

# Getting the Maps of the Bordner Survey Into Google Earth – A Tutorial

By Samuel F.B. Johnson  
University of Wisconsin GIS Certificate Program  
State Cartographer's Office  
Spring 2008

The series of steps that follow detail the methodology used in getting an array of Bordner Survey maps from source digital imagery into Google Earth as georeferenced, tiled image overlays with associated legend information and metadata separately displayed. This document is intended to demonstrate the work flow the author followed and serve as a usable template for any other would-be historical cartographer interested in replicating the process on a subset of the Bordner Survey. Please note that this process, as an initial foray into modern cartographic visualization with the collection, is just one method of potentially many and was judged the most practical and efficient; other methods may work and be as valid. In addition, instructions are written from the perspective of 'lessons learned,' that is, only approaches and solutions that achieved desired results at each step. This being a new methodology, the intent is to create a modest learning curve for other individuals seeking to obtain the same or similar results. Questions or comments are welcome; please refer to the contact information provided at the close of this document. This project would not have been possible without the crucial and constant aid and advice of a number of knowledgeable people, recognized individually in the final section.

## Software Requirements

*ArcGIS suite* (as new a version as possible – I used version 9.2 and relied heavily on Catalog and Toolbox)

*ArcINFO (WorkStation)* – for command line image processing using Arc Macro Language

*FWTools* (<http://fwtools.maptools.org/>) – open source GIS binaries package for Windows and Linux including GDAL/OGR support and Python scripting language; latest version at the time of writing is 2.1.0

Python script *gdal2tiles.py* – information here - <http://www.gdal.org/gdal2tiles.html> - script itself can be found and downloaded here - <http://www.klokan.cz/projects/gdal2tiles/>

Earth Point *Excel to KML* (<http://www.earthpoint.us/ExcelToKml.aspx>) – open source conversion tool for easy exporting of point/placemark data in a spreadsheet to Google Earth KML, based on lat/long coordinates (Excel required)

*Google Earth* – free, basic version is all that is required

You will also need web server space in which to store the resulting Google Earth files.

Note: Adhering to consistent, detailed directory and file management practices is important in a project of this nature; this also applies to naming conventions. Additionally, in particular imagery geoprocessing tasks, both automated and manual, raster file names must be kept to a limited length. These limits will be noted in the instructions that follow.

## Procuring and Georeferencing the Bordner Survey Maps

The first step of the process is the acquisition of maps for any desired area and their assignment to a spatial reference system. The creation of georeferenced Bordner Survey imagery itself makes it a great ‘value-added’ GIS data resource. Any source maps used in this process work best as TIFFs (Tagged Image File Format) due to the format’s inherent qualities of lossless or no compression, generally excellent rendering and resolution, compatibility across most GIS and graphics software, and storage of georeferencing ‘tags’ in file headers.

There are two options for procuring Bordner Survey maps: scanning from the 8-1/2 x 11” county compilation books at an appropriately high and consistent resolution, or acquiring the TIFF imagery from the UW Libraries Digital Collections Center, itself derived from the compilation books, scanned, and used as the source for the Bordner Survey Digital Collection in 2003. Access to any county compilation books may be provided by Jaime Stoltenberg in Robinson Map Library, collection staff at Steenbock Memorial Library, or the Library-Archives staff at the Wisconsin Historical Society. Permission to access and use Bordner Survey maps already digitized may be possible by contacting the UW Digital Collections Center. The collections of Bordner Survey compilation books in the holdings of Robinson Map Library and Steenbock Library are incomplete; the Historical Society and Digital Collections have the only known complete Bordner map holdings. (Note: The author makes no guarantees regarding access to and use of the Bordner Survey maps from any of the above entities, nor any related restrictions, fees, etc.)

In choosing smaller areas of the Bordner Survey to bring into Google Earth—anything from a single township to an entire county is reasonable—it is important to remember that the county compilations contain just one Public Land Survey System township (six miles square) per page, not complete civil townships. Depending on topography and hydrology in a given area, *survey* townships may or may not also conform to *civil* township boundaries – both possibilities may be seen in the results I obtained for the small areas I focused upon in my work. Therefore, mapping areas which include irregularly-sized civil townships will require use of additional survey township maps (pages) to completely cover the selected township(s). Refer to the compilations and their index maps to figure out your needs.

After acquiring maps, you will then need GIS vector shapefiles to use as reference layers in georeferencing. You will enter control points based on these layers; they will also serve as a locational aid. I relied on both PLSS and civil township shapefiles and PLSS section boundaries; others such as Wisconsin state and county boundaries may also be used. PLSS layers and county and state boundaries can be downloaded as zip files at no cost from the WDNR GIS repository (<http://dnr.wi.gov/maps/gis/geolibrary.html>), specifically its external FTP site: <ftp://gomapout.dnr.state.wi.us/geodata/>. Download PLSS township and section zips from the *landnet* folder. Civil townships, also referred to in U.S. Census nomenclature as Minor Civil Divisions and including both township and municipal boundaries, can be downloaded free from the ESRI Census 2000/TIGER line data downloads page here - [http://arcdata.esri.com/data/tiger2000/tiger\\_statelayer.cfm?sfips=55](http://arcdata.esri.com/data/tiger2000/tiger_statelayer.cfm?sfips=55). The layer name is County Census Divisions 2000 and the data will download as zip files. If you are using Bordner Survey map scans (Digital Collections), the TIFFs are named by county abbreviation and ‘local

identifier' cataloging number, so individual images must be previewed in ArcCatalog or a graphics program to select the correct ones. They can be renamed more descriptively in Catalog e.g. *county\_township (civil and/or PLSS).tif*, and should acquire meaningful names (output TIFF names) in georeferencing, at the latest. Similarly, consider renaming the County Census Divisions 2000 layers you download.

Note that WDNR GIS data comes in a standard coordinate reference system – Wisconsin Transverse Mercator (WTM) 83, NAD83 (1991) or WTM83/91 – while the above Census data comes undefined but is in the default NAD83 geographic coordinate system. For georeferencing, I recommend that one create individual ArcGIS projects, one for each non-adjacent area e.g. sites in different counties, as seen in my own work. This will prevent display and projection issues. In ArcMap, for each site chosen use the Select tool to pick all county census divisions (civil townships), PLSS townships, PLSS sections, and/or county boundaries and export your selections as individual shapefiles.

At this point you may choose to re-project (in ArcToolbox) your county census divisions layers to WTM83/91, or simply project those layers 'on-the-fly' in your ArcMap project, provided your DNR PLSS layers are added to the display first. Conversely, given the processing you will do in later steps and because Google Earth utilizes the geographic coordinate system EPSG:4326 (WGS84 lat/long), you may consider re-projecting all your reference layers to that system prior to georeferencing. If you opt not to do this, simply re-project (batch projection suggested, at least by county) your Bordner maps from WTM83/91 to WGS84 lat/long after they are georeferenced. Either way, if you get your Bordner maps into this Google Earth-compatible coordinate system early, you will not have to worry about projection issues later.

You will then georeference your Bordner maps in your individual ArcMap projects, each containing its particular reference layers and the map TIFF. Add the Georeferencing toolbar (View→Toolbars→Georeferencing) and refer to ArcGIS Desktop Help for specific instructions or a refresher on georeferencing raster datasets. In order to fit the previously non-spatially referenced Bordner Survey rasters as precisely as possible to true coordinates, I generally used a large number of control points, generally anywhere from 10 to 19 pairs. In cases where areas covered on a map sheet were unusually small, e.g. fringes of civil townships bordering a river such as Caledonia in Trempealeau County, I found it necessary to download additional Census 2000/TIGER layers like roads and hydrography in order to attain an acceptable number of control points for optimal transformation and rectification. Accurate placement of control points on the source and destination layers at a constant scale (as large as you deem appropriate) is crucial. Ideal placement of control points is at section boundaries as follows: township corners, boundary midpoints, section sub-corners, sub-sub-corners, and center points. This will vary by area and is dictated by the shape of the township – whether it conforms to survey township boundaries or otherwise. Any section corners will work, ultimately. You may find when zoomed in to locate control points that your various reference layers are not precisely aligned despite having the same projection information; if this is so, only register a map to one layer for the sake of consistency. For this I recommend PLSS section boundaries.

Acceptable root mean square (RMS) errors are a concern in georeferencing the Bordner maps. You will find that they are calculated in the toolbar's link table upon entering the fourth control

point, and by default will update as you enter additional points. Keep total RMS errors at a reasonable level; there is no ideal limit, but in my own work resulting RMS errors ranged from about 20 to 60 for the majority of the map images I chose. Use best judgment. In the link table, note the Transformation method – 1<sup>st</sup> Order Polynomial (Affine). For best results, maintain this default, and to complete the transformation process for a map, click Georeferencing on the toolbar and Update Georeferencing. Then, save out the georeferenced map as a TIFF by selecting Rectify from the same menu, keeping Nearest Neighbor resampling method and default compression settings in the dialog. Differentiate your Bordner maps before and after georeferencing with your naming conventions (e.g. *\_refd.tif* file suffix) and/or a separate sub-directory for your georeferenced maps. Refer to ArcGIS Desktop Help for any georeferencing instructions and tips.

### **Collar Processing of Black and White Bordner Maps**

Anyone with a passing familiarity with the Bordner Survey maps will see that, like virtually every reference map, all have collars with detailed names, legends, and other metadata. In order to display adjacent Bordner maps seamlessly in Google Earth, these collars must be dealt with, and the method used – indeed, the only known way – involves processing using the highly robust Arc Macro Language (AML) in command-line ArcInfo. Your georeferenced Bordner maps will display in ArcGIS with all pixels, in the map itself or the collar, in one of two bands (0 - black or 1 - white). View the properties of one of your maps in ArcMap (Symbology tab) to see this. This is normal for raster imagery not previously digitized and otherwise altered. Collar processing makes a distinction between map collar and actual geography based on the map's border, that is, it assigns collar and map to separate raster bands. This processing results in a good value-added product for digital archiving of these raster layers or their use in a GIS. Thus, the resulting collar-processed image files should be copied, and either the originals or copies should be used in future steps. Note: any 3-band RGB/color images among the maps you have selected will be processed in a later step; their collars cannot be processed by the same AML script used for the single-band, 8-bit black and white maps.

Credit for original instructions in this step of the process goes to Mitchell Moline, GIS System Administrator at WISDOT, who adapted an AML script and workflow used in collar treatment of DRGs for the state to my project and the particular GeoTIFF imagery used. Refer to the *HowTo\_BW* Word document in the *student.zip* folder for specific steps to follow in the collar treatment of your black and white Bordner map imagery. Because they are already effectively laid out in that Word doc, in this section I will only enumerate them for the purposes of clarification and to emphasize the purpose of each step.

In the *student.zip* file you will see a directory called *student*. Copy this directory to the root C:\ drive on your own machine; for the AML scripting on your images and collars to work properly you must keep this simple path and the directory structure as is. Take a moment to examine the *student* directory. It contains seven sub-folders: *aml*, which contains the AML script *collar\_bw*, which will be run on your georeferenced b&w maps; *clr*, the location of a universal colormap file for your images; *cov*, which serves as temporary storage for the Arc coverages which are generated in the running of *collar\_bw.aml*; *grid*, which contains the ESRI grids you will create from your GeoTIFF images; *image*, the repository for your GeoTIFFs as you begin processing

their collars with AML; *mx*d, where you will store an ArcMap project you will use in digitizing necessary shapefiles for processing, described next; and *shape*, which will store the polyline 'clip' shapefiles you will create for each map image to serve as the aforementioned border that *collar\_bw.aml* will use in processing. Already in the *student* directory are the files related to a township map in Vernon County -- the collarized map example outlined by Mitch in the instructional Word doc.

To collarize your black and white Bordner map images, refer again to the *HowTo\_BW* document. Ignore all instructions on the first page (irrelevant) except the final direction; copy all your georeferenced maps to the *image* folder in C:\student. Regarding Step 2, defining the projection for your maps in the *image* folder can be done in ArcToolbox as well (Data Management Tools→Projections and Transformations→Define Projection). You will convert your GeoTIFFs to GRID images in Step 3 according to the instructions given; as hinted, you may convert them all at once as a batch process. Perform this conversion in ArcToolbox at Conversion Tools→To Raster→Raster to Other Format (multiple); keep input image names brief but descriptive to prevent naming confusion later, as output GRID names are automatically generated, and are truncated versions of input image names. Next, create your blank polyline shapefiles, one for each map image, in ArcCatalog as described in Step 4 of the document. In doing so, you will Import the corresponding image's coordinate system information. Next, give these shapefiles actual geometry by on-screen digitizing, via the Editor toolbar. Follow Step 5 as outlined, digitizing your clip shapefiles against their corresponding map images in the ArcMap project (in *./mx*d) mentioned. Be sure to digitize these shapefiles at a large and consistent scale, and capture all nuances of the map/collar borders – road curves and endings, road symbols, labeling, and any other details specific to each map. Turn on Snapping (Editor→Snapping, and Options) as well to allow you to easily close the borders you digitize. Follow Step 6 just as directed. To clarify, 'open dos window at *./aml*' means to open said DOS/command line window (e.g. Start→Run→cmd) and navigate to the *aml* folder like so: `cd C:\student\aml`. In the seventh and final step you will run the AML script at Arc: in the command line window. The syntax/arguments in running *collar\_bw.aml* are very simple; however, you must keep output collarized GeoTIFF names short (less than eight characters) and free of numerals. Your .tif extension should be left off as well. The script will take up to about 30 seconds to run on each image.

Be sure to preview your resulting images in ArcCatalog for quality assurance and quality control. If there is an error in running the script, check your syntax and input/output file name length, and delete any temporary files (e.g. DAT files) created in *cov*; be careful not to erase anything important – checking the Last Modified date/time description in Windows Explorer is a clue whether a file was created during execution of the script and will help you decide if it should be deleted. When you have successfully run *collar\_bw.aml* against your map GRIDs and their clip shapefiles, define projections for all your output collarized images in Toolbox. View them in ArcMap, changing raster band display colors in Properties→Symbology tab for each to show the results of your work as well as geographic alignment.

You now have value-added, collarized black and white Bordner map imagery. Copy these images and use either the originals or copies for the next step of the process: clipping off the map collars, for which you will again utilize AML commands in ArcInfo.

## Clipping the Collars from B&W and Color Bordner Maps

The next major step in the preparation of your Bordner maps for seamless display in Google Earth is to completely remove their collars individually, again using AML scripting and the same *C:\student* directory structure. Credit again goes to Mitchell Moline at WISDOT for the script and methodology. In this step, however, you will handle any color (3-band RGB) Bordner map images you are using with a different AML script, *collar\_color.aml*. Also, you will process your black and white maps via individual AML Arc and Grid commands at the command line. You will end up with collar-less Bordner maps (geography only) that will be ready to mosaic into virtually seamless images in ArcToolbox.

For this step, refer to the Word document in *C:\student* entitled *Steps for color images.doc*. Regarding Step 2, *collar\_color.aml* is already located in the *aml* folder in your *C:\student* directory structure (for the purposes of this tutorial); you only need to copy your georeferenced color Bordner maps, if you have any, and collarized b&w Bordner maps to the *image* folder. For Step 3, check on your projection information for each image in ArcCatalog (Metadata tab→Spatial) and Define Projection if needed. In Step 4 you will convert your GeoTIFFs to GRIDs in the *grid* folder (output workspace) as in the b&w collarizing step – Raster to Other Format in ArcToolbox. It is essential to keep the names of input TIFFs brief but descriptive (nine or fewer alpha characters) to ensure the AML script will handle them, and again, to prevent naming confusion as output GRID names are truncated from the input file name. Similarly, in the next step create clip boundary geometry for each color map in the *./shape* folder using the ArcMap project in *./mxd* if you have not already done so. Carry out Step 6 for each color image as directed, again opening the command line window at *aml* and running *collar\_color* from Arc: prompt. This AML script operation bypasses the collar treatment done for the b&w maps and simply removes your color maps' collars based on the clip shapefiles you digitized.

Open the script in Notepad to view the component steps of the processing: 1) *shapearc* (creates coverage from clip shapefile), 2) *build* (create topology for that coverage – line and poly), 3) *grid* (switch from Arc to Grid module) 4) *gridclip* (clips collars based on that shp-turned-coverage), 5) *makestack* (creates GRID 'stack' for each of the color image's three bands), and 6) *gridimage* (converts GRID stack set for each image into output raster, in our case TIFF/GeoTIFF). Steps 1-3 and 6 are carried out from Arc, while 4 and 5 are Grid operations. The script *collar\_color.aml* is written to perform these commands and related sub-steps in one operation for color Bordner map images. For the collarized black and white Bordner maps, these Arc and Grid commands must be entered individually, 'by hand' in ArcInfo as opposed to execution with a single script – all except for *makestack*, because your b&w map images have only a single band, not three.

At any step of the b&w image collar clipping, you can access a command's detailed syntax/usage and other background information (in a pop-up window) by typing 'help' at the ArcInfo prompt. Or, type the name of the command from Arc: or Grid: to get list the usage briefly. Below are the AML commands reiterated, with arguments and notes. Run the series of commands on each b&w image; done correctly, this collar clipping is a very quick process with most of your time devoted to allowing your CPU to crunch.

Available at Arc:

- 1) **shapearc** <input\_shp> <output\_cov> (arguments for out\_subclass and DEFAULT/DEFINE unnecessary)
- 2) **build** <coverage> **poly** – and – **build** <coverage> **line**
- 3) **grid** (switch to Grid)

Available at Grid:

- 4) **gridclip** <input\_grid> <output\_grid> <clip\_cover> # (default BOX parameter)
- 5) **arc** (switch to Arc)

Available at Arc:

- 6) **gridimage** <input\_grid> <input\_colormap\_file\*> <output\_image> **TIFF (output format) NONE (no compression)**

\*In order to get an output TIFF that properly displays, given that the input is a single grid, you must create an ASCII colormap file (a simple text file with a .clr extension) in Notepad – see the notes for *gridimage* in that command's ArcInfo Help doc. I worked around this by using the Flash Color Table file *collar\_bw.clr* in *C:\student\clr* as a template and saving out the new file as *bwcolormap.clr* to (I believe) the same folder. The only alteration done to *collar\_bw.clr* was changing cell 3's RGB values from all zeroes to 255's, the result being that cell 1 is white (value of 255 for RGB) and cell 2 black (value of 0 for RGB). This modification is all that is needed for your *gridimage* command to work on the b&w images – simply use *bwcolormap.clr* as input\_colormap\_file.

You should now have successfully created clipped GeoTIFF images of your color and b&w Bordner maps. Preview them in ArcCatalog for quality assurance and quality control. Check on projection information there (Metadata→Spatial) and define in ArcToolbox if necessary.

## **Create Mosaics for Your Selected Sites of the Bordner Survey**

Your clipped Bordner maps are now ready to become nearly-seamless image mosaics in ArcToolbox. To mosaic your maps by site (e.g. multi-township area in a given county), access the mosaic dialog in Toolbox at Data Management Tools→Raster→Mosaic to New Raster. Enter your individual input rasters (clipped GeoTIFFs). (Note: In my own work the largest site contained eight such rasters. Arc can likely handle many more than this without crashing or otherwise failing to execute such a Toolbox task. Pick your sites and their sizes wisely.) Select your mosaic's output location, output name with extension (.tif), and select or import the required coordinate system (GCS→World→WGS 1984 lat/long). Leave the default pixel type – 8 Bit Unsigned – as this applies to both your b&w and color Bordner imagery. Leave the Cellsize default. Number of bands is 1 for b&w and 3 for color images. Keep the Mosaic Method and Mosaic Colormap Mode defaults, Last and First, respectively. Amount of time needed for the Mosaic to New Raster Tool to finish the mosaic operation will vary depending on the number of input rasters and their sizes. Preview in ArcCatalog for quality control, noting the degree to which your images are now 'seamless' (the mosaic operation is inherently imperfect), and check to ensure that projections are defined.

## Handling the Presence of Colormaps in Your B&W Imagery

At this intermediate step, you will have to carry out a simple raster data conversion on any black and white Bordner map mosaics you are using in order for future processing to give the correct result. Your b&w mosaics at this point will be single-band, 8-bit GeoTIFF images *with* colormaps (view Properties in ArcCatalog at any time to see this). You will remember that colormaps were generated after collars were clipped and your images were converted from GRIDs back to TIFFs (*gridimage*). Any color (3-band RGB) mosaics you seek to display, as a result of differences in image properties and past processing steps, have no colormaps. For the Python scripting you will carry out in the next step to give the desired results, you will need to 'lose' the colormaps associated with your b&w imagery via conversion from single-band to 3-band RGB. This is accomplished in ArcCatalog.

In ArcCatalog, for each image, right-click the image name and select Export→Raster to Different Format. Enter input and output raster names, be sure to check Colormap to RGB and leave all other defaults. Click OK to run the conversion. Preview your new 3-band b&w mosaic and view properties; you will notice image size has as much as doubled or tripled, but colormaps are now absent. Your b&w Bordner maps can now be properly processed in the next step.

## Create Tiled Image Overlays From Your Mosaics

In this step you will use the Python script *gdal2tiles* to process and publish your Bordner Survey mosaics to the Internet (finally!) in the form of individual Google Earth KML layers and, optionally, to create simple Google Maps HTML pages and/or OpenLayers HTML pages. This simple but robust script generates a tile directory (TMS-compatible) one can copy to a web server to allow easy publishing and quick rendering of your raster maps online, which circumvents the user having to author one or more Web Mapping Services in order to get raster content online. Visit the script's project page here - <http://www.klokan.cz/projects/gdal2tiles/>; pay particular attention to the introductory information and the Demo and Installation sections.

If you decide to create Google Maps HTML pages for any image mosaic, you will need to sign up for a free Google Maps API key here: <http://code.google.com/apis/maps/signup.html>. The key is a lengthy alphanumeric string with upper and lower case characters; save it in a secure place, as it will need to be entered manually as an argument every time *gdal2tiles.py* is run.

To start, follow the Installation instructions on the script author's aforementioned project page. From the given link, download the open-source package FWTools, which includes support for both GDAL/OGR and Python, and right-click and save *gdal2tiles.py* to the root C:\ directory. You will also run it in the FWTools shell as specified.

Syntax for *gdal2tiles* in the FWTools shell, with optional arguments, is as follows:

```
python c:/gdal2tiles.py -title "Title" -publishurl http://yourserver/dir/
-nogooglemaps -noopenlayers -nokml
-googlemapskey KEY -forcekml -v
input_file output_dir
```

Refer to the script reference page here for more usage information - <http://www.gdal.org/gdal2tiles.html>. In my own work, I omitted the arguments *-nogooglemaps* and *-nokml* as I wanted both Google Maps HTML pages and KMLs for every site. You may wish to generate OpenLayers-based HTML pages. The 'verbose output' argument, *-v*, is optional; because the script can take up to several minutes to run, however, it can be helpful to view its progress in the output.

Open the FWTtools command shell and enter the script with the above syntax and desired arguments. The *-title* argument is the name to be applied to all output layers, so make it as descriptive as possible e.g. *Fullcountynametownship*. Note that *input\_file* refers to a given map mosaic in EPSG:4326, while *output\_dir* is the destination to which your output tile directory and layers will be piped. For both, you will have to enter the *full* path for the script to run properly.

Run *gdal2tiles* on all of your mosaics and view the resulting tile directory structure and layers in your output directories. Your Google Earth KMLs should load in the Temporary Places folder and zoom in automatically to the overlay; experiment with zooming and the overlay transparency slider and note the alignment of your new overlays to the underlying color base imagery. With the Terrain layer checked, experiment with oblique views and note how your overlays wrap to local topography. If displaying your work on the Web, be sure to copy the complete folder containing your *gdal2tiles* output to the path specified in the *-publishurl* argument.

## Replicating Map Collar Information In Google Earth

Because the individual maps that went into the mosaics and finally tiled image overlays in Google Earth required that their collars be clipped, vital legend information and other metadata specific to individual maps was lost. An effective solution I found to replicate much of this data was, for each site, a KML consisting of a single placemark – at the overlay's approximate centroid – and containing, among other things, legend information and any other historically interesting text. Because of various pictorial symbology used in the Bordner maps, it is only possible to replicate legend items consisting only of alphanumeric characters with this approach. Alternatively, one could incorporate a screen capture or crop of a given map's legend, hosted online, for reference in discerning the complex symbology scheme the maps use. To create this site-specific KML with all the important map information you want to incorporate, the free tool Excel to KML -- <http://www.earthpoint.us/ExcelToKml.aspx> -- is excellent and, if you have some knowledge of basic HTML formatting and tags, easy to use. This tool generates a KML from one or more placemarks or routes based on latitude and longitude coordinates, and an array of other properties, in an Excel spreadsheet. Follow the detailed instructions at the utility's homepage to create your metadata KMLs for each site.

Begin by selecting approximate centroid points on your individual map overlays in Google Earth and recording the coordinates displayed in the browser. Create an Excel spreadsheet for each overlay, named with a convention like *county\_township\_metadata.xls*. Each spreadsheet will have just one record, that is, one coordinate pair for the placemark location at the map's centroid. Follow the instructions on the Excel to KML page, and download the sample data (*ExcelToKmlDemo.zip*) from the link there for reference and examples in formatting your columns. Required columns in your spreadsheets are Latitude, Longitude, Name, Description,

and Icon. Legend information and any other text or links should be entered in the Description column. Use simple HTML tags here as follows:

line breaks - `<br>` or `<br/>`

bold – `<b>heading</b>`

italics -- `<i>text</i>`

Other columns I used included IconScale (1.5), Snippet (a simple description of the KML's purpose), and HideNameUntilMouseOver (False). Use your judgment; columns and metadata you include is determined by the story you want to tell with the areas of the Bordner Survey you are mapping.

With your sites' spreadsheets completed, save them and open them as KMLs in Google Earth via the Excel to KML webpage. Check for accuracy and fix if needed. When finished, right-click the layer names in the table of contents and save them out permanently as KMLs to the folders in which your image overlay tile directories are stored; be sure to copy them to your web server space as well.

Because each Bordner Survey area you have mapped will have both the image overlay KML and associated metadata/legend KML, consider grouping them in a single KMZ (zipped KML) in Google Earth to make loading and distributing your files easier. This can be accomplished by cutting and pasting each KML into a new folder you create in the table of contents (My Places or Temporary Places) and then save out as a KMZ. You can also nest your individual site KMZs into a single 'meta'-KMZ for even greater convenience as well as much reduced file size.

I followed the preceding methodology in mapping the one- to three-township areas I selected in Sauk, St. Croix, Trempealeau, Vernon, Vilas, and Waukesha counties. Readers will now have an insight into the workable process I developed with the crucial input of a number of people. If you have successfully followed the above steps, congratulations – you have contributed to giving the maps of the Bordner Survey a second life in easily distributable form in a modern geographic viewer!

### **Contact Information**

If you have any questions or comments regarding the Wisconsin Land Economic Inventory / Bordner Survey, my work with it, or the instructions in this tutorial, feel free to send me a message at [sfbjohnson@gmail.com](mailto:sfbjohnson@gmail.com).

## Credits and Acknowledgments

This project would not have been possible without the contributions of many people, with whom I conferred or collaborated to different extents over the course of the project, from conception to completion. They are as follows.

*Jerry Sullivan, Wisconsin DNR* – for introducing me to the Wisconsin Land Economic Inventory, pointing me in the direction of key informational and data resources as well as knowledgeable people, and providing ideas for how to approach the survey as a web mapping project

*AJ Wortley, State Cartographer's Office and project oversight* – for giving this project a home under the umbrella of the SCO, providing server space as well as a permanent presence on the Web, offering valuable and valued conceptual and technical expertise at critical points, and generally being available when needed

*Dr. Steven Ventura, UW Soils Department* – for advice early on about how to approach the Bordner Survey and ultimately getting me to seriously appraise the potential for Web cartography and modern re-use of the survey maps in georeferenced form

*Dr. Mark Harrower, Geography Department and project oversight* – for academic oversight of this project, continual generosity with its timeline, and enthusiasm and encouragement shown through its duration – it was very much appreciated!

*Karen Tuerk, UW GIS Certificate Program Manager* – for flexibility and generosity with the project's components and timeline, and for supporting an unconventional project with unconventional final reporting

*Peter Gorman and the staff of the UW Digital Collections Center* – for providing the image scans of the Bordner Survey maps used and for supporting new methods of re-using their historical holdings

*Jaime Stoltenberg, Robinson Map Library* – for material resources, providing a place to work productively, and constant support and encouragement of the project throughout – and sharing in the ups and downs experienced along the way

*Mitchell Moline, Wisconsin DOT* – for volunteering absolutely vital scripting and image processing knowledge and resources (directory structure, AML scripts, and instructions) at two key points and offering ideas regarding overall project direction and conceptualization, along with patient correspondence by email, chat, and other meetings when needed – drinks are on me, Mitch!

*Aaron Stephenson, GeoAnalytics* – for offering solutions regarding conversion and processing of some of the color map imagery used in this project

*Tom McClintock, Land Information and Computer Graphics Facility* – for taking an interest in my work (*at work*), working toward some crucial solutions with my Bordner imagery early on and helping to navigate the world of raster image types, TIFF world files, colormaps, and so on

*Jeff Maxted, UW Limnology Center* – for offering up, via the SIAC list, a simple technical suggestion that removed a huge hurdle preventing my moving forward in the final stages of the project – image conversion to remove colormaps and allow my Python script to work correctly – and thanks as well to all the folks on the SIAC and ESSA lists who offered their own suggestions and solutions (good to know a solid base of practical knowledge is out there!)

*Michael Edmonds, Wisconsin Historical Society* – for graciously providing permissions/clearance regarding WHS photo usage and hyperlinking to the Society website

*John Edwards, State Cartographer's Office student web developer* – for the fine job in getting my dedicated SCO project page for the Bordner Survey up and running in the home stretch

Thanks again to everyone who helped out in any way!

-SFBJ

