O'REILLY® Short Cuts

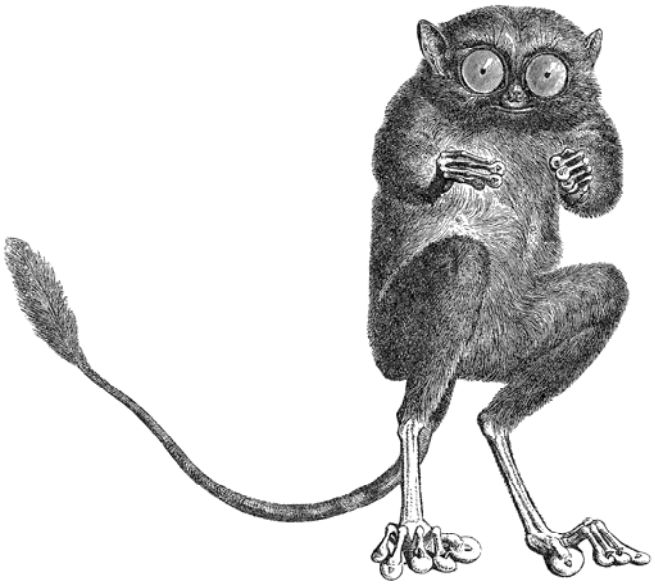
Introduction to Neogeography

By Andrew J. Turner

ISBN: 978-0-596-52995-6
Release Date: December 15, 2006

Neogeography combines the complex techniques of cartography and GIS and places them within reach of users and developers.
This Short Cut introduces you to the growing number of tools, frameworks, and resources available that make it easy to create maps and share the locations of your interests and history. Learn what existing and emerging standards such as GeoRSS, KML, and Microformats mean; how to add dynamic maps and locations to your web site; how to pinpoint the locations of your online visitors; how to create genealogical maps and Google Earth animations of your family's ancestry; or how to geotag and share your travel photographs.

Contents



Introduction to Neogeography

This Short Cut introduces current techniques of online mapping and neogeography. It covers the basic tools for finding out where you are, creating your own maps, and several illustrative projects to get you started and hopefully inspire your own projects.

The code and software examples from this book are available online at **http://mapsomething.com**. This includes a complete listing of the links to sites and resources listed here for easy bookmarking. Feel free to let me know if you have any questions, or want to show off any neat ideas you put together, directly at **shortcut@highearthorbit.com**.

What Is Neogeography?

Cartography enabled and recorded exploration and discovery for ages. It guided sailors across oceans and helped make sense of the wilderness. Like the transformation of paper to hypertext, maps have evolved from arcane lines and arcs on parchment to dynamic displays of remote geographic information. While geographic information systems (GIS) remained expensive programs, restricted to the use of highly-trained specialists, tools like MapQuest and Yahoo! Maps brought easy-to-use mapping tools to the public. More recently, the release of Google Maps demonstrated to web developers and users the possibilities of navigation and opened a floodgate of interest in online mapping.

Google Maps was not released with an *application programming interface* (API), but developers slowly figured out how to use the maps for their own uses, and eventually Google released a public API. The release of the API allowed developers and users to quickly and easily show geographically based data on shareable maps. Once the primary domain of GIS systems and operators, these new maps, or “mashups,” led to a massive increase in their propagation. The very term *mashup* can trace its lineage to the first mapping mashups like Housing Maps and ChicagoCrime. What is it about maps that both fascinates and educates people? Why in the online world of instantaneous global communication and anonymity has "location" become such an important theme?

Neogeography means “new geography” and consists of a set of techniques and tools that fall outside the realm of traditional GIS, Geographic Information Systems. Where historically a professional cartographer might use ArcGIS, talk of Mercator versus Mollweide projections, and resolve land area disputes, a neogeographer uses a mapping API like Google Maps, talks about GPX versus KML, and geotags his photos to make a map of his summer vacation.

Essentially, Neogeography is about people using and creating their own maps, on their own terms and by combining elements of an existing toolset. Neogeography is about sharing location information with friends and visitors, helping shape context, and conveying understanding through knowledge of place.

Lastly, neogeography is fun. Why else would people create a map of the locations of the television show *24*, or share the location of their first kiss? Never again will you struggle to recall "Where was that photo taken?"

Basic Terminology

Several useful terms are essential to discussing neogeography. Some are familiar, while others have specific contexts within a broader meaning. They are presented here for later referral as you progress through the book and are listed in a logical order so that terms build upon one another:

Coordinates

The geographic coordinates are the absolute position on the Earth (or any body). Typically these are Latitude and Longitude referenced to the WGS84 ellipsoid. Sometimes they will be referred to as "lat" and "lon." Latitude varies North-South, with 0 degrees at the Equator, varies from 90 to –90 degrees towards the poles, and is positive North. Longitude varies East-West, with 0 degrees at the Prime Meridian, varies from 180 to –180 degrees and is positive to East.

Coordinates can be represented in several formats:

- Decimal degrees (DD): 29.975
- Degrees-Minutes-Seconds (DMS): N29° 58' 30"
- Degrees-Minutes (DM): 29° 58.8'

The conversion is straightforward. There are 60 seconds in a minute, and 60 minutes in a degree.

Projection

A projection is required in order to display the three-dimensional Earth, which isn't a sphere but actually an oblate spheroid bulging in the center, onto any other shape. Typically, this is projection onto a two-dimensional map display, where Mercator and Rectangular (i.e., no transformation) projections are the most common. For example, Google Maps uses the Mercator Projection, which is good for zoomed-in viewing, but causes distortions when zoomed out. It is important to understand the implication of various projections depending on application and also when mixing together mapping providers.

POI (Points of Interest)

Points Of Interest, frequently abbreviated as POI, represents any significant locations such as public buildings, traveler's services, or user-defined waypoints. These may be categorized: restaurant, trail head, friend's house, scenic overlook, or scuba diving site. There is also AOI, Areas of Interest, which may include multiple POI or just a geographic area instead of a single point.

Extents

The bounding box, or farthest latitude and longitude of an AOI. This may also be referred to as just *bbox*. The extents define the Northern and Southern latitude, and the Eastern and Western longitude. Extents are the simplest means of specifying the area of interest and are usually used for web service queries.

Tiles

Dynamic, or *slippy* maps, are composed of a set of individual square images. Each image is a tile. Together, these tiles are placed next to one another, or stitched, to give the impression of a large, sliding map.

Geolocation

The technique of automatically determining the position of something based on measured data. For example, it is possible to locate a computer given its IP address, or a mobile phone based on the observed cell towers. Geolocation is useful for determining where a user or device is without the user having to manually enter this information. GPS is a specific implementation of a geolocation technology.

GPS

GPS (Global Positioning System) really refers to the U.S. military owned and operated satellite network that provides three-dimensional location. GPS is also sometimes used to refer to any means of geolocation that provides geographic coordinates.

Simply put, GPS operates by a network of high-altitude space satellite broadcasting their position and time. Receivers use several of these observed broadcasts to determine its current position and time.

The European operated GPS system, Galileo, is currently expected to be operational by 2010 and will provide similar, but alternate, functionality to the current GPS.

Geotag

Adding location information to a document, photograph, audio sample, or some other type of data is an example of geotagging. Geotagging formats are not uniform and vary based on the type of document they modify. For example, a photograph can embed location information in the EXIF header of the file itself,

or in many web applications the location information can be stored as triple-tags in the user specified tags.

Web Service

A web service is a resource that allows access to data or functionality from a provider—for example, a geocoder that takes an address and returns latitude and longitude is a web service, or a service to ask for all photos within 50 miles of a location. Web services typically use REST (Representational State Transfer) or SOAP (Simple Object Access Protocol) to allow programs and other sites to access the data.

Data Formats

Location can be stored in many formats. With roots in programming, neogeography pushes data to be stored in a clear and readable "plain language" format. Because broadband allows data to be conveyed quickly, more geographic information can be added to a greater variety of files.

While these formats have adopted a more human-friendly format, they can quickly become large and complex to write by hand. Therefore, in future sections we'll discuss tools that make it easy to add and read these formats to your projects. For now, this section will introduce you to the overall concepts of the data formats, when it's appropriate to use them, and what their capabilities are.

GPX

Vendors and tools use several standards to describe geographic information. The most common file format standard is GPX, or the GPS Exchange Format. This standard uses an XML file definition to store waypoints and tracks from GPS units. Most GPS receivers internally use their own proprietary file format. Therefore, GPX is a common protocol that allows developers to write tools to convert from and to an application's or device's specific format.

The header of a GPX file stores general information such as the GPX version, the application or device that created the file, and various other XML namespace information:

```
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.0"
      creator="GPSBabel - http://www.gpsbabel.org"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.topografix.com/GPX/1/0"
      xsi:schemaLocation="http://www.topografix.com/GPX/1/0
      http://www.topografix.com/GPX/1/0/gpx.xsd">
```

Below that, there are lists of waypoints, tracks, or routes.

A *waypoint* (`wpt` in the file example below) stores the latitude, longitude and optionally the altitude of the location. The body of the waypoint then includes the name, comment, and symbol:

```
        <wpt lat ="42.277881" lon ="-83.740791">
            <name>01WiFi</name>
            <cmt>ESPRESSO ROYAL</cmt>
            <sym>Tall Tower</sym>
        </wpt>
```

A *track* (`trk` in the file example below) is a successive list of waypoints and usually include a time element. This is useful later for geotagging photos or videos, as well as for creating animations of your tracks by providing a time to synchronize with the camera or recording device:

```
<bounds minlat="42.423656691" minlon ="-83.493977330"
              maxlat="42.502436669" maxlon="-83.144850082" />
<trk>
        <trkseg>
              <trkpt lat="42.500937812" lon="-83.147198063">
                    <ele>-0.846357</ele>
                    <time>2006-06-26T21:42:38Z</time>
              </trkpt>
              <trkpt lat="42.500988805" lon="-83.147192333">
                    <ele>-1.071942</ele>
                    <time>2006-06-26T21:42:40Z</time>
              </trkpt>
              <trkpt lat="42.500985367" lon="-83.147242180">
                    <ele>-0.691822</ele>
                    <time>2006-06-26T21:42:42Z</time>
              </trkpt>
        </trkseg>
</trk>
```

The Topografix GPX site (**http://www.topografix.com/gpx_resources.asp**) has a list of GPX data as well as additional resources and utilities.

GeoRSS

GeoRSS (**http://georss.org**) is an extension to the common RSS (Really Simple Syndication) used on web sites to notify readers of new articles or updates. GeoRSS adds geographic coordinates and features to RSS and Atom items.

GeoRSS comes in several flavors: W3C Geo, Simple, and GML. Simple was developed to speed the adoption and use of GeoRSS by providing an uncomplicated format that is sufficient for making points, lines, and polygons. W3C Geo exists based on historical use and is limited to just describing single points. As the need for complex geometry become necessary, GML (Geographic Markup Language) provides the ability to describe complex geographic geometry.

To add GeoRSS Simple to your RSS, you need to first add the GeoRSS namespace reference to your feed definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
        xmlns:content="http://purl.org/rss/1.0/modules/content/"
        xmlns:wfw="http://wellformedweb.org/CommentAPI/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
```

```
      xmlns:georss="http://www.georss.org/georss"
      xmlns:gml="http://www.opengis.net/gml"
>
```

Then, in your RSS entries, you actually add the *georss:point* tag. The most important thing to note is that the latitude and longitude are separated by a single space, and not a comma as is typical in other data formats.

The format of the GeoRSS point tag is:

```
<georss:point>latitude longitude</georss:point>
```

In an actual RSS entry it would look like:

```
<entry>
   <title>M 3.2, Mona Passage</title>
   <link href="http://example.org/2005/09/09/atom01"/>
   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
   <updated>2005-08-17T07:02:32Z</updated>
   <summary>We just had a big one.</summary>
   <georss:point>45.256 -71.92</georss:point>
</entry>
```

As necessary, you can add `radius=""` or `elev=""` attributes to the `georss:` element. Both are assumed to be measured in meters. The radius attribute is useful for giving a scale to the described feature, and elevation is altitude above the WGS84 ellipsoid (typically supplied by a GPS receiver).

In addition to just a geographic definition, GeoRSS uses two optional elements: *featuretypetag* and *relationshiptag* to provide additional metadata to the geographic element.

When noting an element such as a mountain, lake, or residence, the *featuretypetag* is used. The *relationshiptag* relays the geographic coordinates to the actual entity in the RSS item, such as *is-centered-at*:

```
<georss:point>45.256 -71.92</georss:point>
<georss:relationship>is-centered-at</georss:relationship>
<georss:featuretypetag>residential house</<georss:featuretypetag>
```

As mentioned previously, GeoRSS Simple can support more complex geometry such as a line. This may be useful for defining a road or hiking trail:

```
<georss:line>lat1 lon1 lat2 lon2 lat3 lon3</georss:line>

<georss:line>45.256 -110.45 46.46 -109.48 43.84 -109.86</georss:line>
```

A box is useful for defining the extents of an area of interest, or the location of a building:

```
<georss:box>lower_left_lat lower_left_lon upper_right_lat upper_right_lon</georss:box>

<georss:box>42.943 -71.032 43.039 -69.856</georss:box>
```

or a polygon, which can be any number of sides, and is defined by a set of latitude and longitude points describing the outside boundary of the polygon. The last set of latitude, longitude must equal the first set so that the polygon is closed:

```
<georss:polygon>lat1 lon1 lat2 lon2 lat3 lon3 lat1 lon1</georss:polygon>

<georss:polygon>
       45.256 -110.45 46.46 -109.48 43.84 -109.86 45.256 -110.45
</georss:polygon>
```

As the need for more complex geographic geometry arises, it is worth checking out GML (Geography Markup Language) and how to use it in GeoRSS (**http://www.georss.org/gml.html**.

For a list of GeoRSS sources, check out:

- Mapufacture: **http://mapufacture.com**
- PlaceDB: **http://placedb.org**

KML

Another XML format is KML (Keyhole Markup Language) (**http://earth.google.com/kml**), which was developed by Keyhole Technologies and is now owned by Google. Keyhole created the KML file format, Google Earth's predecessor. Since acquiring Keyhole, Google greatly extended the use and capability of KML, which can now define three-dimensional geometry for creating geospecific buildings.

Similar to GeoRSS, KML's major difference is coordinate order, which is longitude, latitude:

```
<Point>
  <coordinates>-90.8694,48.2545</coordinates>
</Point>
```

Like GeoRSS, KML can also define complex geometries:

```
<Polygon id="ID">
  <!-- specific to Polygon -->
  <extrude>0</extrude>                        <!-- boolean -->
  <tessellate>0</tessellate>                  <!-- boolean -->
  <altitudeMode>clampToGround</altitudeMode>
    <!-- kml:altitudeModeEnum: clampToGround, relativeToGround, or absolute -->
  <outerBoundaryIs>
    <LinearRing>
      <coordinates>...</coordinates>          <!-- lon,lat[,alt] -->
    </LinearRing>
  </outerBoundaryIs>
  <innerBoundaryIs>
    <LinearRing>
      <coordinates>...</coordinates>          <!-- lon,lat[,alt] -->
    </LinearRing>
  </innerBoundary>
```

While you would probably never write KML yourself, many tools exist to generate KML from other file formats, or via a drawing program.

One unique feature to KML is the ability to embed 3-D visualization models into the KML file. This is how users insert buildings, objects, and geographic annotations (Sticky Notes for maps) within their files.

The Keyhole BBS (Bulletin Board System) (**http://bbs.keyhole.com**), also known as the Google Earth Community, is an active group of users and developers sharing KML files, how-tos, and ideas, and serves as a clearinghouse for KML information.

Microformats

Microformats are used in web pages to identify common data such as people, places, or events. When browsers show a web page, they are translating the page's Hypertext Markup Language (HTML) for graphical rendering. The actual words or information presented are immaterial to the browser. Microformats add meaning to the HTML by providing a standardized schema applied to the class and ID of HTML attributes, permitting manipulation of this information by other programs.

For example, the hCard Microformat defines how to mark up contact information for a person. Once the information is marked up in a common, meaningful way, other tools can then recognize that information and add the contact to your address book, find her social networking accounts (such as LinkedIn, Plaxo, or even MySpace), send her an email, or instant message her.

For Neogeography there are currently two interesting Microformats: *adr* and *geo*. *adr* is the definition of an address:

```
<div class="adr">
 <div class="street-address">23 Main St.</div>
 <div class="extended-address">Suite 104</div>
 <span class="locality">Northville</span>,
 <span class="region">MI</span>
 <span class="postal-code">48167</span>
 <div class="country-name">U.S.A.</div>
</div>
```

geo defines a geographic coordinate in latitude and longitude:

```
<span class="geo">
 <span class="latitude">42.4266</span>,
 <span class="longitude">-83.4931</span>
</span>
```

Either of these formats can also be embedded in another Microformat in order to give that broader Microformat geographic context. For example, putting an *adr* in an hCal event would tell you where that event was taking place:

```
<span class="vevent">
 <a class="url" href="http://conferences.oreillynet.com/where/">
```

```
 <span class="summary">Where 2.0 Conference</span>:
 <abbr class="dtstart" title="2007-5-29">May 29</abbr>-
 <abbr class="dtend" title="2007-10-31">30</abbr>,
at the <span class="location adr">Fairmont Hotel,
      <span class=”locality”>San Jose</span>,<span class=”region”>CA</span>
      </span>
 </a>
</span>
```

Technorati (**http://technorati.com**) is the primary supporter of Microformats and provides numerous tools and aggregators for them.

For more information on various Microformats, how to use them, and how to help develop new ones, check out Brian Suda's O'Reilly Short Cut "Using Microformats."

Advanced formats

So far, we've just discussed emerging and entry-level technologies and formats. There are many more full-featured, but also more complex, standards such as WMS, WKT, and GML that provide very high-fidelity data representations and can handle complex geometry. They are key components in the underlying geospatial technology, but are beyond this scope of this book. I suggest following up with *Web Mapping Illustrated* by Tyler Mitchell (O'Reilly) for more in-depth information on these advanced concepts.

Right formats for the right job

Each of the formats above have different strengths and weaknesses. They can be used together or separately depending on your application and needs.

GPX is primarily a file format for working with GPS units. It lacks features for better annotation and describing actual geometries. It should be used for archiving and sharing data with devices.

GeoRSS is best used for sharing geographic updates and individual items since it is limited to describing a single RSS item such as a blog post, sensor reading, story, photograph, or the tracking of a moving object. GeoRSS cannot currently handle groups or sets of points or geometries.

KML is very well suited for storing and displaying sets of data such as favorite locations, plot areas, and geospecific buildings.

Microformats should be limited to displayed data that will be read by a human. It doesn't serve as well as a machine-only format, and without an hCard or hEvent, there isn't currently a way to associate an *adr* or *geo* with its contextual information.

Software Tools

Now that you know what types of data are useful, it is important to identify the various tools you'll need to work with the data. Using a text editor is always an option, but only rarely the best solution. Instead, find a good program that supports the various data formats you will be using. If something is missing, try adding it yourself and sharing it back to the neogeography community.

Arguably the most useful tool you'll need in neogeography is GPSBabel (**http://www.gpsbabel.org**). It is a cross-platform application that can convert to and from a large number of geographic data formats. GPSBabel can also upload or download data directly with your GPS unit. This might include support tracks and routes in addition to simple waypoints.

Because GPSBabel is a command-line utility, some users may prefer a graphical user interface (GUI). Users of most platforms can find simple GUIs widely available. In addition to converting between formats, GPSBabel can also apply filters to the data as it is converted. Filters are useful for splitting long tracks into shorter ones where a long time or distance gap exists between points, such as when you have multi-day trips, or take a long break and don't want to store the many GPS points at a single location. Also, you can apply filters to optionally include only waypoints with a specific radius, to smooth tracks, or to remove high-error points from the track log. These filters are useful for finding POI within a specific area, or removing GPS points that have high error due to poor satellite signals.

For easy GPX file management, sorting, and organizing waypoints, Windows users can use Easy GPS (**http://www.easygps.com**), and Mac users should check out MacSimpleGPS (**http://www.macsimplegps.com**).

Currently, most RSS feeds do not contain the GeoRSS extensions, but they may contain useful geographic information. Geonames' RSS-to-GeoRSS converter (**http://www.geonames.org/rss-to-georss-converter.html**) is a great tool for parsing and converting these RSS feeds to GeoRSS ones that can be used in your applications.

Google Earth (**http://earth.google.com**) is a 3-D geographic visualization tool that uses KML for locations, geometry, overlays, and histories. Google Earth also understands network links to remote KML sources, which is very useful for data that is continuously updating, such as weather tracking.

WorldWind (**http://worldwind.arc.nasa.gov/**) is another 3-D geographic tool developed by NASA for scientific visualization and uses the beautiful satellite imagery for their tiles.

These are a few of the tools that will ensure a productive and fun time, with new tools being developed every day. Subscribe to bookmarking feeds such as

http://del.icio.us/tag/geo or http://del.icio.us/inbox/starhill_blend, or join one of the communities mentioned in the resources section at the end of the book to find new sites and tools as the show up.

Hardware Tools

Neogeography does not have specific "tools of the trade" so much as repurposing devices you already use in a geo-centric way. For example, you may want to use a camera to take photographs of places that you visit. An audio recorder can capture the noises or music of an area. Laptops, handheld computers, and mobile phones all assist in locating, tagging, annotating and viewing your creations. But none of these tools were necessarily designed with "geography" as their purpose. Be creative and find new uses for common devices.

GPS Units

GPS units have become prolific and relatively inexpensive. You can purchase a simple USB GPS unit to attach to your laptop for as little as $30 USD.

There are a couple of major factors to consider when choosing a GPS device:

Display
: Do you want to use the unit to display maps, tracks, and information?

Logging
: Some GPS units have built-in logging, so that when you turn the unit on it starts storing your location as tracks.

Bluetooth
: Wireless connection is useful if you want to use the unit with a mobile phone, handheld computer, or just not deal with cables.

Based on these factors, there are four major types of GPS units: Handheld, Vehicle, Logger, and Puck or GPS Mouse. A Handheld GPS unit is a full-featured device with map display, routing, tracking, and a user interface. It can be used independently of any other devices and is the most useful when out in the field. Good Handheld units include the Garmin Geko or Garmin 60CSx

Vehicle GPS units are very similar to handheld units, except that in-vehicle units typically are larger, have car mounts, and assume your position is along roadways. Many cars now come with navigation systems built-in. The benefit of a built-in navigation system is that there are typically additional sensors, such as wheel speed sensors, magnetometers, and accelerometers that aid in navigation when pure GPS location is unavailable, such as in tunnels. However, these built-in systems are also not upgradeable and do not allow for as much modification and uploading of new data. Examples of vehicle GPS units include the Garmin Nüvi and the Magellan RoadMate.

Figure 1: GPS receivers come in different shapes in sizes: GPS logger (top left), handheld GPS (right), vehicle nav system (bottom left)

A Logger GPS is a small, simple device with no connections or mapping display. Typically, you just turn the device on, and it starts storing locations to a track. Therefore, you can toss it into your bag and forget about it. When you have completed your trekking, you can then connect the GPS unit to a computer or other device and download the track. Additionally, many GPS Loggers have Bluetooth, so if you want to display your current location, you can connect a mobile phone via Bluetooth to the GPS and display maps, current position, satellites, and so on. Good loggers include the Wintec WBT-200, NaviGPS, or the Delorme Bluelogger.

A Puck, or GPS Mouse is a simple GPS receiver with a cable for connecting to a computer to actually read and store the positioning data. These are the simplest devices and also don't cost very much. They have also gotten very small and easy to interface with embedded devices, such as microcontrollers or radio controlled vehicles.

Check out the OpenStreetMap GPS reviews page for good discussion on GPS units: **http://wiki.openstreetmap.org/index.php/GPS_Reviews**.

Cameras

There are very few digital cameras that feature actual GPS or geolocation hardware. The Ricoh Pro G3 was the first widely available camera to have an onboard GPS sensor and automatic geotagging of your photographs. The Eye-Fi Eye-Film SD memory card should be available in 2007 and features WiFi connectivity for automatically uploading and geotagging your photos when you are near a wireless network. By embedding the geolocation technology in the common memory card, any camera can gain geolocation ability of photographs.

In general, any camera can be used for Neogeography projects. Later on we will discuss how to automatically and manually add geotags to your photos to locate and map them. You can also use print film that is scanned or any kind of picture.

Other hardware

Neogeographers experiment with a lot of other devices in their projects.

Mobile phones can be used to geolocate people by recording the observed cell tower data. The Boost Mobile and Nextel mobile phones provide an interface for developers to use the geographic location in their applications. Symbian OS mobiles also provide a limited interface to the cell location information.

Handheld computers and tablets such as the Nokia 770 (**http://nokia.com/770**) are being used by neogeographers to display maps using Maemo Mapper (**https://garage.maemo.org/projects/maemo-mapper/**) and GPS. Also, handheld devices can record audio, video, and text for geotagging and uploading or displaying.

Apple and Nike released a Sport Kit (**http://www.apple.com/ipod/nike/**), which is an electronic pedometer that controls your iPod music player based on the distance you have run. Neogeographers are experimenting with using this device for track distance measurement and location augmentation.

Björn Hartmann's "GPS + Google Maps Mash-up in 42 Lines of Code" (**http://regexp.bjoern.org/archives/000186.html**) is a great example of a quick but slick hack for building your own GPS device and using the data.

Finally, geolocation components and technology are continuing to drop in price, size, and power requirements. Therefore, future generations of devices will begin to have GPS built-in as a standard feature.

GeoParsing

Batch geocoding usually relies on us having a fairly well organized list of locations. However, it is more likely we have documents or web sites we want to get the locations from. Perhaps it is a document analyzing various real estate options in the area or a news web site.

GeoParsing is the process of converting a natural language document (free of specific markup) and extracting geographic locations and their contexts. For an excellent example, check out GutenKarte (**http://gutenkarte.org/**), which parses classic literature and generates maps. You can see an automatically generated map of Verne's *Around the World in Eighty Days*, or Tolstoy's *War and Peace*.

MetaCarta Labs provides a GeoParser API. Using this API, you can GeoParse your own documents or web sites and generate maps, images, or feeds.

See: **http://labs.metacarta.com/GeoParser/documentation.html**

Where Are You?

We have now covered some of the general information behind neogeography. The first step, though, is actually figuring out *where* something is. Determining where something is can be referred to as *geolocation*. There are numerous geolocation techniques, some automatic and easy, some convoluted and error-prone. We will discuss some of the techniques below and examples of how to use each.

The following desired data is typical for determining where you are: latitude, longitude, altitude, address, region, country, and named location.

Using GPS

GPS offers by far the highest fidelity means of geolocation. The current U.S. GPS network and upcoming European Galileo satellite network offer sub-meter precision to commercially available devices. A GPS receiver uses triangulation of a minimum of four satellites to locate the user in X, Y, Z, and time.

By using a GPS receiver, you can directly read your current position in latitude, longitude and altitude. As you travel, you can mark waypoints, such as a picnic spot where you took pictures of the mountain range, or the head of a trail. Most GPS devices also can store a time history of locations, resulting in a track.

You can then record these positions from the receiver by hand to a notebook or download them using a software program. Using a notebook offers flexibility and a backup, as notebooks never run out of batteries or get dropped and break. However, as your amount of waypoints grows, or you store tracks, it is much easier to download the data off of your GPS receiver using some software and a cable

(unless the GPS uses a wireless communications like WiFi). Nearly all GPS units come with their own software, or you can use the very versatile GPSBabel program discussed above to download data from your GPS receiver.

Geocoding

While GPS is very accurate and easy to use, we do not always have the benefit of having a receiver with us when we travel or are able to get a good signal at the location, and sometimes we are not even physically at the location we want to mark. A lot of data we will want to use for our projects may simply be named locations such as: “32 Derby Square, Salem, Massachusetts,” “The Eiffel Tower,” or even just “Brazil.”

Geocoding is determining the position of a named location using automatic techniques. It converts “32 Derby Square, Salem, Massachusetts” to the coordinates [42.52111, –70.894885] latitude and longitude using a web service or application. Geocoding is very useful because it can allow you to take a large set of addresses, place names, or a list of places you remember and get more accurate and mappable coordinates.

There are some important considerations when geocoding. Geocoding can vary between different geocoding providers, which may be due to different datasets and different levels of service for different account levels (e.g., free, premium). Additionally, a general location name may have many valid solutions. Therefore, most geocoders return all possible locations and an estimated ranking of accuracy. It will be up to you to choose the most correct geocoded location, perhaps by comparing to other nearby locations or verifying on a map.

Reverse geocoding is converting latitude and longitude into a named location. For example, do you know where N29° 58' 30" x E31° 8' 15' ([29.975, 31.1375]) is? Using a reverse geocoder, you could determine that this is the location of the Great Sphinx of Egypt.

Geocoding with Photos

For a bit of fun, check out Tim Waters' Geocodr. He uses geotagged Flickr images, along with a clustering algorithm, to determine where a location is based on the Flickr tags. So if there are 50 pictures tagged with "Stonehenge," then Geocodr will determine some average centroid of these picture locations and return that as the geocoded location.

By using a constantly updating and changing data source like geotagged photos, Geocodr can provide up-to-date vernacular geocoding. The meaning of *downtown* might change from one month to the next, and traditional data sources take much longer to gather this information.

http://geothings.ning.com/FlickrGeocodrApp.php
http://geothings.ning.com/Flickr/flickrgeocodr.php?place=stonehenge

Basic

There are several easy ways to geocode either a single location or a large number of locations. Geonames (**http://geonames.org**) is a very complete and full-featured geocoder that handles cities, neighborhoods, and geographic features, and can parse general text. Geocoder.us (**http://geocoder.us**) is a simple, single geolocation tool, but only works in the U.S. TravelGIS.com (**http://www.travelgis.com/geocode/**) provides a geocoder for 24 countries

Batch Geocode (**http://www.batchgeocode.com**) is a free web site that allows you to upload a large set of tab-delimited data (say from Microsoft Excel or Outlook/Thunderbird contacts), and it will automatically geocode all of these locations and give you back your locations with latitude and longitude. Additionally, you can save the generated map to a hosted web page or download and save a Google Earth KML file. (See the "Mapping Your Genealogy" project in this book)

GPSVisualizer (**http://www.gpsvisualizer.com/convert**) provides several geocoding tools, including single point, batch geocoding, GPS tracklog conversion, and KML output.

See **http://del.icio.us/tag/geocode** for more links to geocoders and techniques.

Advanced

Filling in an Excel spreadsheet, or geocoding your contacts addresses to create a map is useful. But what if you want to embed a geocoder directly into your web site? For example, you may want to let a user see where he is in relation to where you are, or where your store is located.

The following is a simple web page for embedding a Google Map and an entry location for an address or location. When the user clicks on "geocode," a request is made to Google Geocoder, which then creates an icon automatically for the point:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
      <html xmlns="http://www.w3.org/1999/xhtml">
        <head>
          <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
          <title>Google Maps JavaScript API Example</title>
          <script src="http://maps.google.com/maps?file=api&v=2&key=ADD_YOUR_KEY"
            type="text/javascript"></script>
          <script type="text/javascript">
          //<![CDATA[
         var geocoder;
         var map;
         // On page load, call this function
         function load() {
            // Create new map object
            map = new GMap2(document.getElementById("map"));
               map.setCenter(new GLatLng(0,0), 1);
            // Create new geocoding object
            geocoder = new GClientGeocoder();
            // Retrieve location information, pass it to addToMap()
         }
             function geocode() {
             geocoder.getLatLng(
                          document.getElementById("address").value,
                          function(point) {
                                 if(!point) {
                                        alert("point not found");
                                 } else {
                                // Center the map on this point
                                map.setCenter(point, 13);
                                // Create a marker
                                marker = new GMarker(point);
                                // Add the marker to map
                                map.addOverlay(marker);
                                // Add address information to marker
                                marker.openInfoWindowHtml(address);
                          }});
         }
          //]]>
          </script>
        </head>
        <body onload="load()" onunload="GUnload()">
          <div id="map" style="width: 400px; height: 300px"></div>
             <input type="text" name="address" id="address" value="" width="40"/><a
href="" onclick="geocode(); return false;">geocode</a>
        </body>
      </html>
```

Geolocation

In our geocoder example above, we showed how to have the user specify a location, perhaps where they currently are, in order to get a map of that location. We also demonstrated how you can use a GPS receiver to get your location. Both of these techniques are typically very accurate, but also require having a GPS unit, or knowing and entering your location.

Automatic geolocation is determining where you are based on various other data, such as your IP address, nearby WiFi base station, or cell towers. None of these pieces of data have innate location; however, based on historical data it is possible to associate this data with a location.

Geolocation by IP

An IP address is a unique identifier given to all computers when they connect to the Internet. Typically when you connect your computer to a local network, you are given a local IP address, probably like 192.168.0.103. However, when your network connects via your ISP (Internet Service Provider), whether it is DSL, cable modem, T1, or GSM, then you are assigned a unique IP address.

Since the IP address is unique and you usually connect to the Internet near where you are physically located (such as your office building, coffee shop, or house), the IP can be assumed to be another "address" for where you are located. This is known as GeoIP. GeoIP is useful for tracking where site visitors are coming from, providing local language and cultural interfaces, or automatically showing pertinent data such as nearby shops or resources based on the users' locations.

There are many databases available that associate these IP addresses with a location. This is done using several mechanisms, such as asking the user when they connect to the service for the first time, inferring that you are connecting through a specific ISP, or by knowing IP addresses in the same general numeric range and then assuming that you are also physically in the same vicinity.

The accuracy of GeoIP can vary. If you are connected through a large block of static IPs, for example as part of a business or university, then the accuracy is probably good, because the IP would have been associated with the location and not changed. If you connect through a cable modem, dial-up modem, or GSM cellular modem, then the accuracy is probably very low. This is because IP addresses are given out to users as necessary. You may now have an IP address, sitting in Seattle, Washington, that two hours ago was being used by someone in Spokane, Washington.

Additionally, if a user is connected through a proxy, such as a firewall, or through another remote server, then their apparent location could be very wrong.

Due to the possibly high inaccuracy of GeoIP, it is most often relied on for the country, and possibly the general region. It should rarely be trusted for city or better accuracy, and if used should be assumed to be wrong by some distance and allow the user to correct or ignore the guess.

The largest free database and service for GeoIP is HostIP.info (**http://hostip.info**). If you open the site in your browser, you will immediately see its best guess of your location. You can correct or update this to the database for the future. HostIP also provides a web service you can use for your own programs and sites, or you can download a copy of the database to store and update your own copy.

HostIP.info provides a web service for requesting the location of an IP address. To use the service, pass the IP address and any other options. For example, it will be useful to get the latitude and longitude of the guessed position:

```
http://api.hostip.info/get_html.php?ip=12.215.42.19&position=true
```

The result is a simple text response, which can be parsed using a regular expression:

```
Country: UNITED STATES (US)
City: Sugar Grove, IL
Latitude: 41.7696
Longitude: -88.4588
```

For a more complex response, remove the `get_html.php`. The result is an XML file that uses GML encoding:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<HostipLookupResultSet version="1.0.0"
       xmlns="http://www.hostip.info/api"
       xmlns:gml="http://www.opengis.net/gml"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.hostip.info/api/hostip-1.0.0.xsd">
 <gml:description>This is the Hostip Lookup Service</gml:description>
 <gml:name>hostip</gml:name>
 <gml:boundedBy>
  <gml:Null>inapplicable</gml:Null>
 </gml:boundedBy>
 <gml:featureMember>
  <Hostip>
   <gml:name>Sugar Grove, IL</gml:name>
   <countryName>UNITED STATES</countryName>
   <countryAbbrev>US</countryAbbrev>
   <!-- Co-ordinates are available as lng,lat -->
   <ipLocation>
    <gml:PointProperty>
     <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:coordinates>-88.4588,41.7696</gml:coordinates>
     </gml:Point>
    </gml:PointProperty>
   </ipLocation>
  </Hostip>
 </gml:featureMember>
</HostipLookupResultSet>
```

As mentioned previously, geolocation by IP can be inaccurate due to incomplete data or changed IP addresses. HostIP offers a "rough" estimate by using `rough.php?ip=`. The result is a best estimate, and the result also specifies if it was an estimate:

```
http://api.hostip.info/rough.php?ip=69.19.50.32&position=true

Country: UNITED STATES
Country Code: US
City: Alexandria, VA
Latitude: 38.8147
Longitude: -77.0647
Guessed: true
```

Lastly, HostIP provides a very easy way to embed information about your visitor:

```
<A HREF="http://www.hostip.info">
 <IMG SRC="http://api.hostip.info/flag.php" BORDER="0" ALT="IP Address Lookup">
</A>
```

If you need better accuracy through GeoIP, then there are several commercial databases that promise better accuracy. MaxMind and Plazes both provide GeoIP databases and web services of different accuracy levels depending on the number of uses and cost.

Geolocation by WiFi

Similar to GeoIP, Geolocation by WiFi uses the location-associated Wireless base stations to locate you when you connect to these base stations. This is done by first locating the base station and registering its SSID, the name you see in your wireless connection dialog when you connect.

Whereas Geolocation by IP depends on a single reference for location information, Geolocation by WiFi can use multiple wireless stations and their relative strengths to triangulate location information. However, obstacles, interfering signals, and the moving of base stations or taking them offline can artificially alter the measured strength of WiFi.

For Geolocation by WiFi to work, it usually requires installing device drivers or a small application on the client machine to store and pass all the observed base stations to the server system. WiGLE (**http://www.wigle.net**) is a free and open database of WiFi signals. There are a couple of clients available, JiGLE and DiGLE, that allow for automatic geolocation when running on a client machine.

Loki is a platform developed by SkyHook wireless, who developed a large database of WiFi locations over several years. It now provides an API (**http://developer.skyhookwireless.com**) and device drivers to use WiFi geolocation in programs in Windows XP and Windows Mobile devices.

You can also check out Navizon (http://www.navizon.com), which is another commercial option that allows users to upload their own data to help increase the database, though the database is not available for users to download and use on their own. By contributing data to Navizon, you are then given credits to use in querying Navizon's WiFi geolocation database.

Geolocation by GSM

Lastly, there is the newly popular mobile phone geolocation. Mobile phones simultaneously communicate with several cell towers to provide uninterrupted service as a user travels. As of 2006, all new mobile phones in the US are required to support the e911 protocol that allows mobile service providers (Verizon, T-Mobile, et al.) to locate a mobile phone. However, this information is not typically available to users or developers.

Similar to Geolocation by WiFi, GSM (Groupe Spécial Mobile or Global System for Mobile Communications**)** geolocation measures the relative strength of nearby cell towers, using their known location to interpolate a user's present location. Once again, interfering objects and mobile phone status can alter the measurements. Finally, many mobile phones will preferentially connect to their provider's cell towers, and not necessarily to closer cell towers that belong to other providers. This results in making building of a database of cell tower locations difficult since handsets from different providers must be used to detect all towers in an area.

Some mobile phones provide an interface for the location information or cell tower identifiers. The newest phones on the market actually have an embedded GPS receiver, bypassing the need for triangulation by the protected, and potentially inaccurate, cell towers.

GSM geolocation requires four values from the mobile phone: CID, LAC, MCC, and MNC. These are all parameters that uniquely identify a cell tower:

CID
: Cell Identification is a number that identifies the active cell. The CID is unique to the LA (Local Area).

LAC
: Local Area Code is a regional identifer; several cells are contained in a LA (Local Area).

MCC
: Mobile Country Code is the X.121 code for the country (214 = Spain, 238 = Denmark, etc.)

MNC
: Mobile Network Code is the provider or carrier (T-Mobile, Orange, Verizon, etc.)

GSMLoc (**http://gsmloc.org**) is a project started by Christopher Schmidt to provide a free and open service database for collecting and sharing measured cell locations and areas.

To use GSMLoc, there is a REST web service where you enter the observed cell tower parameters, and get back the bounding box and centroid of the estimated location:

```
http://gsmloc.org/api&nid1=###&nid2=###&&lac=###&&cid=###&format=text
```

where `nid1` is the MCC and `nid2` is the MNC.

You can then get the response in text, XML, or Javascript object notation (JSON).

GSMLoc is very open for users to contribute new data. You need a Symbian Series 60 capable phone, such as the Nokia series. Then install the StumbleStore application (requires Python for Series 60). Start up StumbleStore, connect to a Bluetooth GPS, and then "Start Stumbling." As you wander about, StumbleStore will record the observed cell towers, and GPS location of these signals. When you are done stumbling, you can then upload this information back to GSMLoc.

Mologogo (**http://mologogo.com**) is a small project that works with Nextel phones and various Windows Mobile devices. It lets you upload your location to a central server.

General geolocation

In this section, we covered several mechanisms and tools for geolocation. Primarily, we discussed services and utilities that you could easily integrate into your own projects. There are a lot of other services that do geolocation and provide services built upon geolocation.

However, there are at least several projects that bring all of these techniques together, and provide various services and tools for mixing into your site and applications.

Intel's PlaceLab (**http://placelab.org**) performed extensive research and tool development for many geolocation techniques. The POLS, Privacy Observant Location System (**http://pols.sourceforge.net/**) project spun out and provides tools for various geolocation methods on mobile devices. The PlaceLab project still has numerous papers and research on geolocation technologies and user interfaces.

Plazes (**http://plazes.com**) is a web application that tracks users and locations, and does geolocation by IP, WiFi, cell, and user-defined location. Using their Plazer application, you can easily set new locations and find other people nearby. The Plazer even integrates with Skype to set your location in your status message.

A great part about Plazes is its public API (**http://beta.plazes.com/api/**) that is free for non-commercial use and lets you do all kinds of things with the location data, photos, tracks, and so on.

Making Some Maps

Using the techniques and tools covered previously, we can gather locations appropriate to your project. The following information will help create maps of this data.

Depending on your goal and audience, there are several ways to produce a map. There are many excellent free, open source, or commercial programs that allow you to load your own data, generate a web page, or even print your maps for offline use. You can also use web sites to generate code for embedding maps and location interfaces into your own web sites or generate an entire site. Lastly we will talk about the lower-level tools you can use to program your own maps.

Mapmaking Web Sites

Many web sites provide a way for users to easily make and maintain maps. Some sites merely collect locations and share the map on their service. Other sites help embed the maps in your own site.

Platial has gained attention as a leading social mapping site. Its primary purpose is providing a location where users can create sets of locations and share them with other users. Viewers can then check out these user-suggested locations, such as Parks in Washington, D.C. (**http://platial.com/search?what=park|parks|outdoor|patio|hiking|biking|hike|bike|trail|trails&where=Washington,%20DC**)

It is easy to create a map using Platial. After establishing an account, click "Make a Map" and provide a title, description, and optional icon. On the Map Creation page you can add locations. To get close to your location, enter an address, and then click on the map to add a marker and press “Next.” Fill out the information on the story about the location. Keep going to add all your locations.

[this step-by-step seems overly detailed compared to your other descriptions. I would simplify the directions.]

When finished, view your map or email it to others. Using Platial's MapKit (**http://platial.com/mapkit**) you can embed this new map in your own web page. Click on the MapKit link, fill in the URL, and then click "Make my MapKit."

On the next page, you will see a textbox with HTML and JavaScript code required to embed the new Platial map in your web site. You can further customize your map markers, name, size, starting location, and tags. Your Platial map has GeoRSS output that can be used in an aggregator or viewer like Mapufacture, and KML output for viewing in Google Earth.

Other sites that allow users to build static maps include:

- MapBuilder: **http://www.mapbuilder.net/**
- GMapEZ: **http://bluweb.com/us/chouser/gmapez/**
- QuickMaps.com

For more sites and utilities, check out GoogleMapsMania (**http://googlemapsmania.blogspot.com/**) and *geo* tagged sites on del.icio.us (**http://del.icio.us/tag/geo**).

Build a map site in Ning

Using Platial lets you create a static map for embedding into your web site. You can add your own markers and annotations and have these show up on the map. Although the map is viewable, it provides one-way communication, and does not allow users to add or edit locations. For a more dynamic dataset and web application functionality, there are other applications.

Ning is a web application service that provides users with a sandbox to create their own social web applications. You can start from a template application, start with a copy of an existing Ning application, or completely write your own application in the Ning framework.

For example, in Ning it is trivial to make photo sharing sites, user groups, blogs, or videos. You can then add geographic location to these applications using Ning's Mapping API (**http://mapmap.ning.com/**) to drop in a Google Map. Then you could create a map of your photo locations, your user's favorite hangout spots, or locations mentioned in your favorite television show.

To start, browse around Ning's site and find an application you like, such as a photo mapping application:

1. Navigate to **http://photos.ning.com/**
2. Click "Get your own photos site" in the top bar
3. Choose a Name and a web address

4. Customize your application colors and privacy, and invite people to start using it.

To see all of the Ning Map applications and grab a copy for your own use, see Ning's Map Mash-Ups listing. (**http://categories.ning.com/Map_Mash-Ups**) Ning also provides a site for getting photos of an area like on Yahoo! Maps. (**http://photos.ning.com/index.php/main/photo/map**)

Programming Your Own Maps

Feeling a little more ambitious and want to create your own maps? Many tools and platforms make it straightforward to program your own maps. In this section we will discuss some of the libraries and tools you can use to quickly and easily embed maps and location into your site or application. These libraries require some programming, typically in JavaScript for web sites.

For most web site mapping libraries there is a three-step process:

5. Include the map library's JavaScript file in your header
6. Create an HTML element that will place and size the map
7. Include the JavaScript to call the map's creation function, and add markers, controls, zoom, map type, lines, or style

Each mapping library has slightly different syntax and different features, but this three-step process is a pretty good basis for starting with any of the libraries and then expanding from there.

Mapstraction

Mapstraction (**http://mapstraction.com**) is a JavaScript library that provides a common interface for the various mapping providers. This permits you to create your map and then switch between Google, Yahoo!, Microsoft, MultiMap, or other provider's maps. You can even dynamically switch the map without reloading the page.

To use Mapstraction, you will need to include the headers, and necessary keys of the desired mapping provider's JavaScript definitions in the head of your HTML. The keys are available from the mapping library's home page (listed in the

Mapping Libraries resources section), depending on the terms of service and licensing. Put the following script definitions in the `<head>` section of your web page:

For Google Maps:

```
<script type="text/javascript"
        src="http://maps.google.com/maps?file=api&v=2&key=GOOGLE_KEY"></script>
```

For Yahoo! Maps:

```
<script type="text/javascript"
```

```
      src="http://api.maps.yahoo.com/ajaxymap?v=3.4&amp;appid=YAHOO_APPID"></script>
```

For Microsoft VirtualEarth:

```
      <script type="text/javascript"
            src="http://dev.virtualearth.net/mapcontrol/v3/mapcontrol.js"></script>
```

For MultiMap:

```
      <script type="text/javascript"
            src="http://clients.multimap.com/API/maps/1.1/CLIENTNAME"></script>
```

The required Mapstraction Javascript:

```
      <script src="/mapstraction.js" type="text/javascript"></script>
```

Then, you need to define the location and size of the map, and center it:

```
      <div id="simplemap" style="width: 500px; height: 300px"></div>
      <script type="text/javascript">
            var mapstraction = new Mapstraction('simplemap','google');
            var firstPoint = new LatLonPoint(43.671844983221604,-79.38823699951172);
            mapstraction.setCenterAndZoom(firstPoint, 15);
            mapstraction.addControls(pan:true, zoom: 'large',
                  overview: true, scale:true, map_type:true);
      </script>
      <a href="#" onclick="mapstraction.swap('google'); return false;">google</a>
      <a href="#" onclick="mapstraction.swap('yahoo'); return false;">yahoo</a>
      <a href="#" onclick="mapstraction.swap('microsoft'); return
false;">microsoft</a>
```

See the Mapstraction API for more documentation and examples on how to use the library.

Mike Williams (**http://www.econym.demon.co.uk/googlemaps/**) has a great set of advanced tutorials on Google Maps, and check out the O'Reilly book *Google Maps Hacks* by Rich Gibson and Schuyler Erle for more ideas and information on mapping libraries.

Proxies

Most browsers have a built-in security measure to protect against possible attacks by only allowing your browser to connect to one web site at a time. Thus, a site can't make a JavaScript request to another web site. Therefore, all data must come from the web site you are currently visiting. This isn't always what you want, especially for a mash-up that pulls data from several sources into a single view.

To overcome this problem, the web site needs to provide a local proxy for your browser to request the data from. The proxy routes the remote service through the local server and to your browser. You can download a simple proxy from the book's site: **http://mapsomething.com/demo/proxy.html**.

However, you should be careful when setting up a proxy, as malcontents can use a publicly visible proxy as a gateway for their nefarious purposes, and it would all be traceable to you. Therefore, you can either set up Apache to do your proxying, or hardcode the remote service URL into your proxy.

For more information on proxying, and other options on securing your proxy, visit: **http://developer.yahoo.com/javascript/howto-proxy.html**

OpenLayers

While Mapstraction is a mapping abstraction library that provides a simple interface to a large number of mapping APIs, the OpenLayers project is pushing the envelope with new and advanced functionality for dynamic maps.

OpenLayers (**http://openlayers.org**) is a free and open source JavaScript mapping library that provides developers with a large toolkit not found in other modern libraries. It was spun out from MetaCarta labs and powers a number of their other open projects, such as GutenKarte (**http://gutenkarte.org**).

Some of the advanced features include:

- Multiple layers that can each be turned off or on
- WMS tiles from map servers
- Layers for Google, Virtual Earth, Yahoo!, and MultiMap
- Bounding-box zoom

Using OpenLayers is very similar to using the other mapping APIs: you include your header definition, define an HTML element to use for the map, define the map in JavaScript, and then add layers, controls, and markers.

```
<html>
<head>
<script src="http://openlayers.org/api/2.2/OpenLayers.js"></script>
</head>
```

```
<body>
  <div style="width:100%; height:100%" id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');
    var wms = new OpenLayers.Layer.WMS( "OpenLayers WMS",
              "http://labs.metacarta.com/wms/vmap0", {layers: 'basic'} );
    map.addLayer(wms);
    map.zoomToMaxExtent();
  </script>
</body>
</html>
```

A very advanced and useful feature of OpenLayers is that it can consume geographic feeds such as GeoRSS and automatically create markers and information bubbles. The GeoRSS feed is added as an additional layer and can therefore be turned on and off in the map. To create a map of weather conditions, put the following JavaScript in your page:

```
var weather = new OpenLayers.Layer.GeoRSS( "Weather",
    "proxy.pl?http://dev.mapufacture.com/feed/rss/keyword/weather", {layers:
'basic'} );
map.addLayer(weather);
```

Notice the prefix `proxy.pl?`. This is required to access files from the server. See the sidebar on proxies for more information on how to set up and properly use a local proxy.

Because OpenLayers supports freely available mapping tiles from WMS servers, it is an excellent tool for web applications that display unique or proprietary datasets and locations. In addition, the development team is quickly adding new features and support.

See WMS-Sites (**http://wms-sites.com**) for a growing list of WMS services available and areas of interest.

worldKit

Mikel Maron's worldKit (**http://worldkit.org**) is a free and open source Flash-based map that supports a variety of formats for easily mapping locations and feeds. It is different from the above mapping APIs that rely on JavaScript and an external source of mapping tiles and images. Using Flash for mapping provides better cross-browser support since functionality is provided by a standardized plug-in. Flash is also good for mobile applications.

Configuration of worldKit includes placing the SWF (Shockwave Flash) file and an XML configuration file on your web site.

config.xml:

```
<?xml version="1.0" ?>
<worldkitconf>
  <width>500</width>
  <height>250</height>
```

```
    <displaytype>daynight</displaytype>
    <dayimg>day.jpg</dayimg>
    <nightimg>night.jpg</nightimg>
    <dataurl>rss.xml</dataurl>
    <update>60</update>
    <showonlynew>false</showonlynew>
  </worldkitconf>
```

The "dataurl" can be a local *rss.xml* file, or a remote file by using a proxy mentioned above. worldKit uses GeoRSS and plain RSS with geocoding to get the locations of the data points.

And then add the following code in your HTML file:

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=
7,0,0,0" WIDTH="800" HEIGHT="400" id="worldkit">
<param NAME=”movie” VALUE="worldkit.swf">
<param NAME=”quality” VALUE="high">
<param NAME=”bgcolor” VALUE="#000000">
<embed src="worldkit.swf" quality="high" bgcolor="#000000" WIDTH="800"
HEIGHT="400" NAME="worldkit" ALIGN="" TYPE="application/x-shockwave-flash"
swLiveConnect="true" PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
</embed>
```

That is all there is to it. Check out the worldKit documentation for more information on configuration options, and integration into various weblogs and engines.

OpenStreetMap

OpenStreetMap (**http://openstreetmap.org**) is an ambitious project to create a free and open geographic database of the world using user-collected data.

Users travel with their GPS and cameras, recording streets, parks, paths, trails, and other features, and then upload their GPX files to the central OSM servers. Then, using any of several client applications, they can annotate the GPX track data with actual streets, names, traffic patterns, and so on. After that, all of the data is aggregated together to provide the world with a database of information.

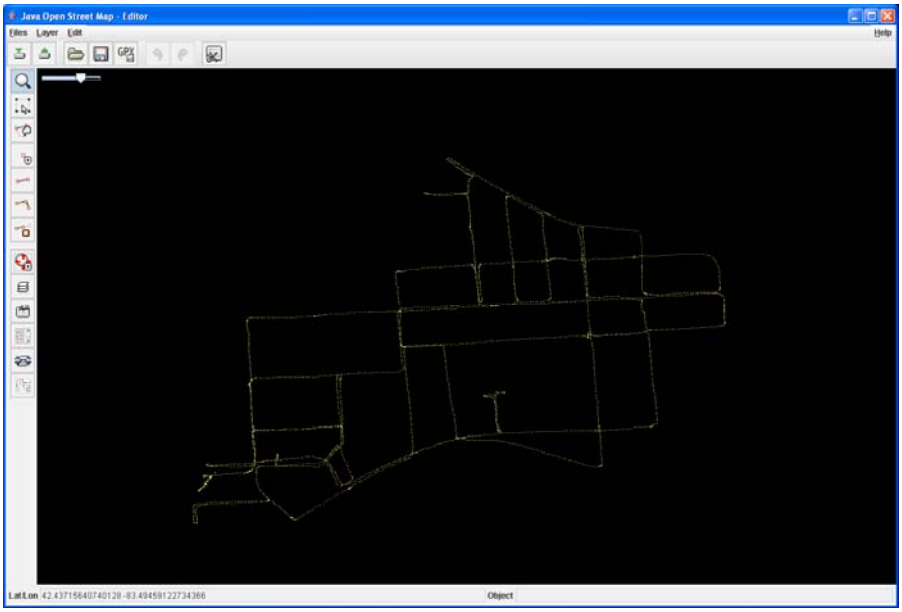


Figure 2: OpenStreetMaps JOSM editor makes it easy to mark up and annotate GPS tracklogs into street data

The gathered data is licensed under the Creative Commons Attribution–Share Alike 2.0 License (CC-By-SA) (**http://creativecommons.org/licenses/by-sa/2.0/**), which allows for free use in your applications. Companies such as Nestoria (**http://www.nestoria.co.uk**) have started using OSM data for their local real estate mapping in parts of the UK.

It is also possible to use the OpenStreetMap tools and data to create your own printed maps for use when you are away from the digital world.

The project started in 2004, and in that time it has gathered millions of data points from countries all over the world. OpenStreetMappers often hold "mapping parties" in various locations where for a day or two large groups run around with GPS receivers to thoroughly map an area. You are definitely encouraged to check out the project and contribute, and see about setting up your own mapping party.

Adding Location to Your Web Site

In the previous section you learned how to build some maps. You probably now have a neat map, with pop-up bubbles and all describing some of your favorite locations. However, just placing a map into a site can make it feel like an add-on, rather than an integral part of your web site.

It is more useful to integrate location and maps into your web site, dynamically adding locations based on the content of a blog post, the location of visitors to the site, or based on aggregated data from other sources.

In this section we will present some of the tools and techniques for integrating neogeography into your site. Once you add location to your site, users can then find your web site based on geographic location (e.g., a web site for a store). Services like GeoURL (**http://geourl.org**) **and** A2B (**http://a2b.cc**) maintain a directory of web sites based on their specified location.

Also, GeoTagThings (**http://geotagthings.com**) is another site that stores user-specified locations of web sites. Using a browser bookmarklet, you can quickly associate any web page with a location. A *bookmarklet* is a small JavaScript bookmark that you can add to your browser's bookmark list or bar and then click to activate some functionality. In this case, you will be taken to a form to locate the page on GeoTagThings.

Location of a Page

The first thing you may want to do is to mark the location of an entire page. The `<meta>` tags in the HTML area can be used for specifying the location.

Unfortunately, as is typical with emerging standards, there are several different formats that can be used for specifying the location of a page.

The ICBM format was defined by Joshua Schachter for GeoURL and further promoted by Matt Croydon (**http://postneo.com/icbm/**). It is useful for specifying the exact coordinates of a page:

```
<meta    name="ICBM" content="42.4266, -83.49307" />
```

Another means of embedding geographic metadata is through geo-structure tags. These geo-structure tags can include latitude and longitude information as well as regional information and an extra placename. The placename could contain the specific address of the person or business. Or, it could be useful for providing a location that may not have a specific point but covering a broader region, such as a city or district. The following example is for the Museo Nacional Del Prado, in Madrid, Spain:

```
<meta name="geo.position" content="42.4266;-83.49307"/>
<meta name="geo.region" content="US-MI">
<meta name="geo.placename" content="Northville">
```

After you have included the meta-markup in your page, register your site with GeoURL and A2B so that other users can find your page or location through their search engines. GeoURL publishes W3C geo RSS feeds of sites, and A2B provides a web service API for querying their database (**http://www.a2b.cc/software-development.a2b**).

Markup places

Location is more than a map. While latitude and longitude offer fairly exact positioning, they do not offer the average user an actual understanding of "where" that location is. Is it in your town or state? Near where you are going on location?

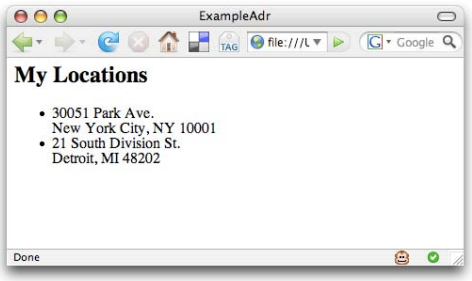
So, in addition to adding maps to your site or application, it is still important to include the address or location of your reference places. Furthermore, there are several easy methods to mark up this information so that other utilities and applications can read and manipulate this geoinformation.

In the section on Microformats we talked about how they allow you to embed location information in the text of your web site. Addresses or coordinates can be included in an hCard or hCal reference, giving context to the location.

Users can use this in their site for a list of store locations, where they want customers to easily get directions, or for a list of photographs.

A simple example of including a list of addresses in your site would look like:

```
<ul>
<li class="adr">
       <div class="street-address">30051 Park
Ave.</div>
       <span class="locality">New York
City</span>,
       <span class="region">NY</span>
       <span class="postal-code">10001</span>
</li>
<li class="adr">
       <div class="street-address">21 South
Division St.</div>
       <span class="locality">Detroit</span>,
       <span class="region">MI</span>
       <span class="postal-code">48202</span>
</li>
</ul>
```



Once you mark up the locations in your site, there are tools that can then use this information for additional functionality. For example, GreaseRoute (**http://code.highearthorbit.com/greaseroute/**) is a Firefox GreaseMonkey script that automatically detects Address (adr) and Geographic Coordinates (geo) Microformation. It then provides viewers with a link to get a map, or directions to the location. Additionally, GreaseRoute's embedded version uses Geolocation By IP to determine where the viewer is and give them directions without requiring them to enter their current address.

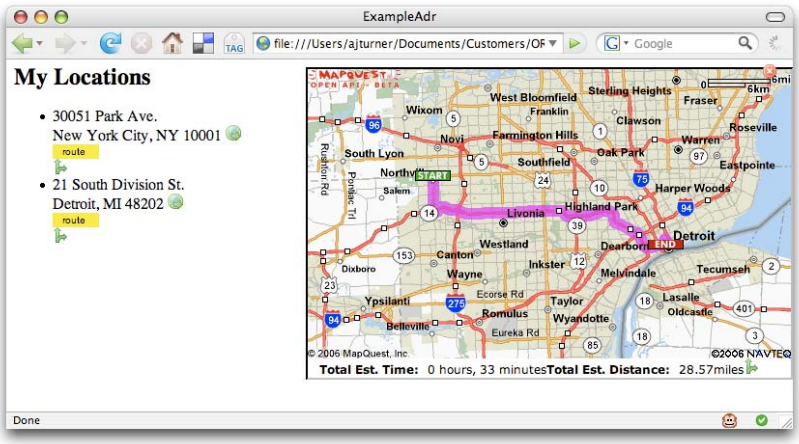


Figure 3: Browser extensions like GreaseRoute can detect addresses or coordinates in a web page and display additional information such as a map or driving directions

Syndicating locations

While it is possible to directly write about locations, and use Microformats and JavaScript to build up your travel log or document various locations, there are tools that make the entire process much easier.

GeoPress (**http://georss.org/geopress**) is a plug-in for the popular WordPress (**http://wordpress.org**) blogging engine that adds the ability to quickly add location, Microformats markup, dynamic Google-Yahoo-Microsoft maps using Mapstraction, and GeoRSS output for syndication.

To start using GeoPress, install WordPress, or choose an existing WordPress blog that you want to add location information to. Then download and copy the GeoPress folder to your *wp-content/plugins* directory. In the Admin interface of your WordPress blog, go to Plugins, and Active GeoPress. You will then need to get a Google Maps API key and a Yahoo! App ID.

Now go to the Write page, and underneath the Post writing section, you will see an area for entering a location and a map. You can enter an address or city and press Enter or Geocode. The map will center to the geocoded location. If you want to save this location for later use, you can give it a name (such as “Home,” “Vacation House,” or “Best Coffee Shop”), and then write your post.

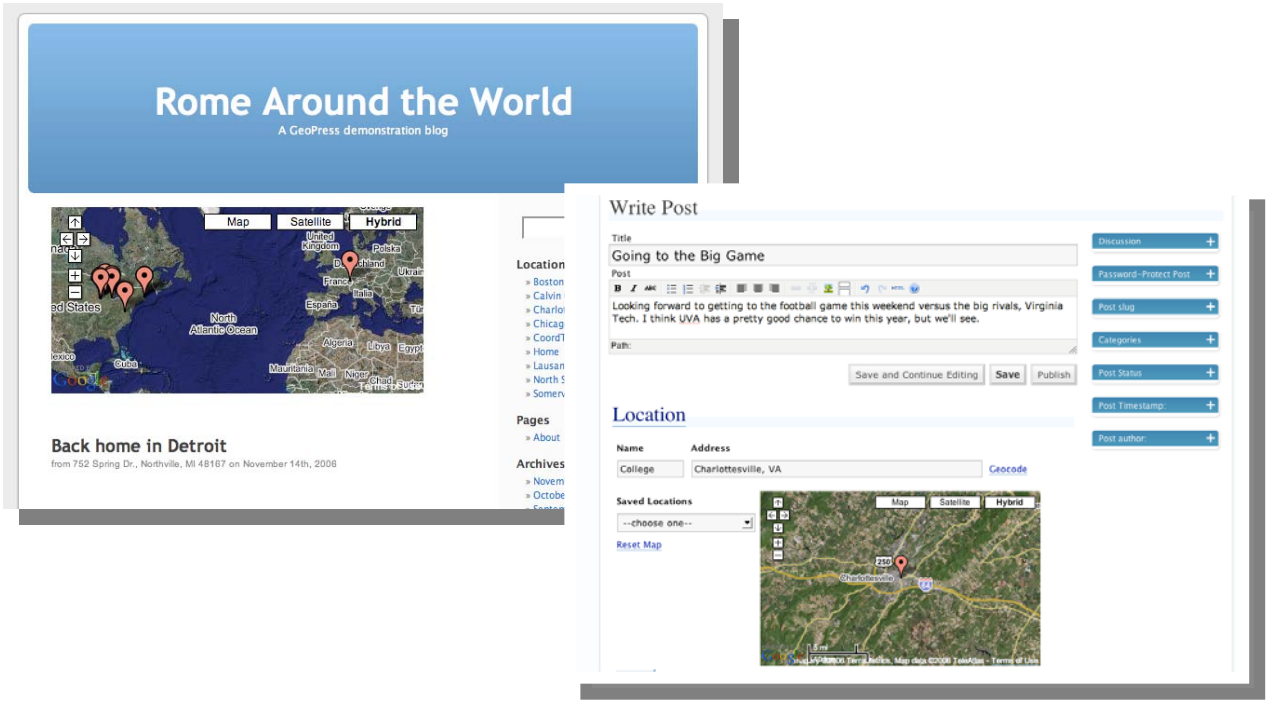


Figure 4: GeoPress adds dynamic maps and Microformat markup to the WordPress blogging engine

When you first start using GeoPress, mapped locations will be automatically inserted into posts. You can optionally turn this feature off and manually enter maps into posts.

To insert a map into the post, just write `INSERT_MAP` anywhere in the post body. This will insert a map based on the default settings you can configure in the GeoPress->Map tab. Alternatively, you can use `INSERT_MAP(height, width)` to set a specific map size.

You can also use `INSERT_ADDRESS` or `INSERT_COORDS` to put the adr or geo Microformat, respectively, in the page.

GeoPress is also particularly useful for automatically adding the GeoRSS information to your RSS and ATOM feeds. Therefore, all of your subscribed readers can now get this geographic information. In the next section we'll discuss how this all works together.

There are similar plug-ins for other blogging engines.

Geo Location for MovableType:

http://locblog.sourceforge.net/geoLocation/

pnh_mf for Textpattern by Chris Casciano:

http://placenamehere.com/TXP/pnh_mf/

A blog is just one way to generate and syndicate geographic data. Blog engines are actually quite flexible, and can be configured to not act like a typical "blog" but just as a simple Content Management System (CMS).

However, there are more full-featured CMSs that offer geographic support. Wikipedia offers a list of GeoCMS (**http://en.wikipedia.org/wiki/GeoCMS**) packages.

Where in the Wiki?

Wikipedia itself has embraced mapping through various projects and now has better built-in support. mySociety's Placeopedia (**http://placeopedia.com**) allows users to attach location to any Wikipedia article. Another project, Wikimapia (**http://wikimapia.org**), creates new entries for using a Wiki and maps.

Several popular CMS platforms have added support for importing and exporting geographic data. Most notably, the Midgard (**http://www.midgard-project.org**) and Drupal (**http://drupal.org**) CMSes have added full GeoRSS support via *modules/plugins*. Therefore, you could make pages for any location or trip, embed maps, syndicate, aggregate, annotate, and share, all in the same application.

GeoStack

There are a lot of technologies and tools that have been presented so far. Individually, these tools provide some nice features for gathering or showing location data. You now know how to mix some of the tools together, like GPSBabel and Google Earth to get data from your device into the 3-D global viewer.

Neogeography is working to bring all these technologies together into a "GeoStack." The GeoStack is a collection of tools and mechanisms that together cover all parts of collecting, gathering, and sharing location information. It enables using a GPS system to capture a waypoint and eventually have other users around the world view and comment on that waypoint.

The GeoStack can be divided into the various steps of data management. Along these steps, there are numerous paths the data can come from or take, depending on the actual application (generating maps, 3-D visualization, adding to my GPS unit POIs). These steps are laid out and illustrated in Figure 5 below:

Capture
: GPS unit, camera, WiFi/Cell/IP logger, notebook of locations

Produce
: The blog, Wiki, sites, databases that contain and generate geographic data

Communicate

A standardized mechanism for generating and transmitting the data

Aggregate

Tools for gathering, storing, filtering, and redistributing the original data

Consume

A viewer, map, or reader that a user uses to view the information; also can upload to GPS to use as direction waypoints or POIs

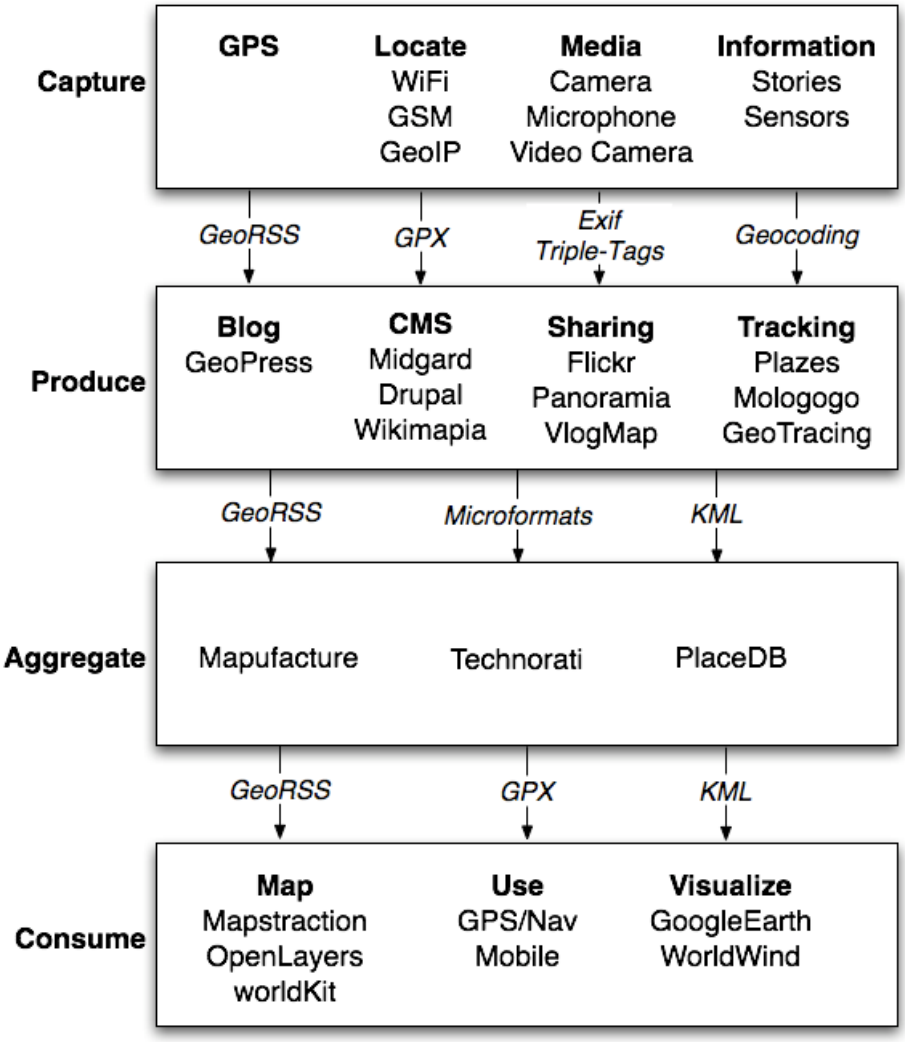


Figure 5: The GeoStack encompasses the entire life cycle of geospatial data, from capture to consume using a variety of tools, formats, and applications

The emerging GeoStack is very exciting. Data is not limited to proprietary formats or locked into a single application. You can gather your tracks and waypoints on GPS, share them in your blog, aggregate them to other users who can then download them and add them to their own GPS units.

Additionally, there are lots of sites showing up with rich sets of data. Just check out Google Maps Mania (**http://googlemapsmania.blogspot.com/**) to see the daily number of new mash-ups that display and generate geographic data.

Platial, Ning, Flickr, and Yahoo! all publish their data in GeoRSS format. Aggregators like Mapufacture make these feeds searchable and locatable and will generate new feeds, such as "kayaking in Colorado" for all the photos, rapids, weather reports, and other information on the sport in the area. You could then save all this data back to your GPS receiver for showing when you are braving the waters.

Future applications and sites should consider how they fit into the GeoStack. By doing so, they enable users to quickly and easily bring in and reuse their geographic data. In the next section we will discuss some projects that use the GeoStack for useful and interesting applications.

Licensing

We have talked a lot about data and mapping providers, but we only briefly touched upon things like terms of service. However, the issue is a very large and important one.

Geographic information such as POIs, maps, routes, and various other data constitute a huge market. There are many companies and clearinghouses that build their entire business on owning and controlling data. They also spend large amounts of money to gather, organize, and distribute this data.

In addition to commercial sources of data, users and neogeographers are constantly collecting and sharing data. Waypoints, hiking paths, photographs, and so on are media that have terms of use associated with it.

Without data, neogeography becomes rather empty. What is a map without any imagery or roads? It is important to understand and comply with the licensing terms that come with data. Sites such as Flickr provide users with an easy mechanism for specifying if their photos are public/private and released under restrictive licenses, or Creative Commons licensing (**http://creativecommons.org/**). However, the distinction is not always so easy to make and understand.

The United States has a good history of sharing geographic data that is gathered by the government. The TIGER/Line data (**http://www.census.gov/geo/www/tiger/**) (Topologically Integrated Geographic Encoding and Referencing system) is all of the census and street data gathered during the U.S. national census. The data is released under very open terms. However, the rest of the world does not enjoy this much free data.

Groups such as the INSPIRE Initiative (**http://inspire.jrc.it**) are working to harmonize data access among nation states in the European Union. Efforts include advocating more open access to data by the public. By comparison, the OpenStreetMap project is working to generate new data that is released under a CC-By-SA License (Creative Commons By Attribution – Share Alike).

In the future, efforts such as the OGC's (Open Geospatial Consortium) (**http://opengeospatial.org**) geographic digital rights working group (**http://www.opengeospatial.org/projects/groups/geodrmwg**) will develop standards for supporting and complying with geographic data licensing. In the meantime, make sure you understand and follow the licensing terms of the data you are using and producing.

Neogeography Projects

That covers most of the tools, techniques, and terms you will need to start making some really cool sites, mash-ups, programs, or whatever. Now we will actually start pulling all these pieces together to make some example projects.

We are going to cover the following projects:

- Locating your photographs
- Directions to your locations
- Tracking your sports
- Mapping your genealogy

However, this is just a starting point to give you some ideas and show you how to bring all these techniques together to a useful end. There are a lot of other projects just waiting for you to start. Perhaps you should make a map of alumni, teammates, club members or forum members, or a carpooling application for where people live and are traveling to.

There are also a lot of possibilities for neogeography games. Every year, the Come Out & Play Festival (**http://www.comeoutandplay.org**) is held in New York City and features dozens of innovative and fun geolocated games.

Locate Your Photographs

You take a wonderful trip, perhaps a cruise in the Bahamas, backpacking across Europe, driving around Asia, or tramping in New Zealand. You have hundreds or thousands of photographs of people, places, and experiences. People ask you "Where was this taken?" or "What was it like in Seville?"

By geotagging your photographs, you can embed the actual location information into the photo itself and map it. Then you can show your trek by train across Germany, or all the photos you took in Hong Kong.

Smart neogeographers will carry a small GPS logger with them as they travel. Then by synchronizing photograph times with GPS tracks, it is possible to infer the location of the photos. Note, for this to work well, you should increase the logging rate of your GPS unit. A good number is probably every two seconds, depending on the type of activity you are doing (the faster you move, the more frequently you should log your position). When you merge together GPS tracks and photographs, where they do not match up, usually some type of interpolation is performed to estimate the position of the photograph between the nearest two known locations and times.

To help calibrate your camera and GPS unit, you should make sure to set the appropriate time zone and time on your camera using your GPS unit. Additionally, it helps to take a photograph of a known landmark before you start your trek. You can then come back and use this known landmark and location to correlate your GPS tracks with your camera clock.

Digital photographs store their metadata in EXIF (Exchangeable Image File Format) (**http://en.wikipedia.org/wiki/Exif**). This usually includes camera make/model, shutter speed, aperture, and date/time. However, there is also the option of adding location name, region, country, latitude, longitude, and altitude, all that is necessary to geotag your photograph.

As mentioned previously, some cameras, and some media storage devices are starting to be sold that automatically tag photographs with the location of the photo. The tools and techniques below can either use your GPS or embedded location information, or provide you with ways to geotag your photographs after you get home.

One common problem with these tools and geotagging techniques is that they assume the entered location or GPS track is where the photo was taken. However, its more meaningful to geotag the photograph with the *subject*. For example, a photograph of Mount Rainier should be geotagged with the location of Mount Rainier, not Seattle. Some tools also provide a means to specify the heading that the picture was taken at.

Basic

There are several desktop applications that provide an easy interface for associating photographs with locations. Typically, these allow you to select a set of photographs and load a GPS track if available. Otherwise, you can manually enter locations.

For Windows users, there is RoboGEO (http://www.robogeo.com), which costs $35–$80, depending on if you will be using the software for commercial purposes.

Alternatively, Microsoft distributes a free program WWMX (http://wwmx.org) and Location Stamper that will synchronize GPX track logs and photographs and modify the EXIF data.

Google's Picasa (http://picasa.google.com) is a photo-management application for Windows and Linux that also supports geotagging photos using Google Earth and Google Maps.

Mac OS X users can turn to GPSPhotoLinker (http://oregonstate.edu/~earlyj/gpsphotolinker/) to load tracks, load photos, and have the geographic coordinates and location embedded into the photo EXIF.

Flickr (http://flickr.com), the photo-sharing site, supports geotagged photos. To use geotagged photos in Flickr you will first need to set a few configuration options. You will probably want to change your Geo Privacy settings (http://flickr.com/account/geo/privacy/) to allow at least your friends, if not other visitors, to see the locations of your photos.

If you are uploading photos with geographic coordinates in the EXIF data, then go to your account settings, and select "automatically geotag photos" in the Geo Import section (http://flickr.com/account/geo/exif), and turn on automatic geo importing. By turning on this option, Flickr will read the EXIF geotags you add to your photos for your map.

If you didn't use one of the programs above, or you want to map photos that you've already uploaded to Flickr, then there are several ways to do this. The easiest is to use the built-in mapping system. Select the Organize menu item, and then select the Map tab. Drag photos from the organizer bar in the bottom to the appropriate location on the map.

The other option you have is to directly tag the photos with the location information. If you select a photo you want to geolocate, then double-click the photo. Select the "location" tab, and then add in the latitude and longitude values.

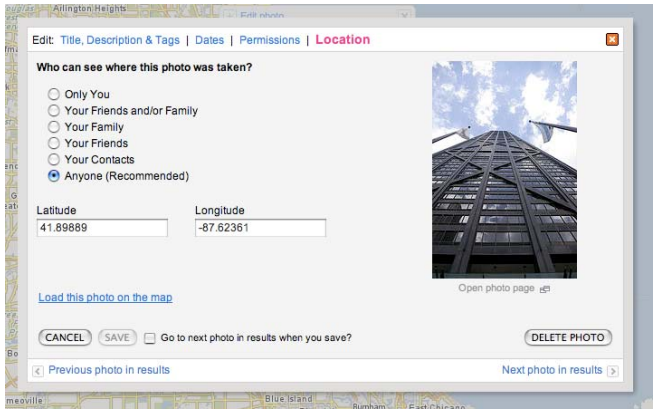


Figure 6: Flickr allows you to directly enter geographic coordinates for photographs

These coordinates can come from a GPS receiver if you carried one with you, or you can use another geocoder to convert the location name to latitude and longitude. For example, MultiMap (**http://www.multimap.com**) provides coordinates for locations all over the world. Geonames (**http://www.geonames.org**) also provides a good geocoder and is especially good at providing coordinates for location names like "Hancock Tower" in Chicago, Illinois [41.89889, -87.62361].

You can directly add these geotags by adding the tags to any photo: `geotagged`, `geo:lat=`, and `geo:lon=`, with the appropriate latitude and longitude entered into the lat and lon tags respectively. Copy the latitude and longitude and then put them into the `geo:` tags. If you have a lot of photos at a single location, you can use the batch organize to quickly apply the location tags to all of the photos at one time.

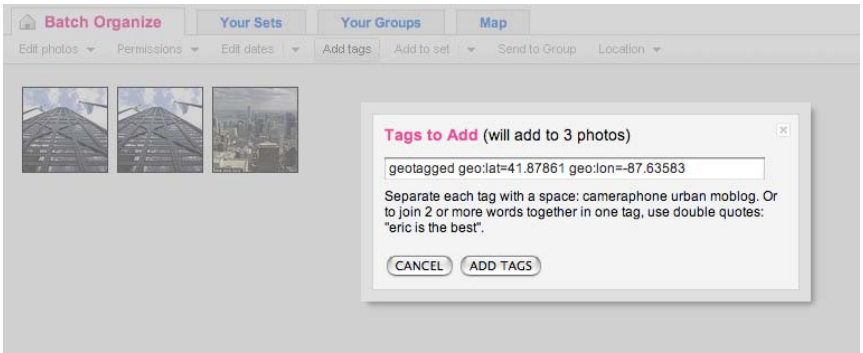


Figure 7: Triple-tagging photos is easy for batch geotagging photos

Lastly, keep on the lookout for more tools showing up that make geotagging easier. For example, Localize Bookmarklet (**http://labs.sumaato.net/tools/flickr_geocode_bookmarklet/**) is a Firefox bookmarklet that lets you geotag your photos directly from the normal view page of the photo.

Now that your photos are geotagged, you will want to see the resulting map. Under the You menu item, select Your Map. It will also have the URL **http://flickr.com/photos/yourusername/map/**.

Additionally, you can get a feed of your photos. At the bottom of the Flickr page, look for the Feed icon and link. This will generate an RSS feed, something like:

```
http://api.flickr.com/services/feeds/photos_public.gne?id=51648834@N00&format=rss_200
```

This feed is just a vanilla RSS feed and doesn't contain the geo information. To get GeoRSS output, add the following parameter to the URL: `&georss=1`. You will now see a GeoRSS feed of photos. Photos without location do not have the `<georss:point>` tag. You can get specifically tagged photos by adding `&tags=tagname`. This is particularly useful for grabbing your geotagged photos if you have added "geotagged" to the photos and then use `&tags=geotagged` in your feed.

SmugMug (**http://smugmug.com/**), Panoramio (**http://www.panoramio.com**), Mappr (**http://www.mappr.com**), and Zooomr (**http://zooomr.com**) are more photo-sharing sites that support mapping and automatically reading GPS EXIF data. Some output Google Earth KML feeds so you can view your photos in Google Earth.

Check out Mark Pilgrim's TripperMap (**http://www.trippermap.com**), a web application that makes a Flash map of your Flickr photos, and even a travel map from the locations and times of your photos.

Intermediate

Flickr and other photo-sharing sites with mapping are very convenient, and also provide a social aspect. However, their customization and integration are limited. Additionally, you need to be aware of terms of service, licensing, and privacy issues when you uploading your media to a photo-sharing site.

Gallery (**http://gallery2.org**) is open source, self-hosted photography album software. It provides a rich set of features that a user can put on his own site, or integrate into his web application. There are modules and plug-ins for adding specific functionality to Gallery. Specifically, there are plug-ins available that allow users to create maps of their photos.

The Map Module (**http://codex.gallery2.org/index.php/Gallery2:Modules:Map**) is an add-on to Gallery that geolocates and maps your photos that you upload. The location of the photo can be entered manually using longitude and latitude, via point-and-click on a map to autofill the GPS coordinates, by geocoding an address, or directly from the photo EXIF data.

The GPS Module (http://codex.gallery2.org/index.php/Gallery2:Modules:Gps) adds the ability to output a KML feed for displaying your photos in Google Earth.

For mobile phone cameras, Yahoo! Labs is trying out ZoneTag (http://zonetag.research.yahoo.com/), which automatically geotags your photos when you upload them from your phone by using cell geolocation.

Advanced

ExifTool (http://www.sno.phy.queensu.ca/~phil/exiftool/) is a command-line program written in Perl by Phil Harvery that provides a direct means for reading and writing the EXIF data of a photo. Using ExifTool, you can write scripts to geotag your photographs based on whatever scheme you want to use, such as folder name, or current location (geotag your photos based on automatic geolocation).

What is especially useful for using a utility like ExifTool is that you can write geotagging into the photo application you currently use. For example, you can download a small AppleScript that will allow you to geotag photos in Apple's iPhoto, Microsoft's iView Media Pro (http://highearthorbit.com/projects/applescript/), or Adobe's Photoshop. (http://www.proxel.se/exif.html)

gpsPhoto.pl (http://www.carto.net/projects/photoTools/gpsPhoto/) is a command-line tool for synchronizing GPX tracks and photographs. This script can be used for your own scripts to geocode your photographs. For example, you could have the script geotag your photos when they are uploaded to your server, or dropped into a folder with a GPX file.

Once you geolocate your photos, Photokit (http://www.eparticipation.com/index.php?option=com_content&task=view&id=47&Itemid=5) is a script that will generate the datafile to display your photos in worldKit. You can then easily put this map in your own site.

You can also use OpenLayers to generate a map of your photos. OpenLayers can directly consume GeoRSS and add it as a layer to the map. Therefore, we can get the Flickr feed for a set of photos and specify that we want GeoRSS output of the feed. See the discussion above for how to get the GeoRSS Flickr feed.

Then add the following to your OpenLayers map code to add the new Flickr layer:

```
var flickr_map = new OpenLayers.Layer.GeoRSS( "Flickr",
"proxy.pl?http://api.flickr.com/services/feeds/photos_public.gne?id=51648834@N00&tags=
geotagged&format=rss_200&georss=1");
openlayers_map.addLayer(flickr_map);
```

Now when you refresh your map, you should see the Flickr layer and icons for your geotagged photos.

Next steps

It is possible to geotag more than just your photos. The FreeSound Project (**http://freesound.iua.upf.edu**) supports geotagging audio files and creating maps (**http://freesound.iua.upf.edu/geotagsView.php**). VlogMap (**http://community.vlogmap.org**) geotags video content.

Additionally, any site or service that supports tagging can use triple-tagging to create geotags. Just use the triumverate of:

- `geotagged`
- `geo:lat=`
- `geo:lon=`

and you are on your way to geotagging anything.

Directions to Your Locations

Locations are great, but unless we are imagining ourselves there, eventually we will want to know how to actually get there. As a business owner or someone that needs to publish locations, you will want to make it easy for users to get directions to your locations from wherever they are. For example, a business with several store locations may want to let a customer quickly find out how to reach the nearest store. For someone that organizes sporting events for kids, they may want to let other parents get easy directions to the various fields or arenas.

Publishing directions

So you have customers that want to get to your store. Of course, you have already published your store locations in Microformat adr (and hCard), so users could use tools like the previously demonstrated Firefox extension GreaseRoute to get directions to your locations.

However, for users that do not use Firefox, or do not have that extension, then you can add your own "directions to here." The easiest way is to just put in a hyperlink to the Google Maps or Yahoo! Maps page with this location. Users can then get directions using that service directly. But for a more unified experience, you can embed the directions right in your web site using the free MapQuest OpenAPI (**http://www.mapquest.com/openapi**). To start, you will need to register and get an account on the MapQuest site.

Now, to add the directions, you will need to do the typical web map dance: header, HTML element, JavaScript.

Header:

```
<script
src="http://web.openapi.mapquest.com/oapi/transaction?request=script&key=API_KEY"
type="text/javascript"></script>
```

HTML Element:

```
<ul>
       <li class="adr" id="location1">
        <div class="street-address">1 Main St.</div>
        <span class="locality">Gainesville</span>,
        <span class="region">FL</span>
        <span class="postal-code">32601</span>
         <a href="#" onclick="getDirections('location1')">Get Directions</a>
       </li>
</ul>
<form id="address_form" name="address_form">
       <fieldset>
              <legend>Your Address</legend>
              <label for="a1">Address:</label><input type="text" name="1a" id="a1"/>
              <label for="a1">City:</label><input type="text" name="1c" id="c1"/>
              <label for="s1">State:</label><input type="text" name="1s" id="s1"/>
              <label for="z1">ZIP Code</label><input type="text" name="1z" id="z1"/>
       </fieldset>
</form>
<div id=”map_container” style=”height:400px;width:600px”></div>
```

JavaScript (this uses Robert Nyman's Ultimate getElementsByClassName (**http://www.robertnyman.com/2005/11/07/the-ultimate-getelementsbyclassname/**):

```
<script type="text/javascript">
var mqRoute = null;
function getDirections(location)
{
       mqRoute = new MQRoute("container");
       var store_location = document.getElementById(location);
       var user_location = document.getElementById('address_form');

       var thumbSize = new MQSize();
       thumbSize.setHeight(150);
       thumbSize.setWidth(300);
       var overviewSize = new MQSize();
       overviewSize.setHeight(400);
       overviewSize.setWidth(600);
       mqRoute.primaryMapSize = overviewSize;

       mqRoute.origin.setAddress(getElementsByClassName(store_location, "*", "street-
address")[0].innerHTML);
       mqRoute.origin.setCity(getElementsByClassName(store_location, "*",
"locality")[0].innerHTML);
       mqRoute.origin.setStateProvince(getElementsByClassName(store_location, "*",
"region")[0].innerHTML);
       mqRoute.origin.setPostalCode(getElementsByClassName(store_location, "*",
"postal-code")[0].innerHTML);
       mqRoute.origin.setName("orig");
       mqRoute.destination.setAddress(user_location.elements.namedItem("1a").value);
       mqRoute.destination.setCity(user_location.elements.namedItem("1c").value);
       mqRoute.destination.setStateProvince(user_location.elements.namedItem("1s").val
ue);
```

```
        mqRoute.destination.setPostalCode(user_location.elements.namedItem("1z").value)
;
        mqRoute.destination.setName("dest");
        mqRoute.doRoute("map_container");
}
function routeReturn(mqRoute, status) {
}
</script>
```

Using directions

Besides creating directions to locations you are publishing you will want to be able to get to various locations yourself. For example, it may be useful to have a listing of coffee shops that offer free WiFi connections. When you're out traveling, you can find and get directions to the nearest one. In this project we'll show you how to load a set of locations onto your GPS device and use that to get to you the closest destination.

Figure 8: Most GPS receivers allow you to upload your own waypoints

To start, you will need a listing of businesses or locations. Sites such as GPSPassion (**http://www.gpspassion.com**) have forums where users build and share GPX or CSV (Comma-Separated Values) files of various businesses. Alternatively, you can use a KML file, GeoRSS, or another GPX file you make yourself.

If you have a spreadsheet of locations, you need to arrange the columns in this order: latitude, longitude, name, and description. If you do not have the latitude and longitude yet, I suggest using Batch Geocoder (**http://batchgeocoder.com**) or a similar service to get the coordinates.

Once you have your columns arranged, save the file back as a CSV file. Then open GPSBabel, and convert from CSV to GPX or the file format appropriate for your

GPS unit. You can even export directly to your GPS unit. Some units, such as the Garmin Nüvi, allow you to drop GPX files into the unit using a USB connection.

For an extra level of utility, you can apply a filter to include only locations within a certain distance of a central location. For instance, you may just want WiFi locations within 300 miles of your house.

Now that you have your waypoints stored into your GPS unit, you can now start wandering and find directions to your locations en route.

Track Your Sports

Many neogeographers find comfort in the knowledge that their hobby can also help them get into shape. Sports such as running, hiking, sailing, and skiing all cover large areas, and it can be fun to record your favorite trips and experiences. In this mini project, we will show you how to track your progress, plot it on maps, and even see a movie of your sport.

The Garmin Forerunner is a GPS receiver that straps to your wrist and is an example of a sport-centric device that is useful during your sport, by directly showing you your progress, but also storing your location history as a track for later downloading. Additionally, a GPS logger is usually small enough to be tucked into a pouch or mounted to your equipment and used to track your location.

A waterproof case is a good idea for water-based sports. For example, when scuba diving you could put a GPS Logger in your dive buoy. Then as you swim around and pull along your dive buoy, it will create a 2-D “water” trace of your location. You can then augment this with the depth measurement from your dive computer to create a 3-D trajectory of your dive.

OK, so let us assume you have gone out for a good hike, you took along your trusty GPS receiver, and recorded your tracks. Here’s what you do:

1. Open GPSBabel and export your tracks to a GPX file. Also, save your tracks to a KML file. If there are multiple tracks, it may help to apply a filter to split tracks separated by a certain distance or time (such as multi-day hikes)
2. Open Google Earth, and open your KML file
3. Notice the time slider at the top, expand/contract the time, or slide it to view your track history. Press the Play button (triangle) to animate the track.

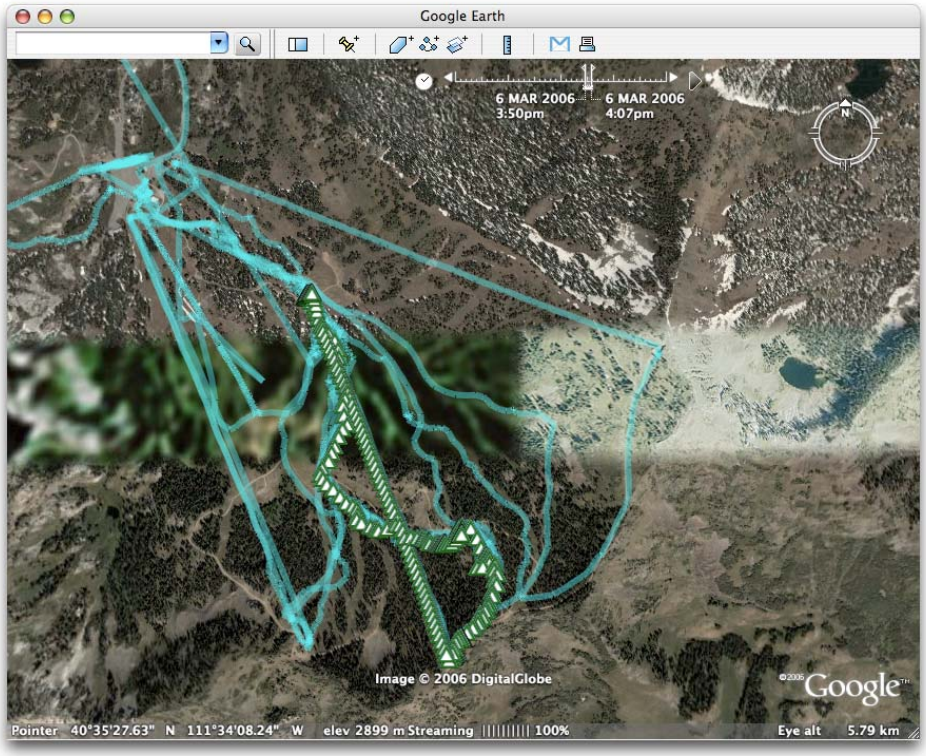


Figure 9: Google Earth supports time in KML files from GPS tracks for visualizing your sports, such as skiing or hiking

There are other sites such as Motion Based (**http://www.motionbased.com**) that provide the uploading, storing, displaying, and analyzing of sport tracks. GPSVisualizer (**http://www.gpsvisualizer.com**) supports a large number of file formats for tracks, and also allows you to generate maps for printing.

GeoTracing (**http://www.geotracing.com/**) is an innovative platform that supports real-time tracking using a Bluetooth GPS receiver and mobile phone to upload your progress. GeoTracing is an open source system, and has been used to build GeoSailing and GeoSkating, just to name a few.

There are a lot of services for specific sports. MTBGuru (**http://blog.mtbguru.com/2006/11/27/introducing-mtbguru/**) is a site for mountain-bike riders, GoFlying (**http://www.goflying.org**) for pilots, and GPSSledMaps (**http://gpssledmaps.com**) for snowmobilers.

Lastly, OpenStreetMaps accepts trails and paths to its database. By annotating and sharing your tracks to OSM, other users can benefit from your tracks.

Mapping Your Genealogy

Genealogy is the study of family history, including lineages (siblings, children), life events (birth, marriages, deaths), and cultures. Our history is comprised of our

family and where they lived, the times they lived in, and the experiences they had. Neogeography is ideally suited to help understand and track your families' lives, travels, and backgrounds.

Perhaps we should call it "geogenealogy."

In this project, we will demonstrate how to record and export your family history, geocode the locations of their events, and then visualize it and share it with others.

Here is a quick listing of the steps we'll be doing:

1. Export/store genealogical information in a tab-separated format
2. Batch geocode to get latitude and longitude of locations
3. Save KML file and display in Google Earth
4. Display a Map or make a web page

To begin, you will want a good start on the documentation of your family's history. There are many programs that are specifically designed for this, such as cross-platform GRAMPS (**http://www.gramps-project.org**) or phpmyfamily (**http://www.phpmyfamily.net**) for web servers. See Wikipedia for a comprehensive list of genealogy software (**http://en.wikipedia.org/wiki/Genealogy_software**). Some programs such as The Next Generation (**http://www.lythgoes.net/genealogy/software.php**) support listing latitude and longitude directly into family events and plotting these on a map.

All of these programs store and can export GEDCOM (GEnealogical Data COMmunication), which is a standard format for storing genealogical information. Currently the format itself is rather odd and ill-formatted standard. The next version, GEDCOM 6.0 will use more consumable and extensible XML. In addition to exporting GEDCOM, the applications can usually also export CSV or Tab-separated file formats, which are more general formats that will be useful later. Alternatively, there are GEDCOM converters, like Oxy-Gen (**http://www.oxy-gen-soft.net/index_en.php**) that can convert GEDCOM to any number of other formats.

You can also just write up your genealogy documentation in a spreadsheet program such as Microsoft Excel. Whether converting to CSV or starting in a spreadsheet, you will want to create several columns: name, location, and description. These will probably be created for you when exporting from your genealogy program. You need to choose what location you will be exporting for this part of the project. You can do birthplace first, and then come back and do death location, or some other events.

Open your exported CSV, or file in Microsoft Excel. Highlight the columns you want, and copy. Then go to BatchGeocode (**http://www.batchgeocode.com**) and copy into the Step #2 box. Click on Validate Source.

After your data is read in, select the appropriate fields in Step #4. If you just have a single Location column, use that for the Address field and leave the others blank. Also choose the Title and Description fields.

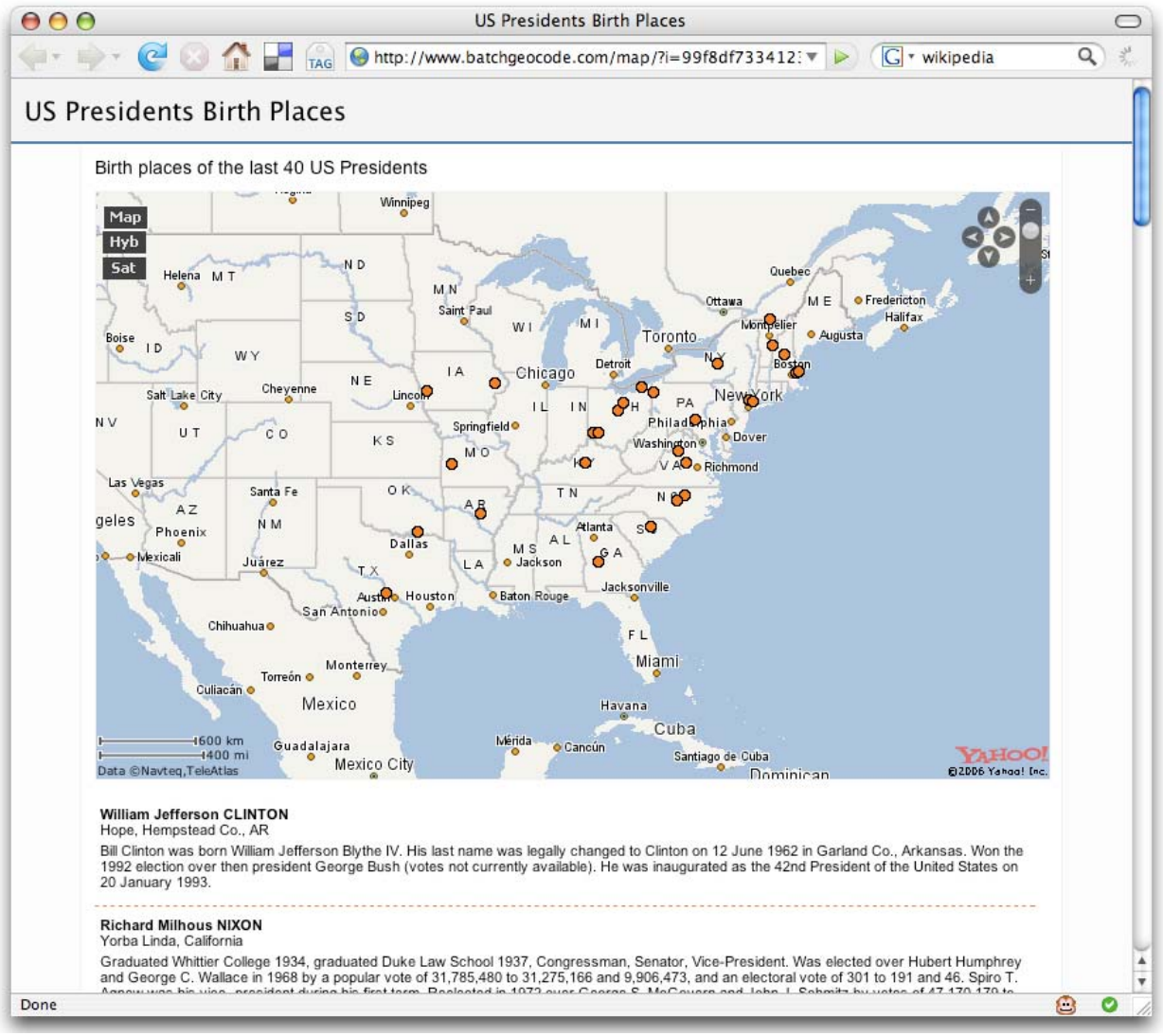
When you are done setting up all the fields, click on Run Geocoder. Once complete, the results will be output in Step #6. You can save this back to your spreadsheet or a text file if you want to keep the data for later. You will also see an example map of the locations.

Below the map, you can "Download to Google Earth (KML) file." Click this and save the file with your spreadsheet. Batch Geocode is also great in that it allows you to make your own web page and map directly from this data.

Click on "Save Map to a Webpage," and fill in the information. You now have a web page you can share with other family and friends.

To visualize the location in 3-D, open the KML file you saved.

I made an example map using the birthplace of the U.S. presidents:
http://www.batchgeocode.com/map/?i=99f8df733412331d3870b25bb99a8cea



For further fun, you can save the spreadsheet file as a CSV again, and convert it to GPX or another format using GPSBabel, and then load it into your GPS unit. Then on your road trip, you can visit the locations of your ancestors.

Where To Next?

Welcome to neogeography. I hope your introduction was a good one. By now, your brain is probably filled with ideas on all the cool projects you can start. The point of neogeography is to have fun with mapping and possibly do something useful in the meantime. The tools and community really strive to make the tools easy and fun to use.

There are a lot of great resources out there for more information. The code and software examples from this book are available online at **http://mapsomething.com**. This includes a complete listing of the links to sites

and resources included in this Short Cut for easy bookmarking. Also, feel free to contact me directly at **andrew@highearthorbit.com**.

Community

Neogeography is more than just the data and applications that let you do fun things. It is also a large community of users and developers that are continuously coming up with new ideas, trying to promote new standards and tools, and having a great time mapping.

You are encouraged to get involved in any of the various geo groups. Here are a couple of starting points that you may find helpful:

*OSGeo (**http://osgeo.org**)*
: The Open Source Geospatial Foundation fosters and supports geospatial technologies and data. Users and developers are encouraged to join and participate.

*Open Geospatial Consortium (**http://opengeospatial.org**)*
: The OGC is an organization that is leading the development of standards for geospatial and location-based services. Membership is limited.

*Geowanking Mailing list (**http://lists.burri.to/mailman/listinfo/geowanking**)*
: An email list of geo-enthusiasts, sharing ideas, projects, announcements and anything geo-related.

#geo on irc.oftc.net
: An IRC channel that includes many people from the geowanking group and other geo-enthusiasts.

*Planet Geospatial: **http://www.planetgs.com/***
: Blog aggregator of numerous neogeography and cartography developers, users, and groups.

Conferences

Conferences are an excellent opportunity to learn about the hottest new projects and technologies as they're being developed and released. You'll also meet other geo-enthusiasts. And once you do some cool projects, you can present them. Here are a couple of good geo-conferences held every year:

*Where 2.0 (**http://conferences.oreillynet.com/where/**)*
: O'Reilly conference on location, mapping, and mobile

*Location Intelligence (**http://www.locationintelligence.net/conference/**)*
: Conference on the business of location technology

*FOSS4G (**http://www.foss4g2006.org**)*
: Free And Open Source Software for Geoinformatics. An international conference covering neogeography, cartography and GIS

Books

Mapping Hacks, Schuyler Erle, Rich Gibson & Jo Walsh
O'Reilly Media, 2005

Google Maps Hacks, Rich Gibson & Schuyler Erle
O'Reilly Media, 2005

Web Mapping Illustrated, Tyler Mitchell
O'Reilly Media, 2005

Else/Where: Mapping, edited by Janet Abrams & Peter Hall
University of Minnesota Press, 2006

Georeferencing, Linda L. Hill
MIT Press, 2006

General links

- **http://del.icio.us/tag/geo**
- **http://del.icio.us/subscriptions/starhill_blend**

Essential Tools

Here is a list of the essential tools you should already have installed or bookmarked on your computer.

- GPSBabel: **http://gpsbabel.org**
- Google Earth: **http://earth.google.com**
- Geonames: **http://geonames.org**
- Batch Geocode: **http://batchgeocode.com**

Mapping Libraries

- Mapstraction: **http://www.mapstraction.com/**
- OpenLayers: **http://openlayers.org**
- Google Maps API: **http://www.google.com/apis/maps**
- Yahoo! Maps API: **http://developer.yahoo.com/maps/**
- MapQuest OpenAPI: **http://www.mapquest.com/openapi/**
- MultiMap: **http://www.multimap.com/share/api_demos/**