**Notes on running gazetteer code**

ews 6/1/2022

The latest iteration of the code is in Github: https://github.com/WildlifeConservationSocietyCI/welikia-production/

Local repository: C:\\_data\SCRIPTS\welikia-production

To run the code, Kim helped me set up a virtual environment using Anaconda, with the given environment file, and SSH configuration to access the Welikia db, which is on an Amazon EC2 server. Those steps are required only once; see notes at the end of this doc.

To run the code

1. Start up Bitvise by double-clicking on the profile file at C:\\_data\keys\ welikia\_db.tlp. This profile enables access to the Welikia db on an EC2 instance on AWS. After opening the profile, log in by clicking the “Log in” button at the bottom.
2. Open Anaconda Prompt from Start Menu
3. Activate the virtual environment “gazetteer” which is Python 3 with the needed modules/packages

(base) C:\Users\esanderson>conda activate gazetteer

1. Navigate to the welikia-production source code folder

(gazetteer) C:\Users\esanderson>cd c:\\_data\SCRIPTS\welikia-production\src

1. If any new places have been added or moved, then it's necessary to recalculate the "place csv" file that specifies the relationship between placename ids and plate/grid ids; see example: "C:\\_data\book\a Welikia Atlas\3 - gazetteer\plate\_grid\_ids\placename\_grid\_v7\_07052022.csv"

There are lots of ways to do the take attributes from the polygons (e.g. Plates\_Grids\_04132022.shp) and attach them to the points (e.g. Placenames\_v7.shp). Probably the easiest way in ArcGIS is to

1. select the Placenames\_v7.shp in the table of contents ,
2. right click, and select "Joins and Relates" then "Joins".
3. At the top of the dialogue, choose "Join data from another layer based on spatial location".
4. In the "choose a layer" dialogue, choose "Plate\_Grids\_04132022.shp"
5. In the "you are joining" dialogue, select the first choice "it falls inside"
6. Specify the output filename: (e.g.) C:\\_data\Welikia\_Version\_7.0\_Wonder\Vector\30km\_Extent\Placenames\_v7\_grid\_08142022.shp
7. Hit "ok"
8. Open attribute table of the joined file, make sure no records are selected, then save out the attributes as a dbf file: (e.g.) C:\\_data\book\a Welikia Atlas\3 - gazetteer\plate\_grid\_ids\placename\_grid\_v7\_08142022.dbf
9. Open dbf file in Excel
10. Eliminate all fields except: Place\_id, Plate2, Grid. Reorganize in that order (three columns with header). Sort by Place\_id.
11. Save out to csv: (e.g.) C:\\_data\book\a Welikia Atlas\3 - gazetteer\plate\_grid\_ids\placename\_grid\_v7\_08142022.csv
12. Note this csv file is hardcoded into the place\_synthesis.py on line 117. Make sure to update the filename for most accurate results!
13. Run the script, specifying the placenames shapefile and the folder with the markdown files, for example:

Shapefile = C:\\_data\Welikia\_Version\_7.0\_Wonder\Vector\30km\_Extent\Placenames\_v7.shp

Markdown directory = “C:\\_data\book\a Welikia Atlas\3 - gazetteer\second draft entries\bronx”

(note quotes are needed around the latter because of the spaces in the directory names)

(gazetteer) c:\\_data\SCRIPTS\welikia-production\src>python place\_synthesis.py C:\\_data\Welikia\_Version\_7.0\_Wonder\Vector\30km\_Extent\Placenames\_v7.shp "C:\\_data\book\a Welikia Atlas\3 - gazetteer\second draft entries\bronx"

Note that the markdown files are expected in this format [name].[placeid].[version].md, for example, Zborowski Point.4492.3.md. If you need to add the version number, use the “version\_copy\_entries.py” script to rename and copy to the hard-coded output directory.

Note that the output goes into the markdown file directory with a name and date stamp like this: gazetteer-entries\_2022-06-01.md

List of runs

7/5/2022

(gazetteer) c:\\_data\SCRIPTS\welikia-production\src>python place\_synthesis.py C:\\_data\Welikia\_Version\_7.0\_Wonder\Vector\30km\_Extent\Placenames\_v7.shp "C:\\_data\book\a Welikia Atlas\3 - gazetteer\third draft entries\bronx"

8/14/2022

(gazetteer) C:\\_data\SCRIPTS\welikia-production\src>python place\_synthesis.py C:\\_data\Welikia\_Version\_7.0\_Wonder\Vector\30km\_Extent\Placenames\_v7.shp "C:\\_data\book\a Welikia Atlas\3 - gazetteer\third draft entries\jamaica bay"

Setting up virtual environment

Following video at <https://www.youtube.com/watch?v=6kXLUvsnhuI> and Kim’s advice

1. Open Anaconda Navigator once just to get the configuration right
2. Open Anaconda Prompt from Start Menu
3. Create the virtual environment giving it a name, here “gazetteer”

(base) C:\Users\esanderson>conda create --name gazetteer python=3

Note: Anaconda goes and gets the version of Python specified, here it’s the latest version of python 3.

1. Activate the virtual environment

(base) C:\Users\esanderson>conda activate gazetteer

Note that the prefix in parentheses is now the name of the virtual environment

1. Navigate to welikia-production folder and then install the needed modules

(gazetteer) C:\Users\esanderson>cd c:\\_data\SCRIPTS\welikia-production

(gazetteer) c:\\_data\SCRIPTS\welikia-production>pip install -r requirements.txt

Collecting mysql-connector==2.2.9

Using cached mysql-connector-2.2.9.tar.gz (11.9 MB)

Collecting pathlib2==2.3.5

Using cached pathlib2-2.3.5-py2.py3-none-any.whl (18 kB)

Collecting python-dotenv==0.18.0

Downloading python\_dotenv-0.18.0-py2.py3-none-any.whl (18 kB)

Collecting six

Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)

Building wheels for collected packages: mysql-connector

Building wheel for mysql-connector (setup.py) ... done

Created wheel for mysql-connector: filename=mysql\_connector-2.2.9-cp310-cp310-win\_amd64.whl size=247961 sha256=a5b58a8344dc092ce9269bfab58094406cd8b9bf32be56880151aa5bc8ecc13f

Stored in directory: c:\users\esanderson\appdata\local\pip\cache\wheels\76\48\9b\da67ff1a18fe8e9d428f9b1a177716d4a7d363d2bbe83bf6cf

Successfully built mysql-connector

Installing collected packages: six, python-dotenv, pathlib2, mysql-connector

Successfully installed mysql-connector-2.2.9 pathlib2-2.3.5 python-dotenv-0.18.0 six-1.16.0

Providing environment variables for database

The .env file specifies important environment variables regarding accessing the database. It’s contents are listed below. This file lives in C:\\_data\SCRIPTS\welikia-production.

DBHOST=127.0.0.1

DBPORT=3307

DBNAME=welikia\_sdr

DBUSER=root

DBPASS=SFv7urYw2bLuUfZQ76CqedtFxn2hKrHZ

Digging an SSH tunnel

Finally Kim helped me set up a profile using [Bitvise client](https://www.bitvise.com/ssh-client-download) that handles the SSH tunnel encryption between the local machine and the Amazon Web Services hosted database.

Host welikia-net

HostName 107.22.180.50

User ec2-user

IdentityFile ~/Documents/welikia/2409/website/bitnami/M2409live.pem

LocalForward 3307 localhost:3306

On the “Login” tab, use the host IP address and user name above. The Authentication mode is “publickey”.

Import the client key using the “client key manager” and assigning it to “Profile 1”.

Locally the key code is found in C:\\_data\keys\ M2409live.pem

On the “C2S” tab, add the listen interface (my local machine) as 127.0.0.1 with port 3307. On the remote database, the destination is “localhost” with port 3306.

The saved profile is C:\\_data\keys\ welikia\_db.tlp