MiscGISKit

Licensing and Warranty

The MiscGISKit is the property of Genesis Project Ltd. It has been donated to the MiscKit and as such falls under the MiscKit license agreement.

What is MiscGISKit?

MiscGISkit is a set of tools for working with coordinate systems, particularly those related to maps and globes. The set of tools will grow and expand over time and at present consists of classes for supporting and translating between Universal Transverse Mercator grid coordinates in the International, the UK or the Irish systems and World Coordinates, the familiar latitude and longitude values. The equations used are accurate to 1 mm going from World coordinates to UTM and approximately .001 seconds of arc going from UTM to world coordinates. The number of digits of precision in double floating point numbers is greater than the number required to express this, but do not be fooled into believing the extra digits are significant!

There are also tools for supporting and translating between the familiar mathematical coordinate systems: cartesian, cylindrical and spherical.

Using MiscGISKit in your Project

Use of the MiscGISkit in your app is quite simple.

First, you must add the library search path to your Makefile.preamble. As an example:

```
OTHER_CFLAGS = -L/LocalDeveloper/Libraries
```

Second, start up the Project Manager for your application and then drag /LocalDeveloper/Libraries/libMiscGIS.a from the Workspace Manager to the Libraries folder in your project. If you just created a Makefile.preamble file (in step one), drag it from the Workspace Manager to the Supporting Files folder of your project.

Third, add

```
#import      <misckit/miscgiskit.h>
```

to each source (.h or .m) file in which you use the kit.

Now all you have to do is write some code!

Getting Started with MiscGISKit

You will find a great deal of documentation in Documentation/MiscKit/MiscGISKit. Fortuneately, you need only understand a small fraction of it to simply *use* the package. For this, it is recommended that you read at least the first two levels of the following plus the specific Coord types of interest.

- MiscCoord.rtf
  - MiscWorldCoord.rtf
  - MiscPlanetCoord.rtf
    - MiscUTMCoord.rtf
    - MiscZoneUTMCoord.rtf
      - MiscUKUTMCoord.rtf
      - MiscIrelandUTMCoord.rtf
      - MiscIrelandOldUTMCoord.rtf
  - MiscMathCoord.rtf
      - MiscCartesianCoord.rtf
      - MiscCylindricalCoord.rtf
      - MiscSphericalCoord.rtf
- MiscCoordConverterClient.rtf

Diagram 1 will give you an idea of how the other classes relate to instances of MiscCoord and its subclasses. Dotted lines are used for targets, ie an object hierarchy; dark lines for inheritance in the class hierarchy.

Next, examine and understand the code in the example to be found in Examples/MiscKit/MiscGISMapCoord.

The remainder of the class documentation is of interest only to developers and midnight hackers. You are on

your own if you tread there.

Diagram 1: Inter-relationship of MiscGISKit Classes

ClassDiagram.eps ¬

Known problem areas and Future Features

There is not as yet a means for translating directly between two different UTM systems. This must be carried out in two steps. First translate the UTM coordinate to World Coordinates; then translate it from World Coordinates to the target UTM system. Capability to do this directly will probably be added in a future release.

There is no means of translation between the geographical and the mathematical coordinate systems at this time. When I come up with a reasonable definition of how to relate them, it will be added.

Constants objects are not yet supported in mathematical coordinate systems. In some future release the constants will contain the matrices necessary to translate between the coordinate point and some base coordinate frame from which it was derived.