

D-Lib Magazine

September/October 2011
Volume 17, Number 9/10

Automating the Production of Map Interfaces for Digital Collections Using Google APIs

Anna Neatrou	Anne Morrow	Ken Rockwell	Alan Witkowski
The University of Utah	The University of Utah	The University of Utah	The University of Utah
anna.neatrou@utah.edu	anne.morrow@utah.edu	ken.rockwell@utah.edu	alan.witkowski@utah.edu

doi:10.1045/september2011-neatrou

Abstract

Many digital libraries are interested in taking advantage of the GIS mapping capabilities provided by Google Maps and Google Earth. The Digital Ventures Division of the University of Utah J. Willard Marriott Library has successfully completed an innovative automated process in which descriptive metadata in the form of place names was used to determine latitude and longitude coordinates for digital collection items. By enhancing digital collection metadata in this fashion, hundreds of records were updated without data entry from project staff. This article will provide an overview of using the Google application programming interface (API) to return geographic coordinate data, the scripting process with XML digital collection data, and the use of online tools and Microsoft Excel to upload digital collection data to Google Earth and Google Maps. The ability to automate metadata changes opens up a variety of possibilities for digital library administrators and collection managers.

Introduction

Many institutions and individuals have begun exploring the use of geographic data to provide an alternate means of accessing digital library collections. Providing additional search interfaces offers new ways for users to engage with digital library content. Understanding of collection items that reflect real world locations is enhanced when users have the ability to view items grouped by state, county or city. A significant value of geographic browsing for a digital library is assisting discoverability, particularly with collections containing large amounts of content. A digital library visitor can easily be overwhelmed by the prospect of browsing a collection of thousands of items distributed over dozens of browser windows. As a result, spatial browsing interfaces have become a popular solution because of their ability to help the digital library visitor access large volumes of content more efficiently.

At the [J. Willard Marriott Library](#), many of our digital collections have strong regional interest, and enhancing discoverability by allowing users to visually browse for materials in nearby counties, cities, or states provides an ideal way to showcase our unique collections. We partner with a variety of local institutions to host regional materials. For example, the [Utah State Historical Society](#) collections contain thousands of historical photos from the late nineteenth through the mid-twentieth century and [Uintah County Library's Photograph](#) collections contain several thousand historical images. The J. Willard Marriott Library's [Multimedia Archives](#) has more than 40,000 images. Due to the high volume of regionally-relevant content in our digital collections, developing a procedure for adding geographic coordinate data to digital content has been a strategic goal.

As we began to develop our strategy for incorporating geographic coordinates, we approached the problem from two different perspectives. First, we addressed incoming collections and content during the production phase in order to avoid retroactive collection editing. Our second strategy was to focus on preexisting collections and content with strong place name metadata. For new content and collections, we initially explored using a crosswalk to transfer coordinate data collected by image capture software to the content management system. This strategy ended up being inefficient and time-consuming, and applied only to images that originated from a digital source. Most of the materials added to our digital collections are historic, without the data that comes from photos captured with modern digital cameras. After investigating this option, we realized that it would be far more efficient to capitalize on the robust place name metadata in existing collections, and the solution could work equally well with incoming materials.

We explored the possibility of creating an automated process for looking up and returning latitude and longitude for place names. We already knew it was possible to modify and re-ingest our core collection metadata. In an experiment to take full advantage of the

automated processes available we decided to take the results of a geographic lookup script and add the coordinate data to new latitude and longitude fields. A program for manually adding coordinate data and exporting to third party programs like [Google Earth](#) and [Google Maps](#) was already well in place by the time we explored utilizing a script and place name metadata to capture coordinates. Once we had verified the opportunity to use an automated means of coordinate capture we were able to connect preexisting workflows for re-ingest, metadata export and map creation.

Sources consulted in preparation for this project included information that the Los Gatos library in New Mexico shared about using [CONTENTdm](#) and XML to create Google maps at a Western CONTENTdm users meeting.¹ The Mountain West Digital Library has also experimented with creating maps for alternate collection access, and we used templates created by the MWDL as models when preparing our own data.²

In an effort to create more robust geographic data for the collection, we developed a three step process:

- 1) Use the Google Geocoding API to return latitude and longitude data based on existing place names in the metadata.
- 2) Create a table and scripting program to add the new latitude and longitude values to the core metadata XML file within CONTENTdm.
- 3) Upload links to the digital collection items with the newly compatible latitude and longitude data to GoogleMaps.

While the second step in the process applies directly to institutions using CONTENTdm as the base software for their digital libraries, steps 1 and 3 can be completed by any digital library running any type of digital library software. This article is designed to provide a detailed overview of our workflow and software development process, so librarians working with application programmers can explore implementing a similar program with their own digital collections. The map generated with this process is available at <http://westernsoundscape.org/googleMap.php>.

The Geocoding Working Group at the J. Willard Marriott Library included a project manager, a metadata librarian in charge of quality assurance, a digital initiatives librarian who explored GIS products and processes, and a computer programmer who developed two customized programs to look up geographic data and add it to the digital collection.

Place Name Metadata and the Google Geocoding API

While the prospect of updating multiple records automatically through a scripting program offers the opportunity to save on staff time, preparing the collection for the scripting process represented an investment in quality assurance from the metadata librarian working on the project. For the first trial of this process, the Geocoding Working Group chose the [Western Soundscape Archive](#), a digital library of over 2,600 animal, natural, and manmade sounds with a regional focus on the American West.³ The Western Soundscape Archive has robust place name data, and the coordinates derived from these place names represent the approximate listening area and not the actual recording location.

Figure 1 shows part of a typical Western Soundscape Archive record with multiple place name metadata.

Common Name	American Avocet	controlled vocabulary
Scientific Name	Recurvirostra americana	controlled vocabulary
Kingdom	Animalia	controlled vocabulary
Phylum	Craniata	controlled vocabulary
Class	Aves	controlled vocabulary
Order	Charadriiformes	controlled vocabulary
Family	Recurvirostridae	controlled vocabulary
Genus	Recurvirostra	controlled vocabulary
Subject	Birds; Places; Nature sounds	controlled vocabulary
Place Names	Baylands Nature Preserve (Calif.); Palo Alto (Calif.); San Mat	

Figure 1: Sample Western Soundscape Archive item metadata.

The first custom program developed by the programmer is a Google Geocoding API Lookup script. To automate the task of geocoding, the scripting language PHP and the command-line tool cURL are used in conjunction with Google's Geocoding API⁴. First we export a list of unique place names. For digital libraries using software that supports import and export of collection data in XML files, the locations can be extracted easily with PHP's `preg_match` function, which is a regular expression matcher used to look for the

applicable xml tag, in our case "covspa."

The following is an example from WSA XML. "covspa" is the CONTENTdm nickname for the Dublin Core Element Coverage-Spatial.

```
<covspa>Pima County (Ariz.); Sonoran Desert (Ariz.); Arizona</covspa>
```

The Google Geocoding API Lookup script searches for all occurrences of <covspa>, reads the metadata, and breaks it up into distinct locations based on semicolons as a separator. Each location is put into an associative array that is later output into a comma-separated values (CSV) file. This spreadsheet is then manually reviewed for errors in the metadata. The second part of our script iterates through the location list, sending locations to the Google Geocoding API one at a time. This is done with the cURL library in PHP, which provides a mechanism for the API to transmit data using a variety of protocols, including automated HTTP requests. Google sends coordinates back if it finds a match. The coordinates are saved and then used to create a table populated with both the place names for the collection and their applicable geographic coordinates⁵.

Google Earth supports a variety of geographic notations⁶. After consulting various standards for geographic metadata⁷, we chose to use decimal degrees for our latitude and longitude data. The metadata librarian ranked the place names and coordinates, so we were able to assign the most specific latitude and longitude coordinates to items with multiple place names in their metadata. This ranking system is necessary to get the subsequent script to update the item with the most local and accurate coordinate data. Since we have multiple place names in records separated by semicolons, the scripting program populates the latitude and longitude fields with the most specific information first. This process would not be necessary for other library collections where items have only one place name assigned. See [Appendix Item 1](#) for the coordinate ranking system.

Using the latitude and longitude values to update the core XML file for the Western Soundscape collection

The second script is an XML Modification script which takes the table of coordinate pairs and collection place names returned by the Google Geocoding API lookup script and inserts them into the core descriptive metadata file for the collection. This is done by loading the table into an associative array in PHP. Each entry in the array contains the location, priority, and coordinates for fast look-up during the conversion process. Since only one coordinate pair is assigned per record, the locations are ranked and chosen by the highest priority number using the loaded table. If two locations have the same priority, then the first one is chosen.

Sample output:

```
<covspa>Pima County (Ariz.); Sonoran Desert (Ariz.); Arizona</covspa>
<latitu>32.057499</latitu>
<longit>-111.6660725</longit>
```

We first performed the operation on a test version of the collection on a test server. See [Appendix Item 2](#) for an outline of the process in a CONTENTdm collection. We found occasional issues with accuracy of the GoogleAPI's coordinate information, particularly in cases where the descriptive metadata listed a broad geographic area.

Uploading Revised Metadata to GoogleMaps

Once the new latitude and longitude coordinates are in the metadata for the collection, the next step is to use the updated metadata to generate a KML file⁸ that can be used in GoogleMaps applications. The display mechanism for a collection populated with latitude and longitude coordinates depends on the number of coordinate points in the collection. Google MyMaps has size limits that restrict KML file rendering⁹. This general process can be repeated for digital libraries run on software that supports the export of tab delimited collection metadata. The complete set of instructions that apply to a digital library running CONTENTdm is available in [Appendix Item 3](#).

Producing a KML file populated with objects housed in CONTENTdm begins with massaging exported item-level metadata in Excel. Basic descriptive metadata for each item, such as title, recording description, latitude, longitude, reference URL (item level link in CONTENTdm), and item number, are included in the Excel spreadsheet. We do not include the degree symbol in our geographic data in Excel, as early testing revealed that the default settings in Excel would result in errors parsing the symbol without additional edits. To comply with Google Maps requirements for objects, the latitude, longitude and description fields are edited and an icon field is added. Latitude and longitude cells are formatted up to six decimal places and the icon column is assigned the numerical code for the Google Maps icon¹⁰.

The most complex modifications involve object description. The original, exported description metadata field is renamed in the spreadsheet and a new description field is added in order to allow the object's description to have a broader set of features including

URL, thumbnail image and additional descriptive metadata. The method for combining these elements is to use a concatenation formula in Excel¹¹. Concatenation formulas allow merger and repetition of columns in a specified order.

	A	B	C	D	E	F	G	H	I
1	Description	Name	Place Names	Recording Description	Latitude	Longitude	Reference URL	CONTENTdm number	
2	=CONCATENATE(B2,C2,D2)	Western Sandpiper	Merced National Wildlife Refuge (Calif.); Merced County (Calif.); California	Audio tracks of animal sounds	37.181796	-120.647313	http://content.lib.utah.edu/u/?wss,1863	1863	

Figure 2: Creating a concatenation formula.

To generate thumbnails we add columns of script that, with exported item identification metadata, execute a command to generate hyperlinked thumbnails. In the formula we include additional descriptive metadata (place name, recording description) and add the persistent URL for the object in CONTENTdm. A final step involves adding a blank second row with a command to allow the Description column to exceed 256 characters.

The cells that are merged by the concatenation formula return formatted HTML that can be imported into Google Maps:

```
<a href="">  </a > <br> Audio tracks of animal sounds Merced National Wildlife Refuge (Calif.); Merced County (Calif.); California http://content.lib.utah.edu/u/?wss,1863
```

	A	B	C	D	E	F	G	H	I	J	K	
1	Icon	 	CONTENTdm number	> 	blank space	Description	Title	Place Names	Recording Description			
2								REPT("",256)				
3	187	 	1863	> 		Audio tracks of animal sounds Merced National Wildlife Refuge (Calif.); Merced County (Calif.); California	http://content.lib.utah.edu/u/?wss,1863	Western Sandpiper	Merced National Wildlife Refuge (Calif.); Merced County (Calif.); California	Audio tracks of animal sounds		
4	187	 	2900	> 		Audio tracks of animal sounds Indian Creek (San Juan County, Utah); San Juan County (Utah); Utah	http://content.lib.utah.edu/u/?wss,2900	Indian Creek crickets	Indian Creek (San Juan County, Utah); San Juan County (Utah); Utah	Audio tracks of animal sounds		

Figure 3: Thumbnail generation scripting, completed concatenation formula and blank row were added.

The formatted spreadsheet is converted to a KML file in a two-step process. In the first step, Earth Point's Excel to KML tool¹² and its View in Google Earth feature are used to open the spreadsheet within the Google Earth interface. As a result of the formatting in the spreadsheet, each row is represented as an individual pinpoint, georeferenced according to coordinate data. The pinpoint content includes the elements of the concatenation formula, in our case, a hyperlinked thumbnail, selected descriptive metadata and the

persistent URL. Next, the Google Earth file is saved as a KML file by using the option Save Place As in Google Earth. (Earth Point has a View File on Web Page option that is a useful quality control measure. Earth Point is available free of charge if working with 200 or fewer entries.)

For embedded maps using the Google Maps API, the KML file can be saved and imported directly into a webpage. If one wants to bypass the Google Maps API, one can import the map in "Google My Maps" and embed the HTML or post a link. Additional "Google My Maps" features are available for customizing the size and magnification. If your collection contains more than 200 geographic coordinates, you should experiment with display options using the [Google Embed KML gadget](#), which allows you to generate custom maps based on a KML file that you upload to a server.

Conclusion

Completing this process successfully has opened up a variety of possibilities at the J. Willard Marriott Library. We are currently reviewing other digital collections with place-name metadata in order to select additional candidate collections that can be enhanced with additional geographic coordinate data. At this time there are several map projects underway at the library, including a map containing georeferenced historic images from the Utah State Historical Society and a map based on the content of the Hidden Water collection which focuses on the surface water systems of the Salt Lake Valley.

While for the time being our efforts will continue to be directed towards replicating the Google mapping process on additional collections, we also plan to continue developing geographic interfaces. What we have set our sights on next are enhancing the maps by including temporal layers; aggregating content sharing spatial and temporal relationships from multiple collections in subject or theme-focused maps; and exploring options for implementing search, retrieval and recommender systems designed to operate within a geographic interface.

Acknowledgements

We would like to thank Sandra McIntyre for sharing her expertise with georeferencing metadata. John Herbert, Kenning Arlitsch, Jeff Rice, and Ann Marie Breznay provided valuable feedback on a draft of this paper. The Western Soundscape Archive was founded by Jeff Rice and Kenning Arlitsch and was supported by a major grant from the Institute of Museum and Library Services (IMLS), with matching funds from the University of Utah.

References

- [1] Bankhead, H (2009). "Historic Locations with Google Maps using XML from CONTENTdm," presentation at the Western CONTENTdm Users Group Meeting, University of Nevada-Reno, Reno, NV, June 5-6, 2009. <http://www.contentdm.org/USC/usermeetings/Bankhead.pdf>.
- [2] McIntyre, S (2009). "Utah Populated Places, with Links to MWDL Resources," Google map created October 16, 2009. Available 3 June 2011. <http://tinyurl.com/yfa3dp5>.
- [3] Arlitsch, K, et al (2008). "D-Lib Featured Collection November/December 2008: Western Soundscape Archive." <http://dx.doi.org/10.1045/november2008-featured.collection>.
- [4] Google Geocoding API documentation. Retrieved March 2, 2011 from <http://code.google.com/apis/maps/documentation/geocoding/>.
- [5] Geocoding Addresses with PHP/MySQL. Retrieved March 2, 2011 from <http://code.google.com/apis/maps/articles/phpsqlgeocode.html>.
- [6] Columns used by Excel to KML. <http://www.earthpoint.us/ExcelToKml.aspx>.
- [7] Cox, S, et al (2006). "DCMI Point Encoding Scheme: a point location in space, and methods for encoding this in a text string." <http://dublincore.org/documents/dcmi-point/>.
- [8] KML standard. <http://www.opengeospatial.org/standards/kml/>.
- [9] KML Support. <http://code.google.com/apis/kml/documentation/mapsSupport.html>.
- [10] Google Earth Icons. <http://www.earthpoint.us/ExcelToKml.aspx#GoogleEarthIcons>.
- [11] Concatenate Function, <http://office.microsoft.com/en-us/excel-help/concatenate-function-HP010342288.aspx?CTT=1>.

[12] Excel To KML – Display Excel files on Google Earth. <http://www.earthpoint.us/ExcelToKml.aspx>.

Appendix

Item 1: Coordinate Ranking System

- 1 = Very localized, such as a specific address within a city or a specific landmark (e.g., Temple Square, Delicate Arch, etc.)
- 2 = Locality, such as a town, city, or smallish region
- 3 = Larger region but within one county (or in the case of Yellowstone and Alaskan regions, no county)
- 4 = County
- 5 = Multi-county regions (e.g., National Forests, mountain ranges like the Cascades, etc.)
- 6 = States, Provinces
- 7 = Countries

Item 2: A general outline of the process in a CONTENTdm collection

- 1) Stop updates to collection, make it read only
- 2) Copy the desc.all file so conversion script can access it
- 3) Run conversion script
- 4) Replace the desc.all file in the collection
- 5) Run the makeoffset script
- 6) Run the full collection index
- 7) Remove read-only status from collection
- 8) Extract the collection metadata
- 9) Examine coordinates for accuracy
- 10) Build Google map with new coordinates

Item 3: Instructions for Converting CONTENTdm Exported Metadata to a KML file

- 1. In CONTENTdm
 - a. Export collection metadata as a tab-delimited txt file
- 2. Open text file in Excel and save as a workbook
- 3. Retain columns:
 - a. Title
 - b. Description
 - c. Place name
 - d. Latitude
 - e. Longitude
 - f. Reference URL
 - g. CONTENTdm item number
 - h. Any additional metadata fields you want included in the map
- 4. Edit column labels:
 - a. Title
 - i. Change to **Name**
 - b. Description
 - i. Change to **CDM Description**
- 5. Edit column format:
 - a. Latitude
 - i. Highlight, click right mouse button
 - ii. Format cells
 - iii. Number tab
 - 1. Number
 - 2. Decimal spaces = 6
 - b. Longitude
 - i. Highlight, click right mouse button
 - ii. Format cells
 - iii. Number tab
 - 1. Number
 - 2. Decimal spaces = 6

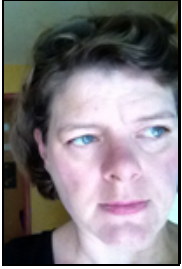
6. Insert columns:
 - a. Icon
 - i. Should be first column, before Name
 - ii. Enter icon number
 1. See <http://www.earthpoint.us/ExcelToKml.aspx#GoogleEarthIcons>
 - iii. copy and paste in column's rows
 - b. Description
 - c. Blank space
 - i. Enter a space in first row
 - ii. Copy and paste space into remaining rows
 - d. <a href="
 - e. ">

7. Insert blank row underneath column heading row
 - a. For this row only enter the following values:
 - i. Latitude
 1. Enter 00.00000
 - ii. Longitude
 1. Enter 00.00000
 - iii. Description
 1. Enter =REPT("",256)
 2. NOTE: This prevents the truncation of information in this column at 255 characters.
 - b. This row will generate a blue (default) icon in the final map which should be deleted before the map is made publicly accessible.
8. Edit row 3 of Description:
 - a. Add formula
 - i. =CONCATENATE(a3,b3,c3,etc.)
 - b. Cells listed in () should correspond with the metadata you want included in the map (e.g., CDM description, place name, thumbnail script).
 - c. Note: Name (i.e., Title) is added by default and does not need to be included in the formula.
 - d. Include the blank space referenced in Step 6c in the concatenation formula as needed to create spaces between combined columns of information, for example, the place names and the reference URL.
9. Save completed spreadsheet. Open <http://www.earthpoint.us/ExcelToKml.aspx>.
10. Browse to spreadsheet. Use view on web page to see if there are errors, otherwise use "view in Google Earth".
11. When Google Earth opens it loads the spreadsheet, which is now converted to a KML/KMZ.
 - a. The blank row (Step 7) will prompt an error message in Google Earth given the 00.00000 latitude and longitude values assigned to it (this entry was necessary to avoid truncation of descriptions). The resulting pinpoint in Google Earth should be deleted.
 - b. Go to File menu in Google Earth; select Save and Save Place As; save as a KML or KMZ to your hard drive.
12. Two methods for adding KML/KMZ to website:
 - a. Embed using Google Embed KML gadget
 - b. Use Google My Maps account and create new map to import retrieved HTML for embedding

About the Authors



Anna Neatrou works in digital projects at the University of Utah J. Willard Marriott Library and previously served as project manager for the Western Soundscape Archive. She is also the executive director of the Utah Library Association. She received her MLIS from the University of Illinois Urbana-Champaign.



Anne Morrow is a Digital Initiatives Librarian at the University of Utah J. Willard Marriott Library. Anne has a background in digital libraries, institutional repositories, metadata and cataloging. Most recently she has been involved in faculty publishing initiatives and was instrumental in developing the use of QR codes to connect mobile device users with library services and programs.



Ken Rockwell is an Associate Librarian who has worked at the J. Willard Marriott Library since 1989. His expertise is in metadata, including the cataloging of maps and description and subject analysis of digital collections. He has been providing metadata for the Western Soundscape Archive from its birth online and did geospatial quality control for the mapping project described in this article.



Alan Witkowski graduated from the University of Utah in 2006 with a degree in Computer Science. He has been at the J Willard Marriott Library's application programming division for two years.

Copyright © 2011 Anna Neatrou, Anne Morrow, Ken Rockwell, and Alan Witkowski

PRINTER - FRIENDLY FORMAT

[Return to Article](#)
