Banding Birds with MapServer CGI

By Robert S. Wardwell and Kevin W. Laurent

U.S. Geological Survey
Cascades Volcano Observatory
1300 SE. Cardinal Court, Bldg. 10, Suite 100
Vancouver, WA 98683
Telephone: (360) 993-8908
Fax: (360) 993-8980
e-mail: rwardwell@usgs.gov

U.S. Geological Survey
Patuxent Wildlife Research Center
12100 Beech Forest Rd. Suite 4034
Laurel, MD 20708 - 4034
Telephone: (301) 497-5652
Fax: (301) 497-5826
e-mail: klaurent@usgs.gov

Since its establishment in 1936 as the nation's first wildlife experiment station, the USGS Patuxent Wildlife Research Center (PWRC) in Patuxent, Maryland, has been a leading international research institute for wildlife and applied environmental research. A primary goal at the PWRC Bird Banding Laboratory (BBL) is to manage the administration of bird banding permits, coordinate banding efforts, and collect data scientists can use to analyze such things as species behavior, migratory patterns, and the overall health of a species. Recently, computer scientists at the BBL have developed an ORACLE based permitting and banding administration system focused on tracking a wide range of data about banded birds and managing the BBL bird banding efforts (Figure 1). One component of this system relies on MapServer, an open source Web mapping solution, to collect an absolutely crucial set of data: the geographic location where a banded bird is encountered or recovered in the field. This paper describes how the BBL uses MapServer to collect scientific data and provides some insight into implementing MapServer.

MapServer was originally developed by the University of Minnesota ForNet project in cooperation with NASA and the Minnesota Department of Natural Resources. Presently, the MapServer project is hosted by the TerraSIP project, a NASA sponsored project between the UMN and consortium of land management interests. MapServer has become a popular tool by which to render spatial data such as vectors, maps, and images to the Web. The MapServer Website http://mapserver.gis.umn.edu/ is a valuable resource for learning more about MapServer.

MapServer's role in the BBL permitting and banding administration system is to generate interactive maps used to plot and record a geographic location where a bird was encountered or recovered. These data are currently captured at USGS call centers in Patuxent, Maryland, and Walla Walla, Washington, where reports are made when a banded bird is encountered or recovered in the field. USGS call center employees ask the caller to describe the bird's location and then use an interactive map to find the location and record the approximate coordinates. First, the caller describes a location using a common feature name, such as the name of a town, landmark, park, or water feature (Figure 2). Next, the BBL call center queries the system to search for the location in a feature names gazetteer that contains millions of common feature names in the US and abroad. Finally, MapServer is called upon to render an interactive map of the selected area, with tools that allow the call center employee to record the location where the bird was encountered (Figure 3). Since every caller is not equipped with a GPS or map, MapServer makes it possible to utilize the caller's relative proximity to known geographic features and calculate an accurate longitude and latitude values for their location.

Prior to using MapServer, these geographic data were collected by the BBL as a pair of vectors showing a spatial relationship between a known geographic feature and the location where a bird was encountered or recovered. For example, a banding encounter may be described as being "2 miles east and 3 miles north of Camas, Washington." These descriptions could be intersected and plotted within a mesh of 10 minute grid cells to add more quantitative measure to the data. This approach, however, led to

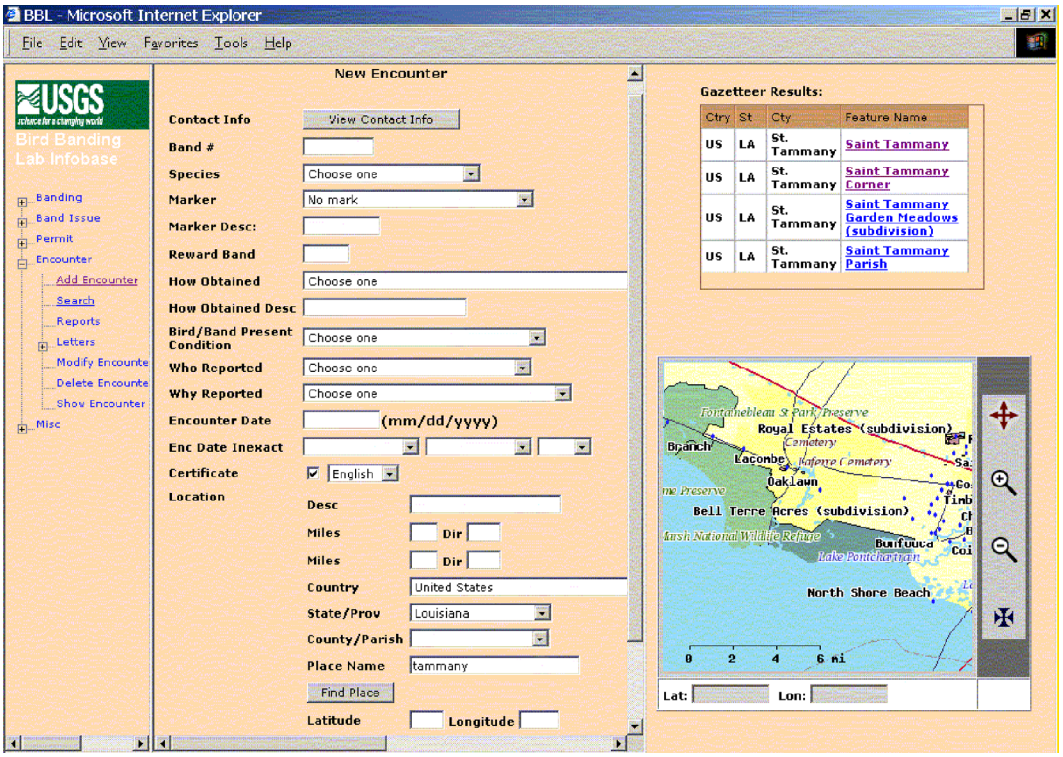

Figure 1. Screenshot of the banding management interface displays a data input form where information is collected about birds that are encountered in the field. A geographic search is initiated in the middle frame, which opens the MapServer interface in the right hand frame of the Web browser. The user identifies the location of the bird encountered by plotting it on the interactive map. Latitude and longitude coordinates are passed back to the main data entry form in the center of the page. Submitting the Web form records the event.

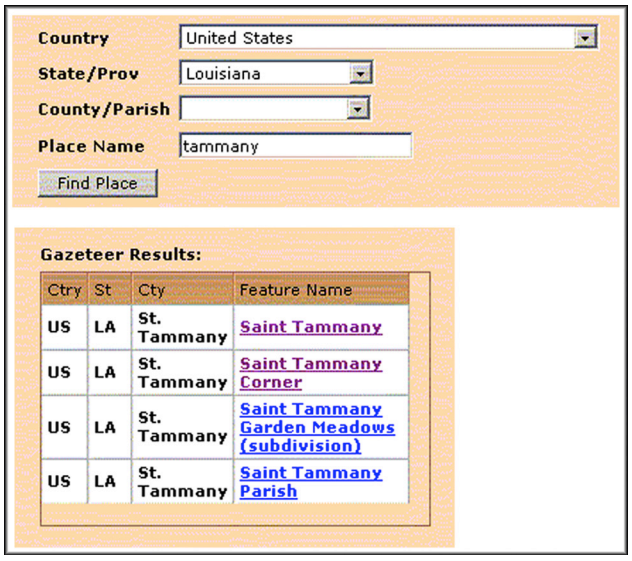

Figure 2. A place name search generates a list of possible name matches being returned in the *Gazetteer Results* list. Selecting a place name from the list passes the feature name coordinates to MapServer, which returns a map centered on the selected location.

a large degree of inaccuracy. Now, with the aid of feature rich maps rendered by MapServer, geographic location may be communicated more effectively and more accurate data is stored as coordinate pairs in the BBL database.

The maps generated by MapServer contain a selection of global and national spatial data layers. These layers include common landmarks, hydrology, urban areas, transportation routes, wildlife refuges, and parks for the United States along with Global political boundaries and populated places for the entire globe. Also included are transportation routes, hydrology, wildlife refuges, and parks for Canada and Mexico. MapServer is capable of serving both raster and vector data in a multitude of formats. Vector data sources, for example, may include ESRI shapfiles, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL. For the BBL banding application, base map data were downloaded from the Web in ESRI shapefile format. Feature data from the USGS Geographic Names Information System (GNIS), Natural Resources of Canada Gazetteer, and the National Geospatial Agency's Geographic Names System (NGA GNS) were made into shape files, which made it possible to search for and label over 3 million features on the maps. Quadtree based spatial indexing

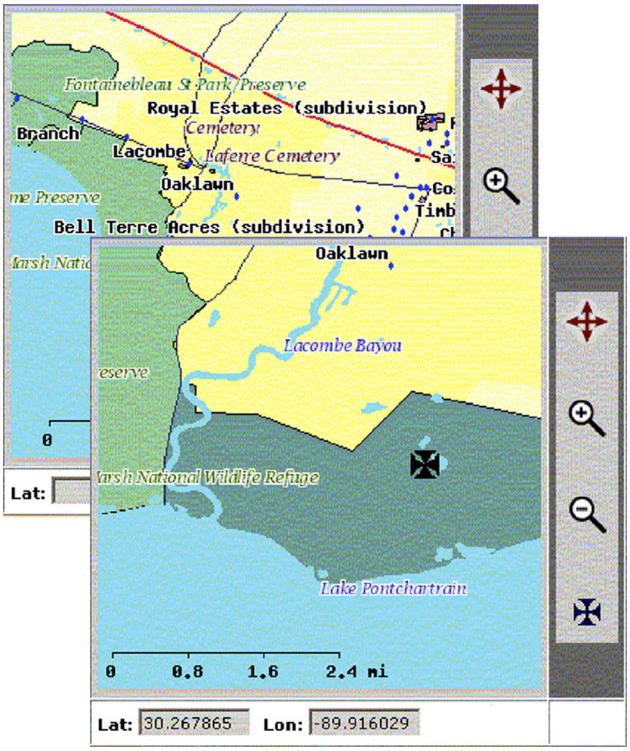

Figure 3. The interactive map created by MapServer contains tools used to zoom in, zoom out, pan, and mark a point on the map. In the above example, the user has zoomed in and clicked on the map, which places an "x marks the spot" symbol at the location where a bird was encountered. MapServer calculates the coordinates of the point which is then submitted to the database.

```
LAYER
  NAME "USParks"
  DATA parks_shp
  STATUS default
  TYPE POLYGON
  MAXSCALE 2000000
  LABELMAXSCALE 2000000
  LABELITEM "Name"
  CLASS
    COLOR 153 204 153
    OUTLINECOLOR 0 0 0
    LABEL
      SIZE 8
      TYPE TRUETYPE
      POSITION auto
      FORCE false
      COLOR 51 102 0
      OUTLINECOLOR 255 255 255
      FONT lucida-bright-italic
    END
  END
END
```

Figure 4. The *map* file (.map extension) is a text file created manually or built using one of many open source tools. This file describes the data sources for your map and defines the map extent, data layers, symbols, and layer classification.

of each shape file was performed using the MapServer `shptree` utility, which helps speed the delivery of map data over the Web.

In the BBL banding application, MapServer operates as a CGI program that handles requests and responses. The map and everything about the layers within the map, such as cartographic symbols and the classification of the layers, are controlled through a MapServer *map* file. This file is a hierarchical text file with a .map extension that describes each of the data layers to be included in the map, and describes how each layer is to appear (Figure 4). The interactive mapping interface, which contains tools used to zoom, pan, or plot the location of a banded bird encounter, is controlled by an HTML and JavaScript template file. The template file contains unique CGI variable tags, which are handled by MapServer each time the user makes a request for a new map (Figure 5). The map itself is rendered onto the Web page as a static image, such as a .gif, .png, or .jpg that is replaced when MapServer processes a new set of input parameters from the HTML template.

When the BBL permitting and band management application went to production use in March 2006, the mapping component contained over 50 data layers, which included data for 3.6 million named points in more than 50 countries. MapServer proved itself worthy by handling these large datasets and displaying them efficiently over the Internet. Furthermore, we found MapServer to be a simple, elegant solution for improving the data acquisition at the BBL. The success of this project can be attributed to the collaborative efforts of biologists, computer scientists, and GIS professionals, along with the open source development community surrounding MapServer.

REFERENCES

Kropla, B., 2005, Beginning MapServer-Open Source GIS Development: Apress Publishing.

Lime, S., (from the MapServer Website), 2006, Welcome to MapServer, accessed at http://mapserver.gis.umn.edu/.

Mitchell, T., 2005, Web Mapping Illustrated - Using Open Source GIS Toolkits: O'Reilly Publishing.

Patuxent Wildlife Research Center Website, 2006, Patuxent Wildlife Research Center – Mission: accessed at http://www.pwrc.usgs.gov/aboutus/mission.cfm.

```
<h3>USGS Breeding Bird Survey Interactive Route Map</h3>
<form action="/cgi-bin/mapserv" name="mapForm" method="get" target="_self" onsubmit="return(map_Submit(this))">
   <input type="hidden" name="map" value="[map]">
   <input type="hidden" name="imgext" value="[mapext]">
   <input type="hidden" name="mode" value="browse">
   <input type="hidden" name="zoomdir" value="[zoomdir]">
   <input type="hidden" name="zoomsize" value="[zoomsize]">
   <input type="hidden" name="layers" value="[layers]">
   <input type="hidden" name="imgxy" value="149.5 149.5">
   <input type="hidden" name="imgbox" value="-1 -1 -1 -1">
<table width="460" border="1" cellspacing="0" cellpadding="4" align="left" bgcolor="#666666">
 <tr>
  <td align="center" bgcolor="#CCCCCC" valign="top">
   <input property="" type="image" name="img" src="[img]" width="400" height="550" border="0" alt="map image">
  </td>
 <td>
```

Figure 5. The HTML template file defines a map interface that can be customized and enhanced with JavaScript. The MapServer CGI program will process your map file and pass values to “substitution strings” that are enclosed in square brackets (‘[ ]’). The CGI variable examples in this code snippet include `map`, `mapext`, `zoomdir`, `zoomsize`, `layers`, and `img`.