Water Allocation Tool – User Instructions

TOOL OVERVIEW

This document is intended to provide the necessary information to implement the Division of Water Rights Allocation Tool (DWRAT), previously called the Drought Water Rights Allocation Tool. The Tool is an optimization program that allocates stream flow according to physical availability, and water rights law, using a series of integrated mathematical equations. It is open source and purely quantitative, and users collect or provide all of the data required to implement tool. Most of the theoretical framework, and its first iteration, were developed by the UC Davis Center for Watershed Sciences in a project funded by the State Water Resources Control Board (State Water Board) during the drought beginning in 2014. The Tool has since been improved and implemented to issue curtailments of water use in the Russian River Watershed during the 2021 drought.

AVAILABLE BACKGROUND DOCUMENTS

In addition to these instructions, Tool users or those interested will find the following resources available.

Academic Papers and Journal Articles

There are numerous academic papers summarizing work done for graduate theses, that were the first early evaluations of allocations and exploration of different data sets, in various watersheds. A collection of these papers is available here. Additionally, a journal publication is available here.

Formulations

The formulations contain the mathematics behind the derivation of the variables within the constraints, objective functions, and other equations. Use of the Tool does not require advanced mathematical training or understanding of the formulations, but they will prove useful to some. They are available on the GitHub page here.

GitHub Page

A GitHub repository for DWRAT is located on the GitHub page of the California Water Board Data Center under DWRAT or https://github.com/CAWaterBoardDataCenter/DWRAT. This page contains all the input and output data for the implementation of the Tool in the Lower Russian River Watershed. This is also the location of the most current codebase.

Script Comments

Comments and annotations are contained within the scripts themselves that elaborate or explain many of the matrix operations and many of the programming steps. The main code is on the GitHub page and viewable here.

SOFTWARE REQUIREMENTS & RECOMMENDATIONS

Anaconda Individual Edition

- The water allocation tool and supporting scripts are modeled in the Python computer programming language. Python is widely used in scientific computing and data analysis. It is open source and free to download. In addition to Python, an editor or an integrated development environment (IDE) will be needed to change some settings within the scripts.

- The recommended way to install Python, and other required software libraries is with **Anaconda Individual Edition**, a cross-platform Python distribution. Anaconda will work on Windows, macOS, or Linux and is free to download. Anaconda is used to download the required software packages below, although these can be obtained by other means.
    - Here are the **instructions for downloading Anaconda**. Anaconda includes two editors that may be used to run and manipulate the scripts.
        - **Jupyter Notebook**, a web-based programming environment, and
        - **Spyder**, the Scientific Python Development Environment is also included. Instructions for running the tool will be for the Spyder IDE.

Required Python Software Libraries

- **Pandas** is an open-source data manipulation and analysis software library built upon Python and NumPy. It is included with Anaconda.
- **PuLP** is a Linear Program modeler and solver package and can call many different algorithms to solve linear problems.
    - **Download PuLP** with Anaconda Powershell by running `conda install -c conda-forge pulp`
- **NumPy** is an open-source numerical Python library used for manipulating arrays and matrices, and includes many mathematical functions. It is included with Anaconda, however some functions may be edited over time between versions. For example, arrays may be broadcast differently between versions, which can return an error for function matmul().
    - **Set NumPy version**. After you have downloaded Anaconda there will be a folder with several programs. Open Anaconda Powershell and run: `conda install numpy=1.19.2` (for example) or another release, then select y if prompted. To determine if numpy 1.19.2 has installed successfully you should run >>> **np.__version__** in the Python console.

The Spyder IDE

Included within the Anaconda distribution is a recommended program, called Spyder. Spyder is a programming environment designed for scientific and mathematical analysis and script development. It includes an integrated Python console, a variable explorer, and a Python script editor. An editing program is required to use one of the supplemental Python scripts in the codebase that will most likely save considerable time. An editor will most likely prove very useful. More information on Spyder here.

Creating a Spyder Project

- Tool users are not required to use Spyder to run the programs, or create a Spyder Project if they do. However, by creating a Spyder Project one can browse through project files more easily, and the root folder will be added to the PYTHONPATH which can avoid confusion.
    - Open a Spyder window and select the Projects menu, New Project…, Existing Directory.
    - Browse to your project folder and click Create. This folder will now be available in the Projects menu by selecting Open Project.
- The code color scheme can be selected using the Tools menu, Preferences, Syntax highlighting theme.

- Various window configurations are available by selecting the View menu, Window layouts, and selecting among the options. The R studio layout is convenient for exploring output tables and individual variables in the editor. This layout will contain the **editor** on the upper left, the **console** on the lower left, a **variable explorer** on the upper right, and a **file explorer** on the lower left.

Geographic Information Systems

- The analyst will likely need, or want to make use of a GIS software program such as ArcGIS or the open-source QGIS. The instructions in Appendix A make use of Esri's ArcGIS.

REQUIRED DATA

As discussed, the Water Allocation Tool is only a mathematical framework implemented with computer programming. The resulting allocations will be only as robust as the input data. The data requirements are as follows:

Water Users Attributes

- Water user demand in volume per unit time
- Riparian or appropriative water right designation for all users
- Seniority rankings for appropriative and pre-1914 right holders

Water Supply and Subbasin Attributes

- Defined subbasin delineations
- Subbasin unimpaired flow (supply, water availability, etc.) in volume per unit time
- Subbasin flow path, and downstream subbasin

Derived Data Sets

- Riparian / Appropriative User Matrices
    - Matrix indicating the user's subbasin location
- Basin Connectivity Matrix
    - Matrix indicating downstream or upstream connectivity between all subbasins.
- Riparian / Appropriative User Connectivity Matrices
    - Matrix indicating downstream or upstream connectivity between all users.

Subbasin Delineations & GIS

Subbasins delineations will be required to estimate flow, so these are often defined independently of DWRAT implementation. There are no spatial data inputs required for the tool. Spatial data manipulation is not strictly necessary if water users' subbasin locations are known, however this will more likely be a geoprocessing step in the procedure. It is also likely that a spatial representation will prove very useful.

Separate instructions are provided for Geoprocessing / GIS steps to create, refine, or otherwise process DWRAT polygons, and to further subdivide subbasins into headwaters and mainstem areas (discussed in ADDITIONAL CONSIDERATIONS below). See Appendix A following this document.

DWRAT IMPLEMENTATION PROCEDURES

The Division of Water Rights conducted an extensive effort to apply quality assurance and corrections to the values of monthly water use as required to be reported by Russian River water users in annual reports. The documentation and procedures for this process are available. Likewise, modeled flows were developed for both the Lower and Upper Russian Watersheds. The documentation and procedures for this process are available.

Once good data have been collected, there are broadly three main tasks to complete in the following order to implement DWRAT.

DATA INPUT PROCESSING

Three tables in .csv format are the first inputs that are required to be in the input subfolder. The data fields and formatting instructions are as follows.

Flows

Flows for each subbasin should be contained in a file called **flows.csv** It needs the following fields:

- BASIN – the unique ID for each subbasin, in text format, and sortable as text. If basins are numbered, it's recommended to add a letter prefix, and keep all the basin numbers the same number of digits. For example, B_01, B_02 …B_10, not B_1, B_2 … B_10. This will avoid sorting difficulties later on.
- FLOWS_TO – the subbasin ID representing the downstream basin.
    - The outlet subbasin for the watershed flows to itself, so it will have its own subbasin ID for FLOWS_TO
- Supply flow fields labelled according to their date.
    - The preferred date format is yyyy-mm-dd or yyyy-mm for monthly data (although other formats can be accommodated). Note that opening .csv files containing date fields in Excel may lead to the dates being automatically reformatted, make sure to check that they remain in the format you prefer by either manually reverting the field before saving (you may have to set it to text format) or by accessing the file through a different program.

It should look like **Figure 1** below.

	A	B	C	D	E	F	G
1	BASIN	FLOWS_TO	2021-05	2021-06	2021-07	2021-08	2021-09
2	U_01	U_04	1826.90	504.36	245.26	79.29	48.74
3	U_02	U_03	2536.50	626.62	259.74	73.20	41.19
4	U_03	U_04	273.01	64.67	28.53	8.63	4.95
5	U_04	U_05	1264.36	419.52	198.16	64.07	37.91
6	U_05	U_06	2205.88	523.58	200.39	59.18	32.36
7	U_06	U_09	3836.13	933.21	427.89	119.24	65.32
8	U_07	U_08	329.95	65.97	31.59	10.08	5.50
9	U_08	U_09	1632.27	368.76	176.71	56.80	32.62
10	U_09	U_10	2234.38	541.12	193.57	53.17	28.96
11	U_10	U_12	897.12	229.00	97.77	29.53	16.77
12	U_11	U_12	1466.01	364.75	130.27	40.45	19.68
13	U_12	U_13	1293.85	334.18	145.62	63.07	30.82
14	U_13	U_13	26.95	12.50	7.32	2.48	1.50

Figure 1. Example of flows.csv table

Riparian Demand

Demand for each riparian user should be contained in a file called **riparian_demand.csv** It needs the following fields:

- USER – the unique ID for each riparian user
- BASIN – the unique ID in which that riparian user is located
- Water demand fields labelled by date.
    - As above, the preferred date format is yyyy-mm-dd or yyyy-mm for monthly data (although other formats can be accommodated)

It should look like **Figure 2** below.

	A	B	C	D	E	F	G
1	USER	BASIN	2021-05	2021-06	2021-07	2021-08	2021-09
2	S017751	U_01	0.51	0.45	0.00	0.65	0.48
3	S022410	U_13	0.26	0.28	0.39	0.64	0.17
4	S024481	U_13	0.16	0.27	0.43	0.61	0.16
5	S021321	U_01	0.15	0.58	1.01	0.59	0.23
6	S015790	U_06	0.5	0.53	0.57	0.57	0.52
7	S014865	U_01	0.53	0.53	0.53	0.53	0.53
8	S014864	U_01	0.53	0.53	0.53	0.53	0.53
9	S014861	U_01	0.66	0.66	0.53	0.53	0.53
10	S016911	U_02	0.00	0.7	0.89	0.52	0.49
11	S026934	U_06	0.48	0.48	0.48	0.48	0.48
12	S009525	U_05	0.1	0.29	0.42	0.43	0.38
13	S027851	U_12	0.00	0.25	0.33	0.42	0.69
14	S023897	U_06	0.34	0.34	0.4	0.4	0.06
15	S022803	U_06	0.1	0.1	0.3	0.4	0.4
16	S024188	U_02	0.00	0.36	0.36	0.36	0.36
17	S023711	U_06	0.35	0.35	0.35	0.35	0.35
18	S022335	U_02	0.17	0.55	0.62	0.35	0.03
19	S019924	U_01	0.00	0.33	0.00	0.33	0.00
20	S006491	U_09	0.33	0.33	0.33	0.33	0.33
21	S026123	U_13	0.08	0.5	0.4	0.29	0.17
22	S017040	U_13	0.36	0.41	0.37	0.29	0.36
23	S022409	U_13	0.17	0.29	0.38	0.28	0.29

Figure 2. Example of riparian_demand.csv table

Appropriative Demand

Demand for each appropriative user should be contained in a file called **appropriative_demand.csv** It needs the following fields:

- USER – the unique ID for each appropriative user
- BASIN – the unique ID in which that appropriative user is located
- PRIORITY_DATE – a date indicating the seniority of the water right.
- PRIORITY_RANK – a ranking of seniority from 1 to the number of appropriative users.
- Water demand fields labelled by date.
    - As above, the preferred date format is yyyy-mm-dd or yyyy-mm for monthly data (although other formats can be accommodated)

It should look like **Figure 3** below.

	A	B	C	D	E	F	G	H
1	USER	BASIN	PRIORITY	2021-05	2021-06	2021-07	2021-08	2021-09
2	A021429C	U_01	463	0.45	0.45	0.45	0.45	0.45
3	A028787	U_01	893	6.76	1.10	0.43	0.00	0.00
4	S020096	U_01	33	0.41	0.40	0.41	0.41	0.40
5	A024774	U_01	683	0.00	0.41	0.41	0.41	0.41
6	A023835	U_06	599	0.22	0.39	0.39	0.39	0.39
7	A017240	U_01	353	0.00	0.33	0.33	0.33	0.33
8	A023707	U_01	590	0.33	0.33	0.33	0.33	0.00
9	D030985	U_01	1023	0.33	0.33	0.33	0.33	0.33
10	A022070	U_09	498	0.83	0.33	0.33	0.33	0.33
11	D031005	U_11	1017	0.25	0.27	0.33	0.39	0.37
12	D032762	U_01	1141	0.00	0.31	0.31	0.31	0.31
13	S015196	U_11	16	0.06	0.14	0.18	0.31	0.33
14	A024931	U_13	718	0.18	0.18	0.18	0.18	0.18
15	S018456	U_01	49	0.05	0.08	0.11	0.10	0.06
16	A022614	U_08	519	0.09	0.09	0.09	0.09	0.09
17	S000114	U_09	71	0.13	0.08	0.09	0.09	0.08
18	S017301	U_11	14	0.06	0.06	0.08	0.09	0.08
19	A028715	U_01	890	0.07	0.07	0.07	0.07	0.31
20	S018523	U_01	51	0.04	0.05	0.06	0.07	0.05
21	S015198	U_11	46	0.05	0.05	0.06	0.06	0.05
22	A016141	U_08	310	0.05	0.05	0.05	0.05	0.05
23	S000115	U_09	72	0.09	0.05	0.05	0.05	0.05

Figure 3. Example of appropriative_demand.csv table

SETTING UP A PROJECT FOLDER

The DWRAT tool was designed for users with very minimal coding experience. However, the modules are implemented in the Python programming language. The tool relies on reading in the user's input data for manipulation, and outputting. If the standardized project folder structure is followed, the path names should not need to be edited in the code itself. The elements and structure of the project folder may be obtained by cloning a GitHub repository or by manually setting it up.

Cloning the Repository with GitHub

The Lower Russian River DWRAT code and input files are hosted on the California Water Board Data Center GitHub page here: https://github.com/CAWaterBoardDataCenter/DWRAT.

- Downloading a Zip folder: Clone the repository either on the GitHub page above by selecting code -> download zip folder and extract the files in the folder.
- Cloning the Repository with Git:
  - Alternatively, use Git. Open a powershell window, and change directory to the folder above.
  - Run the following command: git clone https://github.com/CAWaterBoardDataCenter/DWRAT

You will now have the input and output folder and all the required scripts.

None of the existing data for the Lower Russian River in the repository is required. It is not necessary to delete anything either as the required files will be over-written by the Tool. However, the existing data and subfolders provide a useful template and example formats for the inputs.

Manually Assembling the Project Folder

- Create a project root folder in your Users directory – usually C:\Users\yourname:
  - C:\Users\yourname\project_folder
- Download the scripts individually on the GitHub page to the root folder.
  - **appropriative_user_matrices.py**
  - **basin_connectivity_matrix.py**
  - **main_date_range.py**
  - **riparian_user_matrices.py**
  - **water_allocation_tool_2021.py**
- Within this project folder create two additional folders with the (exact) names input and output:
  - C:\Users\yourname\project_folder\input
  - C:\Users\yourname\project_folder\output
- Create any folders and sub-folders containing raw data, geospatial data, or other relevant files as desired.

It should look like **Figure 4** below.

Name	Date modified	Type
input	12/3/2021 6:29 PM	File folder
output	12/3/2021 6:29 PM	File folder
appropriative_user_matrices.py	9/10/2021 9:05 AM	PY File
basin_connectivity_matrix.py	10/29/2021 12:16 PM	PY File
main_date_range.py	9/10/2021 9:05 AM	PY File
riparian_user_matrices.py	9/10/2021 9:05 AM	PY File
water_allocation_tool_2021.py	10/29/2021 12:16 PM	PY File

Figure 4. Example of Project folder after assembly before the next step

ADDITIONAL CONSIDERATIONS

Depending on the conditions in which the Tool is being used, there are additional considerations for processing and potentially modifying the input data. These may not be necessary or appropriate under particular geographic, management, or legal contexts. Furthermore, they will depend on the actual physical characteristics of the watershed, the water users, and the supply flow regime, and the resolution of the data. Neither is this list comprehensive. Other factors may be far more important to consider, for example reservoirs releases or uncertain groundwater-surface water interactions.

Subdivision of Subbasins and User Basin Assignments

One of the key assumptions in the DWRAT framework is that all water users have access to all of the available flow, or flow as predicted or measured at the subbasin outlet, in practical terms. This may or may not be reasonable depending on the actual location of users' points of diversion, the resolution of subbasin delineations and flows, and whether the available flow is from accumulation of upstream basins or evenly available through a network of tributaries. These attributes will need to be evaluated for each specific watershed. If significant user demand is physically located in the upper reaches of large, coarsely delineated subbasins, then it is recommended to further subdivide subbasins into areas where users only have access to subbasin inflow, and areas where users have physical access to any accumulated flows from upstream basins. These areas can be thought of as headwaters or mainstem.

In terms of processing the input data, this simply means assigning users either to the headwaters or the mainstem of each basin, and adding new subbasin representation. This technique doesn't require any additional input data, but it does require professional discretion to define the boundaries of headwaters and mainstem areas.

- Appendix A, step 5 provides detailed instructions. Once they are successfully completed you should have all users designated as headwaters or mainstem with a field indicating which one, for any basins where this is being implemented.
- For each subbasin being subdivided, a new BASIN ID will be required. This can be done simply in GIS or in Excel by making use of the mainstem Y/N field. Follow the suggested naming convention above, and add a suffix for headwaters and mainstem subbasins, but keep the numbering the same. For example, B_01_HW, and B_01_MS.
- Once the **riparian_demand.csv** and **appropriative_demand.csv** files have been updated with the correct new BASIN ids, the **flows.csv** table will need to be modified as well.
- Updating the flows table is conceptually straightforward. All new subbasins need their ids added under the BASIN field.
- All subbasin flow starts in the headwaters basin and flows to its mainstem counterpart. Mainstem basins only flow to the next downstream mainstem basin. Carefully update the FLOWS_TO field in the **flows.csv** table accordingly.
- Next, enter 0 for all mainstem subbasin flows as they do not have direct flow inputs outside of the headwaters inputs, and keep the flow values in the headwater subbasin or the undifferentiated ones.

See **Figure 5a** below a "simple" flows.csv tables and **Figure 5b** modified versions and note *not all subbasins have been subdivided.*

	A	B	C	D	E	F	G
1	BASIN	FLOWS_TO	2021-05	2021-06	2021-07	2021-08	2021-09
2	U_01	U_04	1826.90	504.36	245.26	79.29	48.74
3	U_02	U_03	2536.50	626.62	259.74	73.20	41.19
4	U_03	U_04	273.01	64.67	28.53	8.63	4.95
5	U_04	U_05	1264.36	419.52	198.16	64.07	37.91
6	U_05	U_06	2205.88	523.58	200.39	59.18	32.36
7	U_06	U_09	3836.13	933.21	427.89	119.24	65.32
8	U_07	U_08	329.95	65.97	31.59	10.08	5.50
9	U_08	U_09	1632.27	368.76	176.71	56.80	32.62
10	U_09	U_10	2234.38	541.12	193.57	53.17	28.96
11	U_10	U_12	897.12	229.00	97.77	29.53	16.77
12	U_11	U_12	1466.01	364.75	130.27	40.45	19.68
13	U_12	U_13	1293.85	334.18	145.62	63.07	30.82
14	U_13	U_13	26.95	12.50	7.32	2.48	1.50

Figure 5a. Example of "simple" or undifferentiated flows.csv table.

	A	B	C	D	E	F	G
1	BASIN	FLOWS_TO	2021-05	2021-06	2021-07	2021-08	2021-09
2	U_01	U_04_MSRR	1826.90	504.36	245.26	79.29	48.74
3	U_02	U_03_MSRR	2536.50	626.62	259.74	73.20	41.19
4	U_03	U_03_MSRR	273.01	64.67	28.53	8.63	4.95
5	U_03_MSRR	U_04_MSRR	0.00	0.00	0.00	0.00	0.00
6	U_04	U_04_MSRR	1264.36	419.52	198.16	64.07	37.91
7	U_04_MSRR	U_05_MSRR	0.00	0.00	0.00	0.00	0.00
8	U_05	U_05_MSRR	2205.88	523.58	200.39	59.18	32.36
9	U_05_MSRR	U_06_MSRR	0.00	0.00	0.00	0.00	0.00
10	U_06	U_06_MSRR	3836.13	933.21	427.89	119.24	65.32
11	U_06_MSRR	U_09_MSRR	0.00	0.00	0.00	0.00	0.00
12	U_07	U_08	329.95	65.97	31.59	10.08	5.50
13	U_08	U_09_MSRR	1632.27	368.76	176.71	56.80	32.62
14	U_09	U_09_MSRR	2234.38	541.12	193.57	53.17	28.96
15	U_09_MSRR	U_10_MSRR	0.00	0.00	0.00	0.00	0.00
16	U_10	U_10_MSRR	897.12	229.00	97.77	29.53	16.77
17	U_10_MSRR	U_12_MSRR	0.00	0.00	0.00	0.00	0.00
18	U_11	U_12_MSRR	1466.01	364.75	130.27	40.45	19.68
19	U_12	U_12_MSRR	1293.85	334.18	145.62	63.07	30.82
20	U_12_MSRR	U_13_MSRR	0.00	0.00	0.00	0.00	0.00
21	U_13	U_13_MSRR	26.95	12.50	7.32	2.48	1.50
22	U_13_MSRR	U_13_MSRR	0.00	0.00	0.00	0.00	0.00

Figure 5b. Example of modified flows.csv table.

Treatment of Null or Zero User Demand Values

Some analytical contexts may require differentiating among any zero allocation results, especially for appropriative diversions. This can occur when water users with no demand are included in the input data, which may be advantageous for a number of reasons. One reason might be simply to keep a continuous record count through a larger analysis. But more significantly if the Tool will be used in a curtailment or implementation context, one may need to know whether an allocation is zero due to zero demand, or whether there is zero water availability. One approach is to replace any zero demand values with an identifying, miniscule amount. The amount should be small enough so as to be rounded off below the reporting number of significant figures. Users who would otherwise have water available to them, will receive this miniscule allocation in contrast to those who would not. 0.00099 acre-feet per month was used in the Russian River.

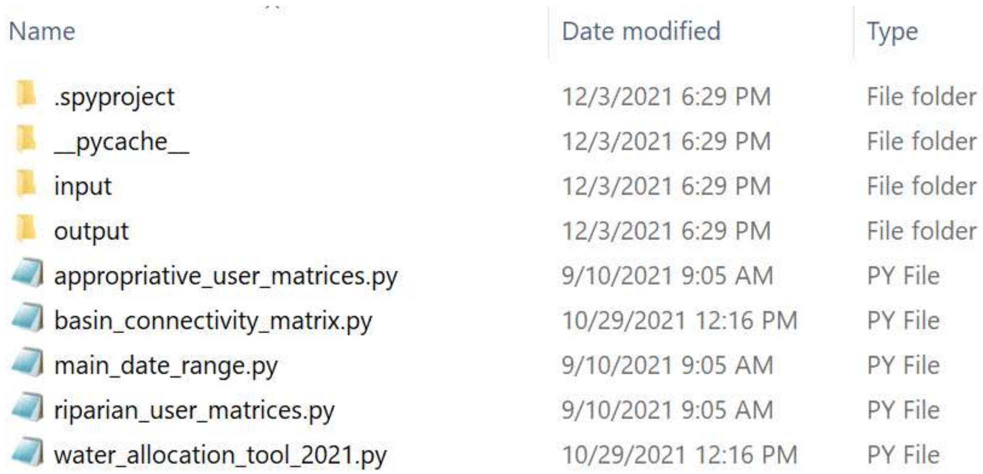
Treatment of Conditional User Priorities

California water rights can be quite complex, and though rare, there can be situations where a right to divert water under a license is dependent on factors other than simple seniority. It may be possible to address this situation through manipulation of the appropriative users' PRIORITY values, or manually running tailored scenarios. However, there can be situations where certain allocations will need to be solved outside of the allocation tool analysis.

DERIVING ADDITIONAL MATRICES

Required Connectivity Matrices

This step will require running three of the Python scripts above using your data. There are many options for using and implementing Python scripts, but Spyder is the suggested method, and creating a Spyder Project is also highly recommended. Instructions above. Once a Spyder Project has been create the folder should look like **Figure 5** below.



Name	Date modified	Type
.spyproject	12/3/2021 6:29 PM	File folder
__pycache__	12/3/2021 6:29 PM	File folder
input	12/3/2021 6:29 PM	File folder
output	12/3/2021 6:29 PM	File folder
appropriative_user_matrices.py	9/10/2021 9:05 AM	PY File
basin_connectivity_matrix.py	10/29/2021 12:16 PM	PY File
main_date_range.py	9/10/2021 9:05 AM	PY File
riparian_user_matrices.py	9/10/2021 9:05 AM	PY File
water_allocation_tool_2021.py	10/29/2021 12:16 PM	PY File

Figure 5. Example of Project folder after creating a Spyder Project

Constructing the Basin Connectivity Matrix

- First open the **basin_connectivity_matrix.py** so that it will appear in the editor. This script calls the **flows.csv** for the FLOWS_TO data.
- **IMPORTANT** – The script will need to be modified to specify the watershed outlet basin, otherwise the while loop will never end.
- There are comments that indicate where to do this in the script. If your basin ids are text, you will need to surround the text with quotes " ".
- Run the script with the F5 key or the Run menu.
- Once this script runs successfully, you should have the **basin_connectivity_matrix.csv** file in the input folder. This file will be used in the following routines.

Constructing the User Location and Connectivity Matrices

- The next step is to proceed with running **appropriative_user_matrices.py** and **riparian_user_matrices.py**.
- These scripts call each of **riparian_demand.csv** and **appropriative_demand.csv**, so they must be complete in the standard format.
- Run the code with the F5 key or the Run menu.
- Once this script runs successfully, you should have the additional four files above.

Data Range (Optional)

- The final input file needed is **data_range.csv**. This file tells the tool what time-series of data will be used, and needs to be set whether all the data is being used or not.
  - The main script **water_allocation_tool_2021.py** calls a function date_string defined in **main_date_range.py** and writes the output to **data_range.csv** within the input folder. You will need to set this range (making sure you have data for the days selected).

The input subfolder should now look like **Figure 6** below.

Name	Date modified	Type
appropriative_demand.csv	7/15/2021 11:24 AM	Microsoft Excel C...
appropriative_user_connectivity_matrix.csv	7/15/2021 10:57 AM	Microsoft Excel C...
appropriative_user_matrix.csv	7/15/2021 10:57 AM	Microsoft Excel C...
basin_connectivity_matrix.csv	7/15/2021 10:57 AM	Microsoft Excel C...
data_range.csv	7/22/2021 10:48 AM	Microsoft Excel C...
flows.csv	7/14/2021 12:34 PM	Microsoft Excel C...
riparian_demand.csv	7/15/2021 11:25 AM	Microsoft Excel C...
riparian_user_connectivity_matrix.csv	7/15/2021 10:57 AM	Microsoft Excel C...
riparian_user_matrix.csv	7/15/2021 10:57 AM	Microsoft Excel C...

Figure 6. Example of the <u>input</u> folder after derivation of the additional matrices.

RUNNING THE TOOL

*Running the Code (***water_allocation_tool_2021.py)**

- The script requires that a data range and format be selected.
- Follow the steps in the comments to do this, before running the tool.
- The code may be run with the option in the Run menu, or with the F5 key shortcut. Highlighted sections of the code may be run with the F9 key shortcut. Double clicking a variable and pressing F9 will highlight then output the variable in the console if it has been defined.
- The script displays the value of Objective functions, whether there is flow available after riparian allocations, a counter, the date being processed, and whether the status of the allocation is optimal.
- Riparian basin proportions and appropriative allocations are not displayed during execution to increase speed. However, they may be displayed in the console output by uncommenting the various print() statements (remove the # preceding the code).

Output Files

- The riparian basin proportions and appropriative allocations will be output in the folder named output.

APPENDIX A

Draft DWRAT Geoprocessing Steps

This document describes the recommended steps to:

1. Create a basins polygon layer for input into the DWRAT modeling tool
2. Designate mainstem water rights, if necessary

This process was created using ArcGIS Pro 2.6.2, however the general steps will easily translate to other GIS software.

Geoprocessing Steps:

COMING SOON