About:	
Designed and Developed By	**KIRAN PURANIK**
Email	**GkiranP@gmail.com**

Table of Contents

PREREQUISITES

Below table lists components required to compile and run the application,

Application Name	tz
Operating System	Linux or Unix based Operating System
Programming Language	C and C++
Third Party Components	1. Shapefile C Library [shapelib-1.3.0] 2. tz_world, an efele.net/tz map [tz_world_mp\world\tz_world_mp]
License Reference	1. Shapefile C Library [http://shapelib.maptools.org/] 2. tz_world, an efele.net/tz map [http://efele.net/maps/tz/world/]
Compiler	GCC 4.7 and above
Build System	Makefile based build system

APPLICATION DESIGN DETAILS

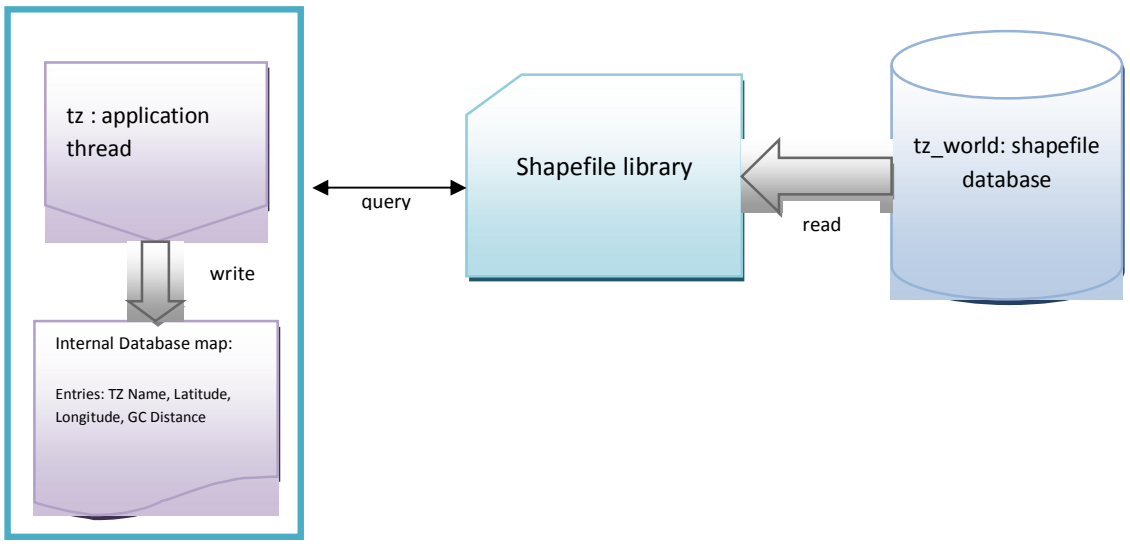
This section describes the application design information. Application uses 2 third party components, namely,

1. Shapefile C Library: The Shapefile C Library provides the ability to write simple C programs for reading, writing and updating (to a limited extent) ESRI Shapefiles, and the associated attribute file (.dbf). Please refer to this link [http://shapelib.maptools.org/] for more details and license terms and conditions.

2. tz_world, an efele.net/tz map: The tz_world shapefile captures the boundaries of the TZ timezones across the world. The geometries are all POLYGONs, and a TZ timezone will sometimes have multiple polygons. There are about 28,000 rows. Please refer to this link [http://efele.net/maps/tz/world/] for more details and license terms and conditions.

APPLICATION INITIALIZATION

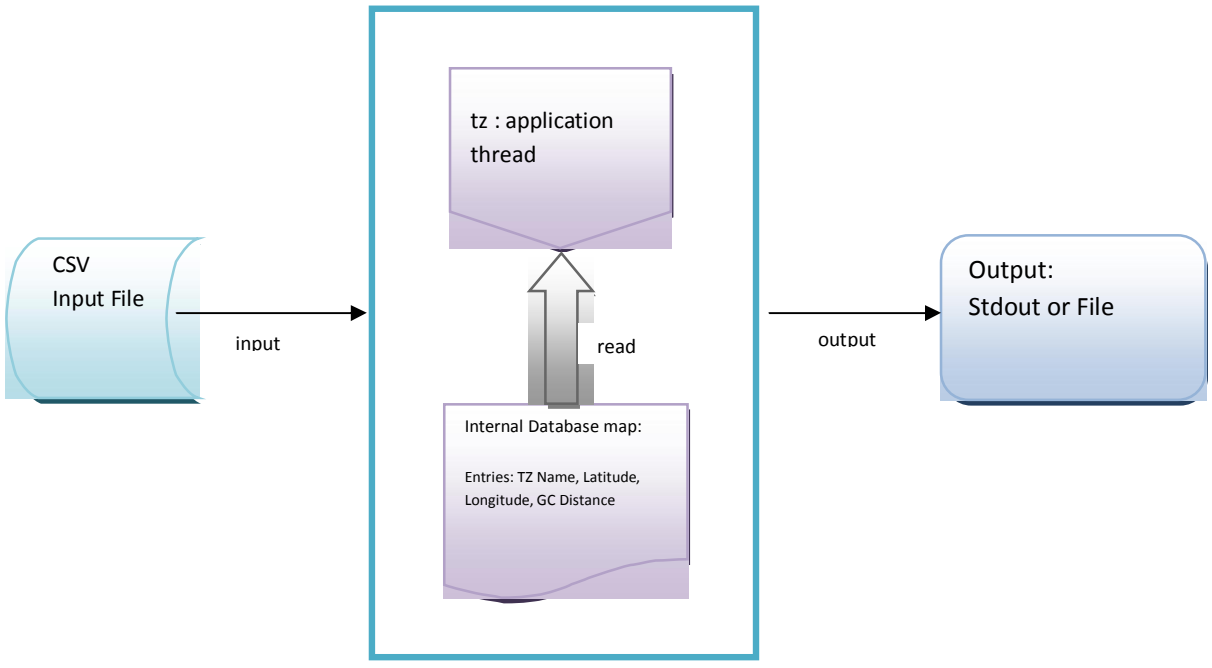
Below block diagram depicts the initialization of application.



As shown in above diagram, **tz** application, during initialization stage, parses all the entries in tz_world shapefile world map database and stores below entries in a container (Internal Database Map). These entries will contain City name/timezone information, latitude, longitude and great circle distance calculated.

APPLICATION PROCESSING

Below block diagram depicts the application process information, when user inputs a CSV file to parse,



As shown above, **tz** application will get CSV input file, parse the contents of it and looks into Internal Database Map for corresponding latitude and longitude information.

There is a possibility multiple entries of matching latitude and longitude may be found for one entry in CSV input file. In this case application will use Great Circle distance parameter to determine proper entry matching it. Below example describes the logic behind this GC distance, determined in this application;

For example:
Let user input latitude and longitude be,
UI_Lat = -33.912167 and UI_Lon = 151.215820
When enquired with Internal Database Map, application returns,
`Pacific/Auckland,-29.2411,179.102,-52.6076,-178.628,2613.19`
`Australia/Sydney,-28.1601,153.639,-37.5053,141,1654.97`
As you can see Internal Database Map outputs will be of format:
<City name>,<Latitude-N>,<Longitude-E>,<Latitude-S>,<Longitude-W>,<Actual_Distance>
Since, there are multiple output entries corresponding to user input, application will calculate two GC distances, namely DistanceMin and DistanceMax, by below method;

```
DistanceMin = GC_DISTANCE(<Latitude-S>,<Longitude-W>, UI_Lat,UI_Lon)
DistanceMax = GC_DISTANCE(UI_Lat,UI_Lon, <Latitude-N>,<Longitude-E>)
```

Now, for each of entries, calculated Distance_Min and Distance_Max, the original timezone will be found using below condition:

```
{Distance_Min <= <Actual_Distance>} AND {DistanceMax <= <Actual_Distance>}
```

If above condition is met, the city name will be attached with given file entry and sent to output.

APPLICATION ADVANTAGES AND LIMITATIONS

Below are advantages of “tz” application:

1. Application accepts CSV file as user input and automatically parse it to get required information and displays corresponding output quickly.
2. Application has provision to save output into another file directly, without displaying anything on standard output.
3. Since application uses tz_world pre installed map, which has highest precision information, the output displayed are accurate.
4. Since map files are dynamically loaded during application runtime, user can update tz_map and entries from outside with more recent updates.
5. Application uses Shapefile library to read from database map, which is known to be highly stable open source library for shape file read-write operation.
6. Since application, during initialization stage, reads and prepares Internal Database Map of all the entries in tz_world, the processing time is much faster than traditional external database query based applications.
7. The whole application is written using C and C++ programming language, which is industry proven fastest and accurate programming language.

Below are the limitations of this application:

1. Since application depends on third party tz_map database, the entry precision may vary compare to other very high precision map databases. More the precision the database entries will be, more accurate the output will be.
2. The GC distance calculated in application may not be accurate, as it depends on number of accurate parameters.