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 quantitive, 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 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.

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 here.

GitHub Page

A GitHub repository for DWRAT is located on the GitHub page of the California Water Board Data Center here. 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 programming 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 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 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. The following Numpy versions are suggested.
    - **NumPy version 18.1.1**. After you have downloaded Anaconda there will be a folder with several programs. Open Anaconda Powershell and run: `conda install numpy=1.18.1` then select y if prompted. To determine if numpy 1.18.1 has installed successfully you should run the line: **np.__version__** in the Python console.

The Spyder IDE

Included within the Anaconda distribution is a the 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 most likely need, or want to make use of a GIS software program such as ArcGIS or the open-source QGIS.

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 (usually only for appropriative 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

Separate instructions are provided for Geoprocessing and GIS steps. Spatial data is not necessarily needed if the basin IDs are provided.

See Appendix I following this document below.

DWRAT IMPLEMENTATION PROCEDURES

The State Water Board 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 here. Likewise, modeled flows were developed for both the Lower and Upper Russian Watersheds. The documentation and procedures for this process are available here.

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
- 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)

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	A	D	364.9468	136.4509	74.09335	43.78711	27.11484
3	B	C	254.7392	102.0952	58.18691	35.35517	22.27914
4	C	D	35.91392	13.72602	7.572932	4.464099	2.725507
5	D	E	239.9048	88.14942	48.93739	29.46982	18.48893
6	E	F	265.9021	106.8238	58.22519	34.44099	21.44053
7	F	I	533.9385	205.2595	110.3702	65.11993	40.50863
8	G	H	56.40893	19.90017	10.10202	5.623973	3.294625
9	H	I	151.8228	48.19914	23.82803	13.12939	7.718479
10	I	J	112.5493	38.06737	21.15134	12.8162	8.129454
11	J	L	41.90479	11.48648	6.619378	4.26117	2.856694
12	K	L	159.721	56.53388	25.50764	13.11015	7.473957
13	L	M	67.8	24.03233	12.77197	7.466658	4.668178
14	M	M	3.554221	0.865091	0.49874	0.328826	0.227332

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	S002059	F	0	0	0	0	0
3	S002341	E	0	0	0	0	0
4	S002379	L	0	0	0	0	0
5	S004241	D	0	0	0	0	0
6	S004997	L	0	0	0	0	0
7	S006298	I	0	0	0	0	0
8	S006491	I	0.33	0.33	0.33	0.33	0.33
9	S008071	B	0.63	18.36	29.06	28.53	13.46
10	S008074	F	0	0	0	0	0
11	S008080	A	0	0	0	0	0
12	S008081	E	0	0	0	0	0
13	S008082	E	0	0	0	0	0
14	S008121	I	0	0	0	0	0
15	S008122	I	0	0	0	0	0
16	S008128	L	3.89	2.63	13.81	13.34	7.01
17	S008184	B	0	0.76	0.56	0	0
18	S008348	F	0	0	0	0	0
19	S008384	E	0	0	0	0	0
20	S008392	D	0	0	0	0	0
21	S008451	B	0	0	0	0	0
22	S008453	B	0	0	0	0	0
23	S008596	D	0	0.84	3.22	0	0
24	S008659	A	0	0	0	0	0
25	S008703	L	0	0	0	0	0
26	S008765	E	1.55	1.98	2.33	7.99	1.79

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	I
1	USER	BASIN	PRIORITY_DATE	PRIORITY	2021-05	2021-06	2021-07	2021-08	2021-09
2	A004413	5	1/8/1925	1	4.643364	6.135694	10.46143	9.216146	7.612899
3	A005317	8	12/28/1926	2	0.338845	0.280049	0.303256	0.227444	0.317957
4	A005356	16	2/14/1927	3	1.237782	2.475563	1.547227	1.547227	2.475563
5	A006426	14	9/3/1929	4	0.475772	0.475772	0.475772	0.475772	0.475772
6	A007409	17	10/19/1932	5	0	0	0.413416	0.413416	0.398291
7	A008060	14	8/9/1934	6	0.02785	0.02785	0.02785	0.02785	0.02785
8	A009518	14	3/2/1939	7	0.368762	0.409735	0.409735	0.368762	0.286815
9	A009686	5	7/31/1939	8	4.804654	4.649665	4.804654	4.804654	4.649665
10	A009788	1	12/16/1939	9	0	0	0	0	0
11	A010198	6	5/3/1941	10	0	0	0	0	0
12	A011876	5	5/12/1947	11	21.2587	30.48176	41.48395	40.25271	33.34174
13	A013240	16	7/20/1949	12	0.019804	0.023208	0.026303	0.025375	0.025684
14	A013912	20	8/23/1950	13	0	0	1.512497	2.016662	0
15	A014029	3	11/2/1950	14	0.575568	0.575568	0.575568	0.575568	0.575568
16	A014076	3	11/28/1950	15	0.095928	0.095928	0.095928	0.095928	0.095928
17	A014080	1	11/30/1950	16	0	0	0	0	0
18	A014652	8	1/22/1952	17	0	0	0	0	0
19	A014691	9	2/28/1952	18	0.003094	0.003094	0.003094	0.003094	0.003094
20	A016088	18	10/13/1954	19	3.024993	4.033324	4.033324	4.033324	0
21	A016449	20	7/6/1955	20	0.002321	0.004178	0.005415	0.00557	0.005338
22	A016788	10	12/12/1955	21	0.046417	0.092834	0.247556	0.309445	0.309445
23	A017465	1	2/13/1957	22	0.009593	0.011604	0.011991	0.011991	0.009283
24	A017809	20	8/30/1957	23	0	0	0	0	0

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. See Suggest Software below. 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. The suggested

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 powershel 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 not required. It is not necessary to delete anything either. The files will be overwritted 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

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 input 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 two separate files in the folder named output.

APPENDIX 1

Draft DWRAT Geoprocessing Steps

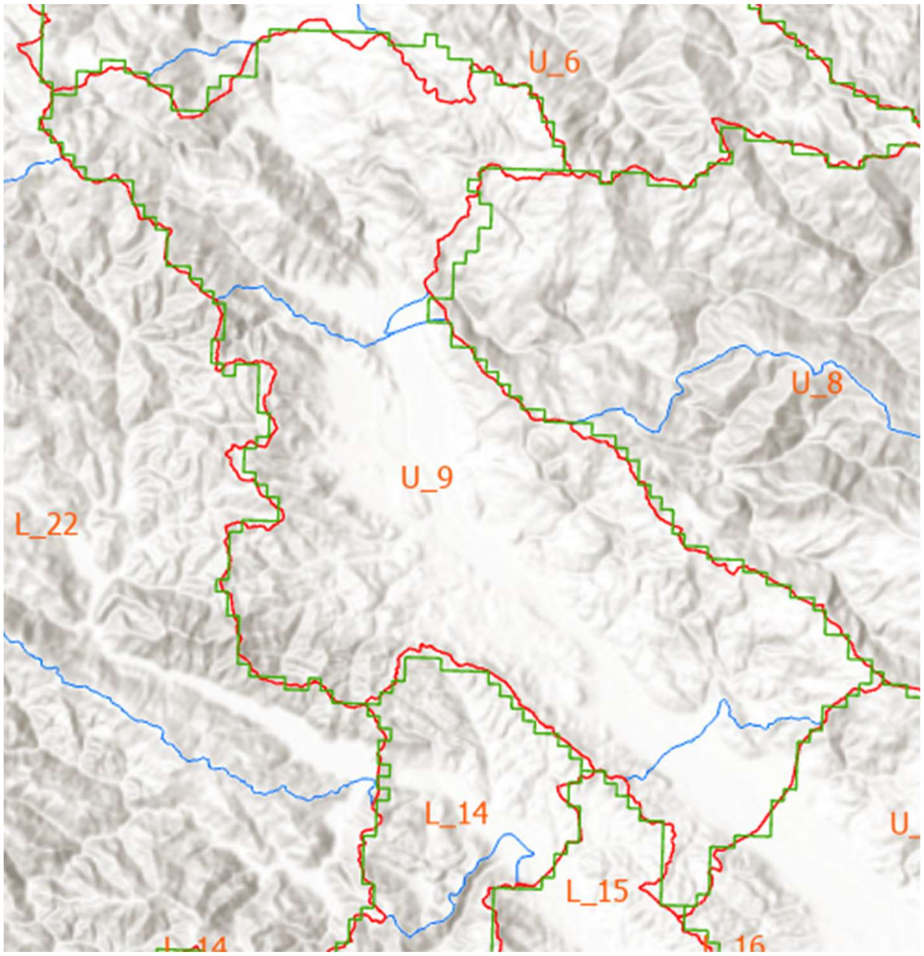
This document describes the recommended steps to:

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

Geoprocessing Steps:

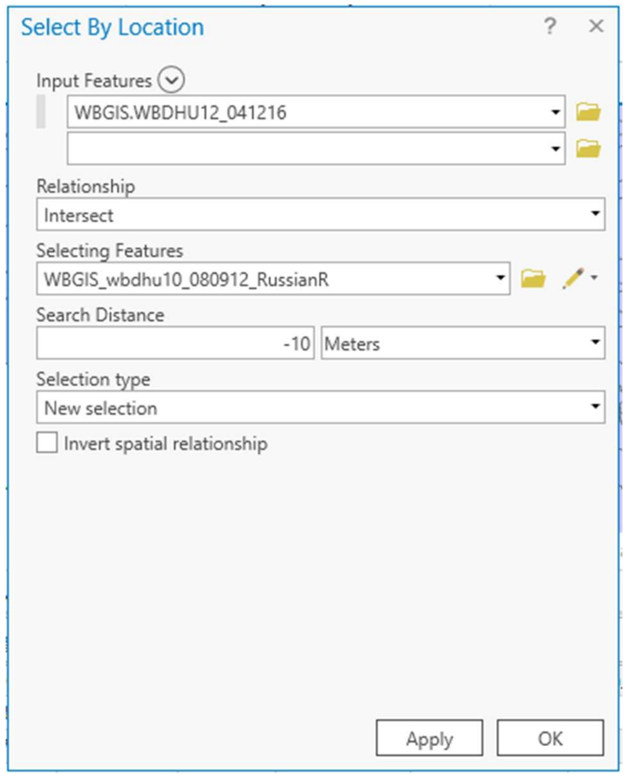
1. Add the necessary data layers to the map
    a. Add the medium resolution NHD flowline layer to the map, if not already added, and turn on the labels to display stream names (GNIS Name)
        i. The layer can be downloaded from USGS or it can be found on the Water Board's "water49" database
            1. To connect to water49, follow the directions from the GIS wiki page: http://wiki/gis/doku.php (If trying to access the webpage remotely, you will need to do so via the virtual desktop or VPN)
        ii. The specific version of this layer is unimportant, as it is used simply for general reference
    b. Add the polygon output from the Hydro Model (PRMS/GSFLOW) that displays the rough (pixelated) basins
    c. Add the HUC12 boundaries from USGS's Watershed Boundary Dataset
        i. Can be downloaded from USGS or it can be found on the Water Board's "water49" database
            1. To connect to water49, follow the directions from the GIS wiki page: http://wiki/gis/doku.php (If accessing the webpage remotely, you will need to do so via the virtual desktop or VPN)
            2. In water49, you will find the HUC12 layer under WBGIS.WatershedBoundaryDatasetApr2016 > WBGIS.WBDHU12_041216
    d. Add the "Topographic" ESRI basemap
2. Create the DWRAT basins polygon layer
    a. You will need to reconcile the Hydro Model basins with HUC12 boundaries to create the DWRAT basins
        i. Many portions of each Hydro Model basin boundary will generally coincide with the HUC12 boundary lines. For these portions, go with the HUC12's smooth line. However, most basins have portions of their boundaries that do not coincide with the HUC12s (the Hydro Model basins may either be larger or smaller than the HUC12s). For these portions of a basin's boundary, where the boundaries diverge completely, you will have to digitize your own boundary line freehand (described below). It is best to start with the HUC12 polygons and then edit them where necessary to approximate the Hydro Model basin polys. See the screen shot below for an example (Blue = HUC12

boundaries, Green = Hydro Model boundaries, Red = final reconciled DWRAT basin boundaries)

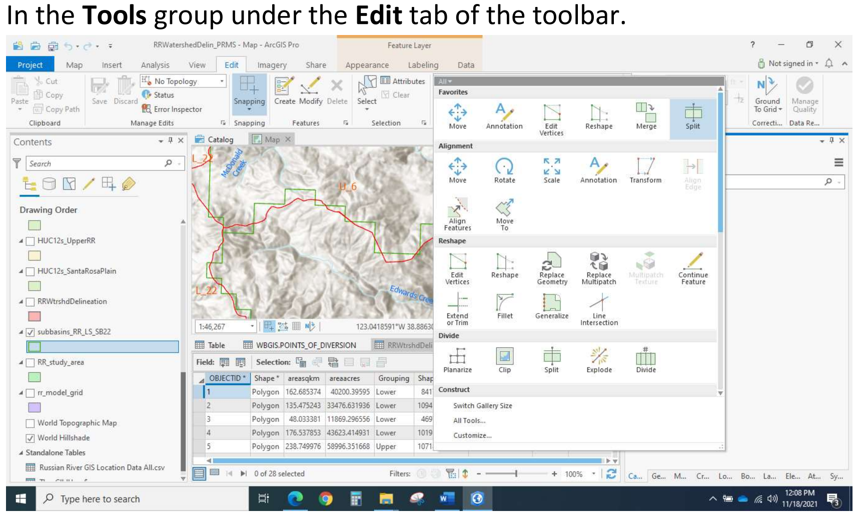


1. First, you need to create a layer containing only the HUC12s that are in the area of interest (AOI).
    a. **Select By Location** to select all HUC12 polys that intersect with your AOI
        i. Hint: If your AOI is composed of one or more larger HUC polys (HUC10, HUC8, HUC6, etc.), then use the **Select By Location** tool's "**Intersect**" option, but set the **Search Distance** to **-10 meters** so that you do not accidentally grab surrounding

HUC12s (see screen shot).



   b. Confirm that only the intended HUC12s in the AOI were selected
   c. Export the selection to a new layer
      i. **Data** > **Export Features**
      ii. Add the new layer to the map
2. Reshape the HUC12 polygons to reconcile them to the Hydro Model polygons
   a. In cases where the HUC12 poly needs to be split up, use the **Split** tool
      i. In the **Tools** group under the **Edit** tab of the toolbar.



      ii. When drawing the boundary portion, use the Hydro Model line only to approximate where the basin boundary should be drawn. The Hydro Model's resolution is low so it often does not accurately follow the ridgelines that make up the intended basin boundaries. Add the “Topographic” basemap to the map and reference the hillshading of that layer to accurately draw that portion of the basin boundary.

1. Hint: The Topographic basemap consists of two components in the Table of Contents: The World Topographic Map and the World Hillshade. You will want to turn off the the World Topographic Map component to better see the World Hillshade.

iii. Be sure to occasionally save your edits to the layer (**Edit** > **Save**)

b. In cases where two or more HUC12 polys must be combined to match the Hydro Model poly, use the **Merge** tool

i. In the **Tools** group under the **Edit** tab of the toolbar (right next to the Split tool)

ii. Be sure to occasionally save your edits to the layer (**Edit** > **Save**)

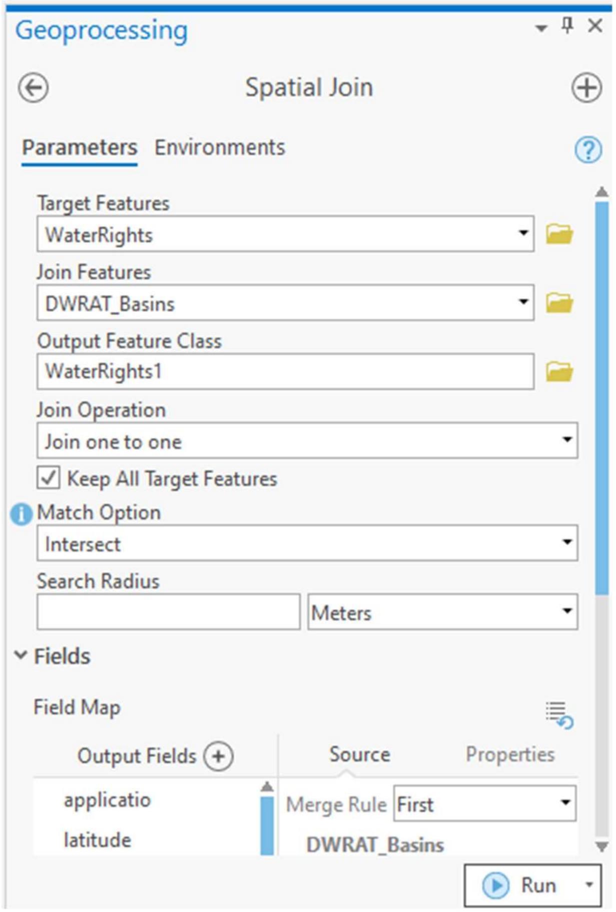
b. With the DWRAT basin boundaries established, add a text field to that layer and populate it with unique identifiers for each basin (identifiers could be as simple as 01, 02, etc.). These will be the basin IDs.

3. Assign the corresponding basin IDs to the water rights point features contained within

a. Add to the map the water rights points layer (which contains only a single point for each water right) created previously

b. Perform a **Spatial Join** with the basins layer to join/append the basins layer attributes to the intersecting water rights points

i. Using the Field Map option of the tool, set the tool to only join/append the basin ID field to the water rights layer (there is no need to bring any other fields along)

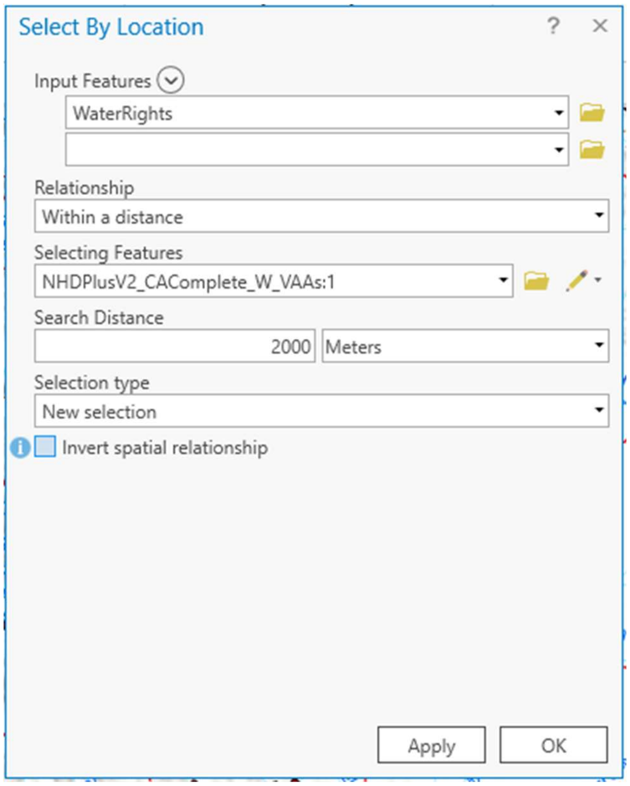


4. Designate mainstem water rights (this step may not be necessary, read on for guidance)
    a. For each DWRAT sub-basin, you need to determine whether a simple basin approach is adequate or whether the basins need to be further divided into mainstem vs headwater areas. Those basins with inflow from outside the basin (whose flow is available for mainstem diverters to divert) may warrant using the mainstem vs headwaters method, whereas headwaters basins that have no inflow from other basins may not require a headwaters vs mainstem distinction (the entire basin is considered headwaters). Proceed with the following steps for those basins that need a mainstem/headwaters distinction.
    b. Add the layer created previously that contains all the correct POD points (that have been corroborated by the PLSS Sections)
    c. Select all POD points that are within a certain distance of the "mainstem" stream line. The distance will depend upon geology and/or topography of the area. You want the distance from the stream to be big enough to select POD's that may be slightly misplotted, but not SO big that the list is onerous to review. For mainstems In the Russian River Watershed, 2 km seemed to strike a nice balance.
        i. Use the **Select By Location** tool (see screen shot for example):
            1. Set the relationship to "**Within a distance**"
            2. Your **Selecting Features** will be the **NHD flowline** layer. But before the tool is run, you must limit the tool to search only near the stream of interest. This can

be done by performing one of the following to the NHD flowline layer prior to running the Select By Location tool:

   a. select the mainstem flowlines of interest (**Select By Attributes** on **GNIS Name**) or
   b. apply a Definition Query (**Layer Properties > Definition Query**) to the flowline layer so that only the flowlines of interest are displayed

3. Define your search distance and run the tool



d. Remove from the selection all PODs whose water right is Cancelled or Revoked (but keep those that are Inactive)
e. Sort PODs alphabetically by the Source Name field and remove from the selection any PODs that list a source other than the mainstem of interest or its underflow. Unspecified or uncertain sources should also remain selected.
f. Now visually inspect the selected POD points on the map and remove from the selection any that appear to have other sources or are clearly outside the alluvial plain
   i. The alluvial plain can be distinguished by displaying the Topographic basemap in the map. The shaded hillslopes can help you determine the toe of the hillslope where the topography transitions out of the alluvial plain.
g. From the attribute table, copy the selected records to the clipboard and paste them into a blank Excel spreadsheet
h. In Excel, remove all duplicate App IDs from the list based on the application ID (Data > Remove Duplicates)
i. Save the spreadsheet as a .csv (ESRI products do not seem to work well with .xlsx)
j. Add the .csv to the map

k. Also add the water rights points layer (*__the one with only one point per water right__*), if not already in the map
l. Add a text field to the water rights points layer and name it “Mnstm”
    i. Note: You may have several mainstem rivers in your AOI. For instance, in the Russian River watershed, we identified three mainstem rivers: The Russian River, Dry Creek, and Mark West Creek. You may want to create a separate field for each mainstem, as this will afford you greater filtering specificity later in the process. In the Russian River example, we had four mainstem fields: “Mnstm_RR”, “Mnstm_DryC”, “Mnstm_MWCk”, and then one roll-up field that indicates general mainstem (regardless of the specific stream) named “Mnstm”.
m. Join the .csv table to the water rights points layer (the one with only one representative point per water right)
n. Any water right with a corresponding joined record from the .csv table will be considered “mainstem” so **Select By Attributes** to select all water rights points that have a corresponding record from the joined .csv
    i. Hint: You can apply the IS NOT NULL statement on the App ID field of the joined .csv table to find all records with a corresponding join
    ii. Note: Any water right will be considered mainstem if it has one or more mainstem POD points
o. Now **right click** on the heading of the Mnstm field in the attribute table and **Calculate Field** to set all the selected records to “Y” for that field, indicating that they are mainstem water rights