**GISC 6388 – GIS Application Development**

**Final Project Report**

**A Geocoding Locator of UT Dallas**

*By Caiyi Zhong*

1. **What is Geocoding?**
   1. **Geocoding Concept and Issues**

Simply said, geocoding is a process to convert place names, street addresses, postal codes, etc. to its associated geographic coordinates that is often expressed as latitudes and longitudes. The result of geocoding can either be mapped then entered into databases, or directly be evaluated and analyzed.

Geocoding is a bridge that connects “human-read” data and “machine read” data. It is widely used in web based map applications, data analysis and many more aspects.

* 1. **Potential Users**

There is a wide range of users of geocoding. A tourist may want to know how to get a feature spot by searching on a web based map with that location’s name, while the government may be able to create a map showing the distribution of crimes by geocoding a table of addresses where crime occurs.

* 1. **Geocoding applications**

There are many existing on-line geocoding services nowadays. But to make our own applications that may be used to solve certain problems, we need to use geocoding APIs provided by Google, Microsoft, Yahoo, etc. to build web based applications (using http interfaces). Besides, we can also use ArcObjects to build desktop applications.

1. **Platforms**
   1. **Development**

Microsoft Visual Studio 2010 + ArcObjects SDK for .NET

* 1. **Deployment**

ArcMap 10.1

1. **Application Appearance and Features**
   1. **Features**

This application would be deployed to ArcMap as an add-in toolbar with one status bar and several buttons:



The status bar is made with a read-only combo box indicating whether the locator has been loaded. The locator is originally created and saved in the hard drive, so we need to load it before using it.

**Button 1: Open or Create Locator**

Select a folder which contains the UTD Locator. If UTD Locator does not exists in this folder, try to find the reference data. If reference data exists, use it to create the UTD Locator and open it. If reference data *utd\_road.shp* does not exist either, report an error.

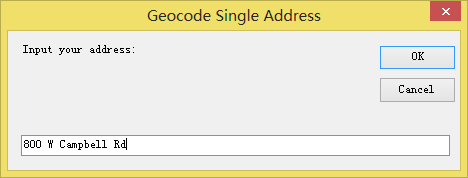
The status bar would be changed after opening the locator successfully.



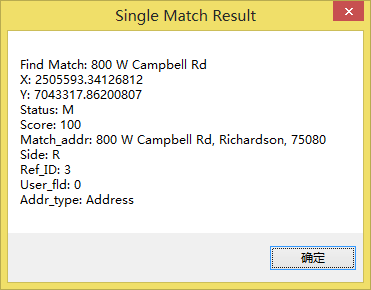
After the locator has been opened, the reference data would also be added to the map. But it is recommended to open the prepared *UTDallas.mxd* first to have a better performance.

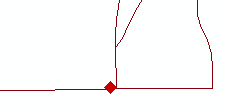
**Button 2: Geocode Single**

Geocode a single line of address.



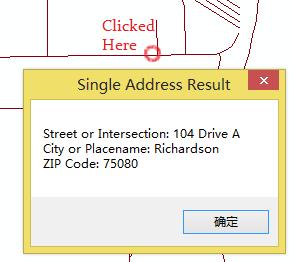
Related information could also be shown in a message box. The result would also be shown as a diamond marker on the map.

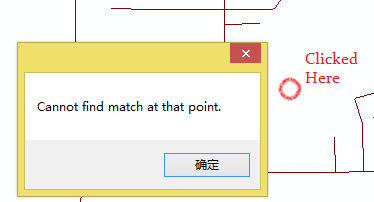




**Tool 3: Reverse Geocode Single**

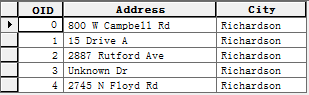
Allow user to click anywhere on the map to perform a reverse geocoding.





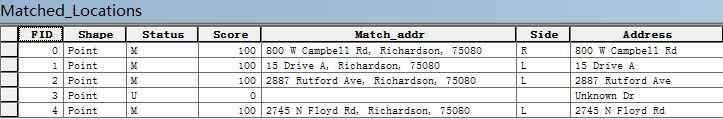
**Button 4: Geocode Table**

Geocode a table of addresses. Before using this function, users must add a .dbf table into the current document. The table must also have specified field names. Like below:



The result could be either multiple markers or a point feature class. Illegal input will not be reverse geocoded. E.g. the 4th line above.

To use it, first select the table you want to geocode in Table of Contents, then click this button. Then choose whether make the output as a new point feature class or not. If not, the output would be shown as multiple markers.



**Button 5: Clear Markers**

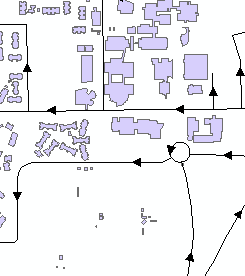
Clear all the markers (added by Geocode Single or Geocode Table) on the map.

1. **Implementation**
   1. **Collecting Data**

Data Source: City of Richardson GIS Data

<https://www.cor.net>

Clipped the UTD part out of the whole city.

**Issues on this:**

1. Data is incomplete or incorrect:

- Some of the roads will not present

- The direction of road may be wrong

- The fields does not meet the needs to create locators

Solve: Manually edit data



Use this symbol to show the direction of roads. The correct direction should be east to west, or south to north. If we find any incorrect direction, use Flip Line tool. It can flip the from-to property of selected lines of a polyline feature class.

Use google map and UTD campus map to create roads that does not exist in the GIS data.

Manually edit the data’s data fields to make them fit the need of creating locator.

2. Some on-campus roads does not have street numbers. E.g. Drive A, Armstrong Drive, Loop Road

Solve: Artificially specify street numbers for them. Normally assign 0-98 (right) and 2-99 (left) for them.

* 1. **Creating Locator**

To create locator, our reference data must have such fields:

**PreType**: Street type prior to name. E.g. **Ave.** Green

**PreDir**: Direction prior to name. E.g. **W** Campbell St.

**StreetName**: Name of the road.

**SufType**: Street type after the name. E.g. W. Campbell **St.**

**SufDir**: Direction after the name. E.g. Ave. Green **NW**

**FromLeft**: The lowest street number on the left. Must be odd number.

**ToLeft**: The highest street number on the left. Must be odd number.

**FromRight**: The lowest street number on the right. Must be even number.

**ToRight**: The highest street number on the right. Must be even number.

**CityLeft**: The city name on the left of the road.

**CityRight**: The city name on the right of the road.

**ZIPLeft**: The ZIP code on the left of the road.

**ZIPRight**: The ZIP code on the right of the road.

Then we can follow these steps to create our needed locator:

1. Create locator manager with ILocatorManager2.
2. Set the locator style. In this case *US Address – Dual Ranges*
3. Prepare and assign the map fields.
4. Create locator workspace using GetLocatorWorkspaceFromPath() method of ILocatorManager2, and then add the locator using AddLocator() method.

*Reference [1]*

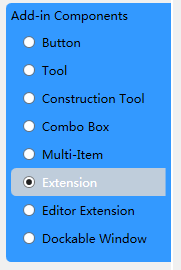
* 1. **Wrap into Add-in**

Finally we can create the locator. See *Implementation* chapter.

1. **Issues and Solution**

**Issue on Data Exchange Among Add-in Components:**

E.g. Geocoding could be executed only if the locator has been loaded to program. The status of locator (i.e. whether the locator has been loaded or not) could be a shared data. That means, data exchange occurs between the *Open or Create Locator* button and the status bar.

**Solution for this:**

1. Add an *Extension* component. It is designed to achieve inter-component communications.

Shared data could be defined as the extension class’ member variables. And this class may also have a shared member which holds the reference of itself and a public method which returns this member. Then, other components could call this method to get the reference to the extension, and then fetch the shared data.

1. Directly get the reference of other component using Addin.FromID() method.

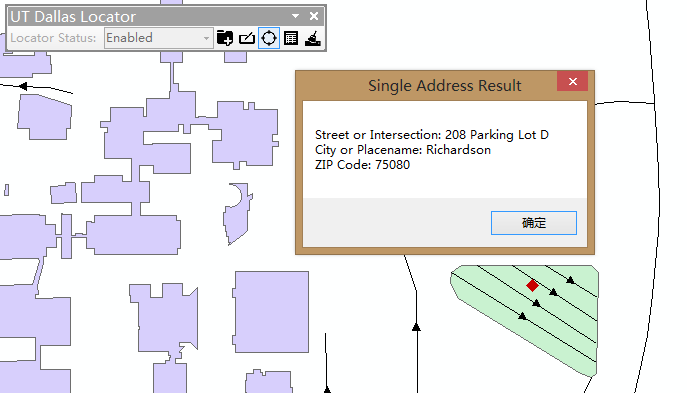
Simply import the ESRI.ArcGIS.Desktop.AddIns namespace and call FromID() method to get a certain component’s reference.

*Reference [2]*

1. **More We Can Do**

This system could be extended. Reference data could be updated, and all we need to do is just rebuild the locator.

If we have time, we can create a much more detailed reference data table.



After I create data for Parking Lot D in UTD, we can geocode such address: “208 Parking Lot D”. Just imagine we can use this to locate cars.

What’s more, we can extend our locator into buildings. In UTD, every building has a name and an abbreviation. After adding “roads” into buildings, we can geocode rooms within buildings with some “addresses” like this: “2102 Green Hall”, or “2102 GR”. Which can help people locate class rooms.

1. **References**

[1] Creating an Address Locator, ESRI, ArcGIS Resources

<http://resources.arcgis.com/en/help/arcobjects-net/conceptualhelp/index.html#//0001000004pm000000>

[2] Add-in Coding Patterns, ESRI, ArcGIS Resources

<http://resources.arcgis.com/en/help/arcobjects-net/conceptualHelp/index.html#//0001000000zz000000>