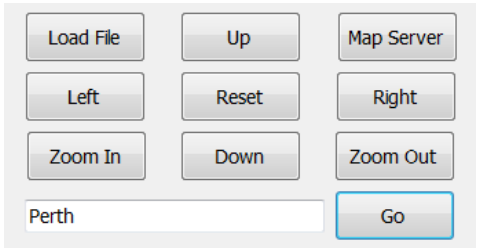


The generated offline map file include index for each map tile, it also stores the size of map area, zoom levels so that Guidebee Map API can quickly retrieve the map tile requested.

7.2 Quick manual of the Offline Map Tool

The tools requires.net framework 3.5 is installed on your computer.

Move map



Load File --> Load the downloaded .map file, you can view the download file or use it as a source to write another file.

Up, Left, Righ Down, move the map to one direction.

Reset --> Adjust the map tile to the screen.

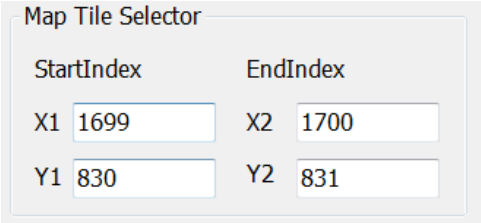
Zoom In, Zoom Out --> Zoom the map

Map Server --> after you read the download the file, you may want to reconnect to a map server, use this button to do so.

Go --> you can search an address and then move the map to the location you want.

You can use your mouse to drap the map just as what you normally do with google map.

Map selector



Before you create the offline map, you may need to select which area you want to download

Each map tile on screen has an index on the top left area,

You input the top, left and bottom right index for the area you want download

An easy way is to hold your rigth mouse button, and the drap the mouse, the selected are will be shown in blue.

Map Level Selector

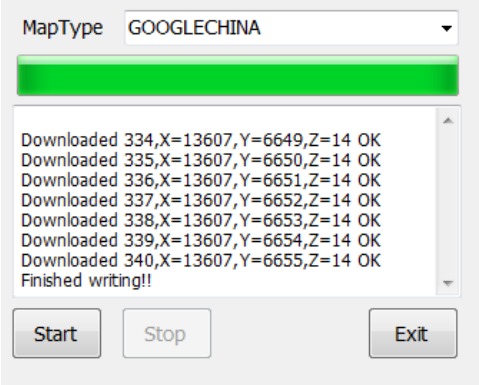
You also need to select which levels you want them to be loaded.

Assume your current map zoom level is 10 (shown at the left bottom of the map), you can not select zoom level which is below your current level, (i.e 1 -9).

You can select the levels above current zoom level here, like 10,11,12,13 ... 17

Totally xxx tiles will be downloaded, the xxx gives you how many map tiles will be downloaded

Map type selection



You can also select the map type, like open street map, Microsoft map etc.

Start --> start the downloading

Stop --> stop the downloading

Exit -> exit the application.

The message is display the current downloading status, and when the entire tiles are downloaded .it displays finished writing!

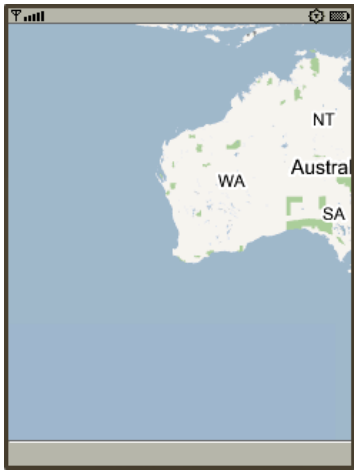
The file is downloaded at the same directory as the exe file and the file name format is StartX_StartY_EndX_EndY_ZoomLevel_MapType.map

7.3 Offline map demo

Normally the offline map is stored on SD card on mobile device, then use file access API to read these offline maps. For simplicity, this demo stores the offline map as the resource file. Guidebee Map API support multiple offline maps, this is done by class MapTiledZone and class MapTileStreamReader. Each offline map is mapped to one instance of MapTiledZone. And MapTileStreamReader provide methods to manage these map tiled zones.

```
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(-31.948275, 115.857562);
        map.setCenter(center, 3, MapType.GOOGLEMAP);
        Display.getDisplay(this).setCurrent(canvas);
    }

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

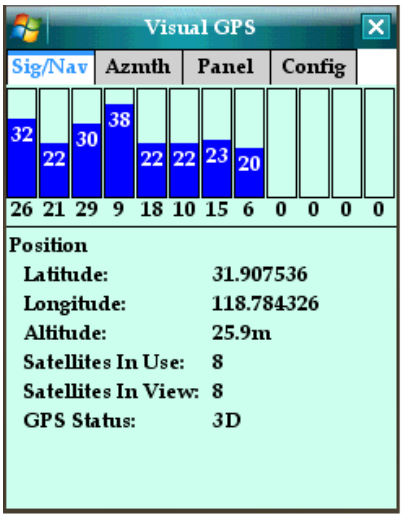


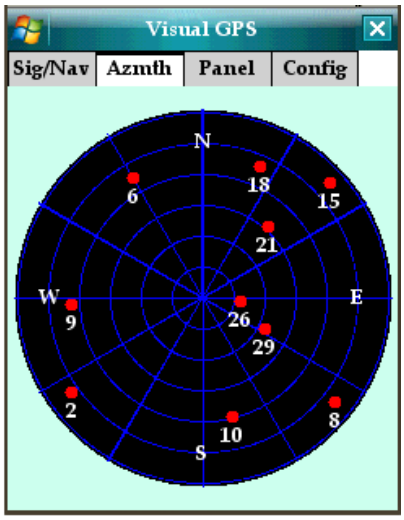
World03.map stores the world from zoom level 1 to level 4. MapTileManager has a reference to MapTileStreamReader. If the internal MapTileStreamReader is set, MapTileManager will first try to get the map tile from local map file. If the map tile is not found, then MapTileManager will try to download the map tile from map server. You can add multiple offline map files with method addZone, these map zone can have overlaps ,in this case, the order of addZone is important, the first map tile with given index is returned.

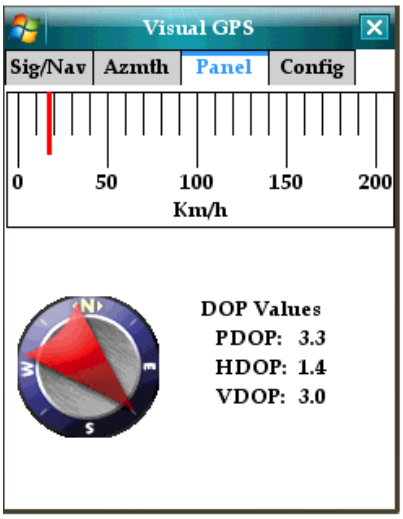
8.0 Location API

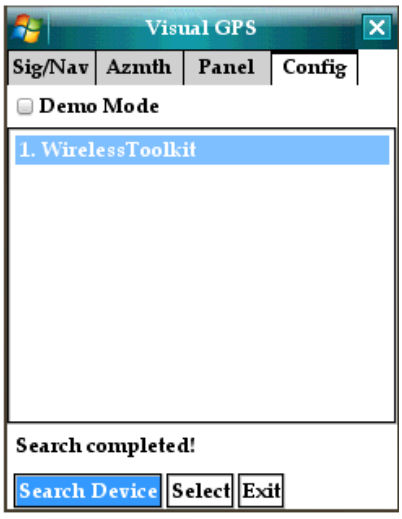
There's also location API included in Guiddebee Map API to support get location from GPS device. You can also use platform dependent location API together with Guidebee Map API, like on Java ME platform you can use JSR179 API; on iphone you can use core location API with Guidebee Map API.

The location API within Guidebee Map API has support data parsing for NMEA data .Here is the demo using guidebee location api ,you can the source code from our website.









9.0 Vector Map and Navigation API

Vector Map and Navigation API are more advance topics deserve their own developer guide. The Vector map API is defined in Gis.Vector package, which navigation API is defined in Gis.Navigation package. For more information, you can refer to our forum at www.guidebee.biz .