# SWAT+ Editor 1.2.0 Documentation

Prepared July 5, 2019 by Jaclyn Tech
Texas A&M AgriLife Research

plus.swat.tamu.edu

# About SWAT+

Over the past 20 years, the Soil and Water Assessment Tool (SWAT) has become widely used across the globe. The large numbers of applications across the globe have also revealed limitations and identified model development needs. Numerous additions and modifications of the model and its individual components have made the code increasingly difficult to manage and maintain. In order to face present and future challenges in water resources modeling SWAT code has undergone major modifications over the past few years, resulting in SWAT+, a completely revised version of the model.

Even though the basic algorithms used to calculate the processes in the model have not changed, the structure and organization of both the code (object based) and the input files (relational based) have undergone considerable modification. This is expected to facilitate model maintenance, future code modifications, and foster collaboration with other researchers to integrate new science into SWAT modules. SWAT+ provides a more flexible spatial representation of interactions and processes within a watershed.

The Soil and Water Assessment Tool Plus (SWAT+) is a public domain model jointly developed by the USDA Agricultural Research Service (USDA-ARS) and Texas A&M AgriLife Research, part of The Texas A&M University System. SWAT+ is a small watershed to river basin-scale model to simulate the quality and quantity of surface and ground water and predict the environmental impact of land use, land management practices, and climate change. SWAT is widely used in assessing soil erosion prevention and control, non-point source pollution control and regional management in watersheds.

# Installation

## 1. Install QGIS 3

If you plan to use QSWAT+ to set up your watershed, please install QGIS 3 before installing SWAT+. It can be downloaded from the QGIS download page, where you should select the **Long term release repository**, and the **64 bit standalone installer**. Use the default folder C:\Program Files\QGIS 3.4 as the installation folder. See the QSWAT+ manual for further instructions.

**Please note you must use QGIS version 3, not version 2.**

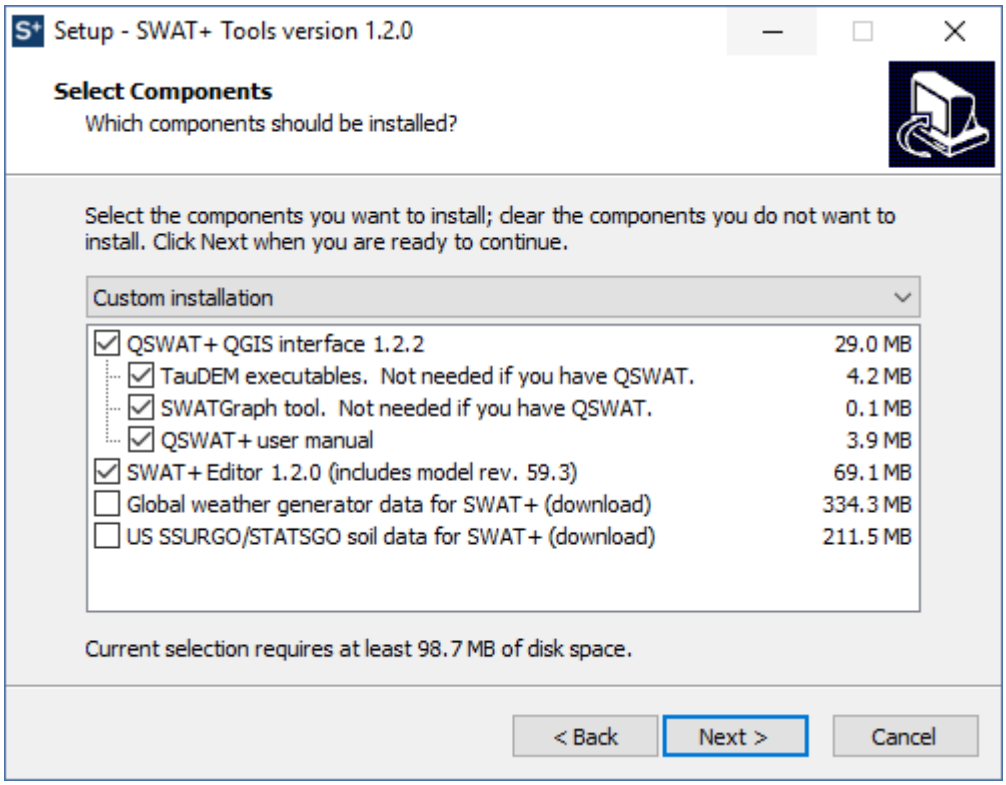
## 2. SWAT+ Tools Installer

To install the SWAT+ model and interface components, please use the installer linked below. The installer is for 64-bit Windows machines. Administrator privileges are not required, however you must have access to your C drive.

### Download SWAT+ Installer 1.2.0

Windows 64 bit
Includes SWAT+ rev. 59.3, QSWAT+ 1.2.2, SWAT+ Editor 1.2.0, and SWAT+ databases
Release notes

Upon opening the installer, you will be presented with a screen asking which components you would like to install. The soils and weather generator databases will be downloaded in the next screen if checked.




If QSWAT+ and SWAT+ Editor are selected, their individual installers will open next.

# SWAT+ Model (Command Line Executable)

The model itself is packaged with SWAT+ Editor and we recommend using the editor to modify your inputs and run the model. However, if you would like to download just the command line executable file, it is available below.

## Windows 64 bit

Windows SWAT+ rev. 59.3 executable — swatplus_exe_rev59_3.zip - 7MB

## Linux 64 bit

Linux SWAT+ rev. 59.3 (static) — swatplusrev59-static.zip - 1MB

Linux SWAT+ rev. 59.3 (dynamic) — swatplusrev59.zip - 972KB

Note: the QSWAT+ and SWAT+ Editor available above require a Windows machine. However, we plan to compile for Linux at a later time. Please contact us if you need this.

## SWAT+ SQLite Datasets

We recommend using the SWAT+ Tools installer above for installing the SWAT+ datasets databases to their proper locations. However, if you need to access these components individually, they are linked below:

- SWAT+ datasets
- SWAT+ global weather generator data
- SWAT+ US SSURGO/STATSGO soil data

These files should be placed in `C:\SWAT\SWATPlus\Databases`

# Getting Started

Download a sample project and watch the guide video to get started with QSWAT+ and SWAT+ Editor

---

## What you'll need

> Make sure you have downloaded and installed QSWAT+, SWAT+ wgn and soils databases, and SWAT+ Editor as described in the installation section linked below.

→ Installation /installation

The following guide will show you how to get started with SWAT+ Editor. This guide does not cover how to set up your watershed in QSWAT+. Please refer to the QSWAT+ manual for this step.

# Demo project files

Please use the following Robit demo project, which has already been set up in QSWAT+. Alternatively, you may use the Robit project you set up on your own after following the steps in the QSWAT+ manual.

Robit Demo Project for SWAT+ Editor 1.2.0 robit_demo_1.2.0.zip - 4MB

## Watch the SWAT+ Editor guide video

Please watch the guide video before launching SWAT+ Editor. This video will quickly walk you through the steps needed to bring your QSWAT+ project into the editor, as well as show you how to catch and report errors. If you prefer a written guide instead of or in addition to the video, a walk through is shown below.

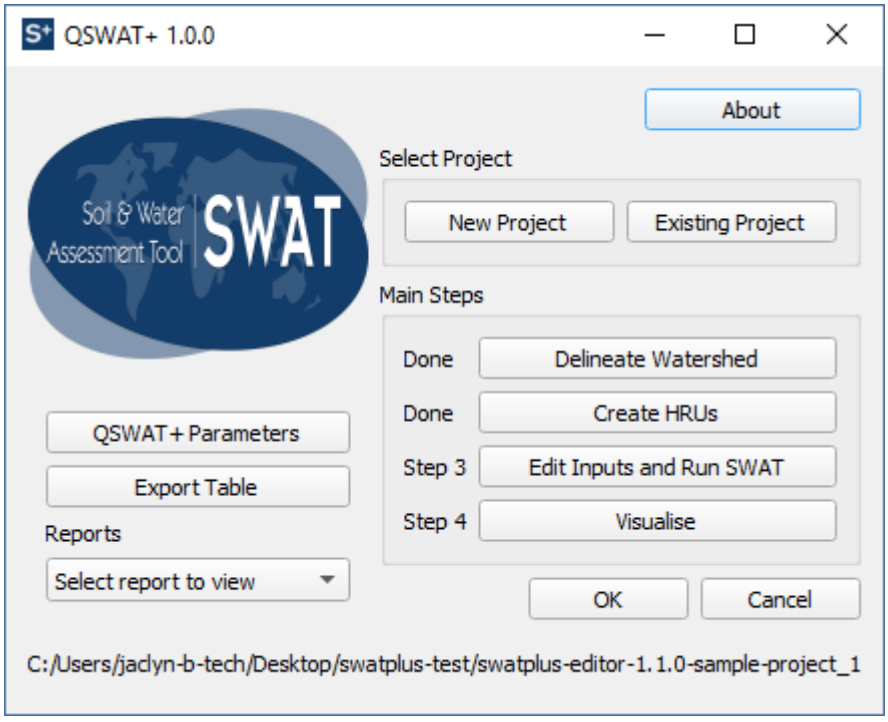
# Walk through guide for SWAT+ Editor

If you do not want to watch a video, follow the steps below. This guide shows the basic steps you need to take to set up your model, but does not cover all input parameters available through the editor. Please refer to the SWAT+ Editor documentation section for help editing specific input parameters.

→ SWAT+ Editor Documentation /user/editor

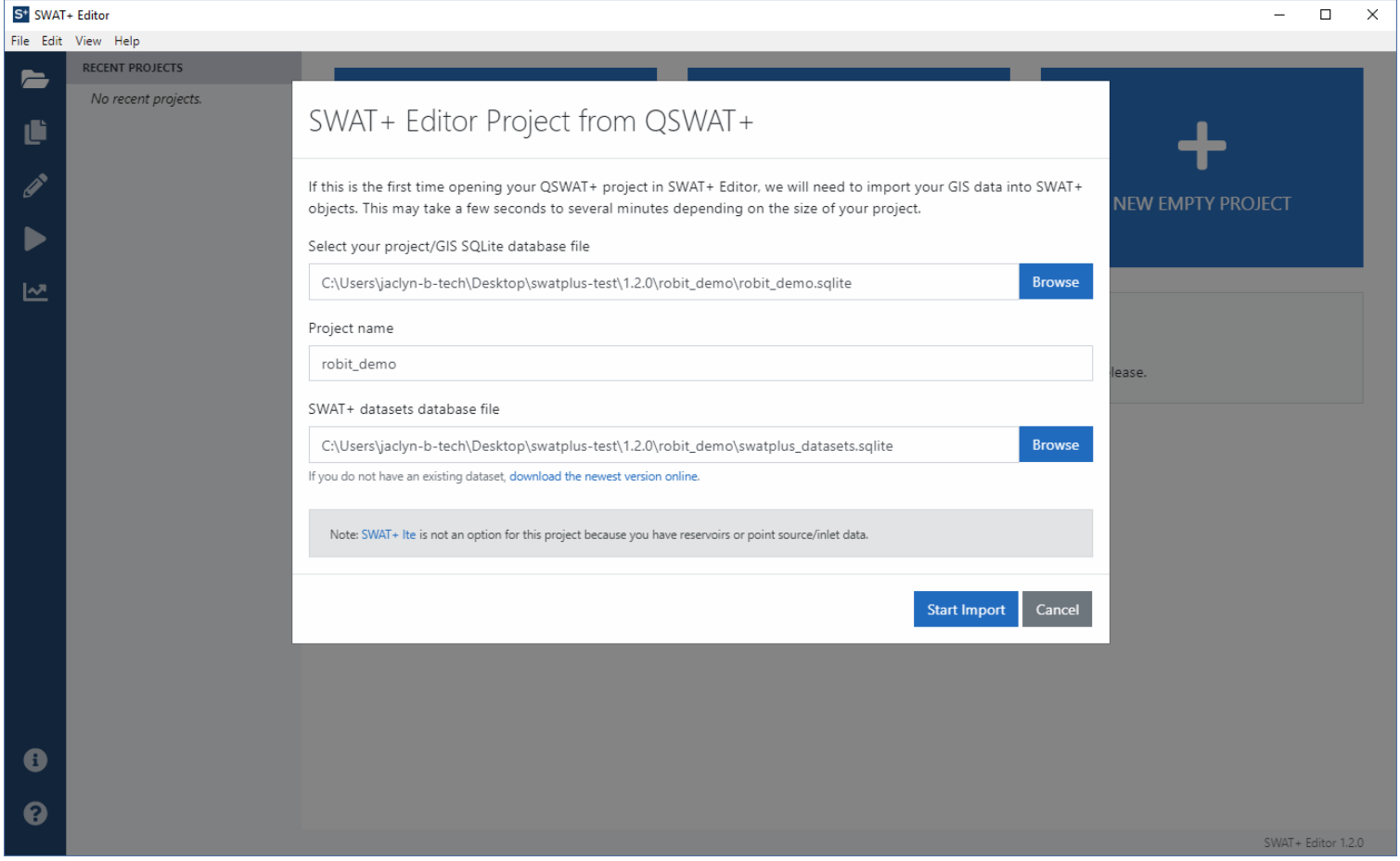
## Set up your project in QSWAT+

Follow the QSWAT+ manual to set up your watershed. To open SWAT+ Editor from within QSWAT+, click the button for Step 3: Edit Inputs and Run SWAT.

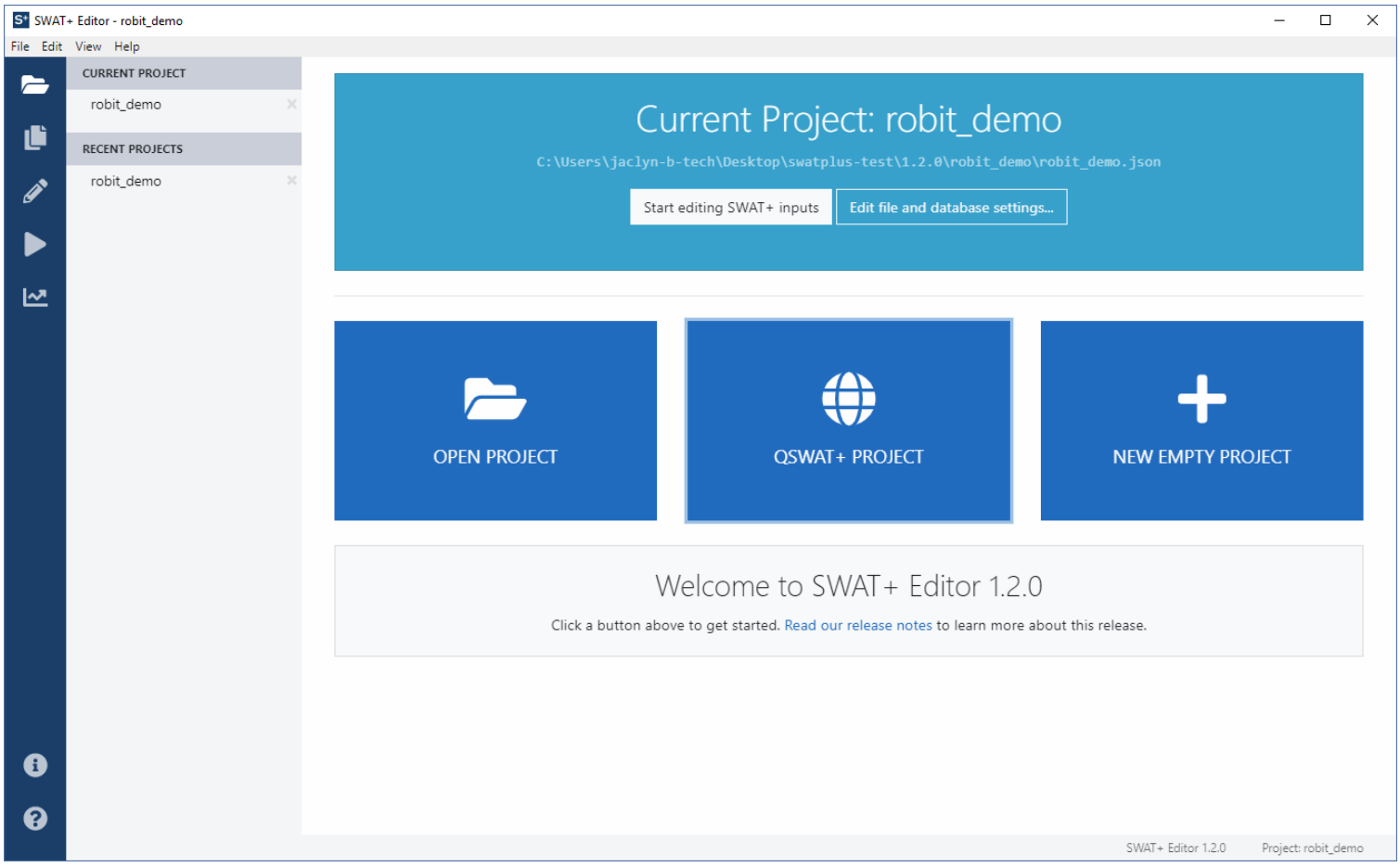



## Open your project in the editor

The first time your open your QSWAT+ project in SWAT+ Editor, your data must be imported from the GIS tables into SWAT+ database format. This may take a few seconds to several minutes depending on the size of your watershed.

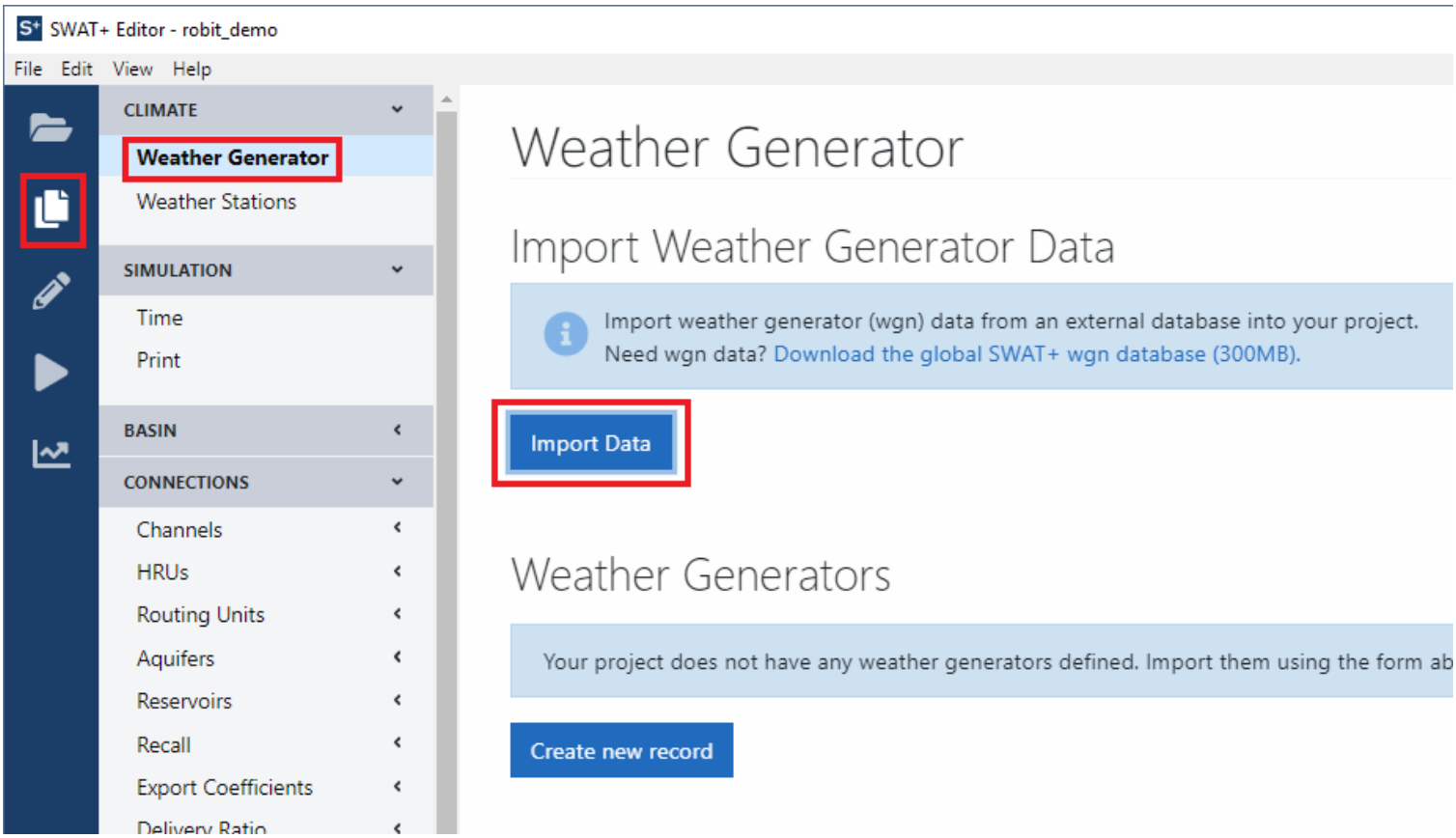



When your GIS data is done loading, your project will be displayed in the light blue box in the top center of the editor. Click the start editing SWAT+ inputs button to begin.

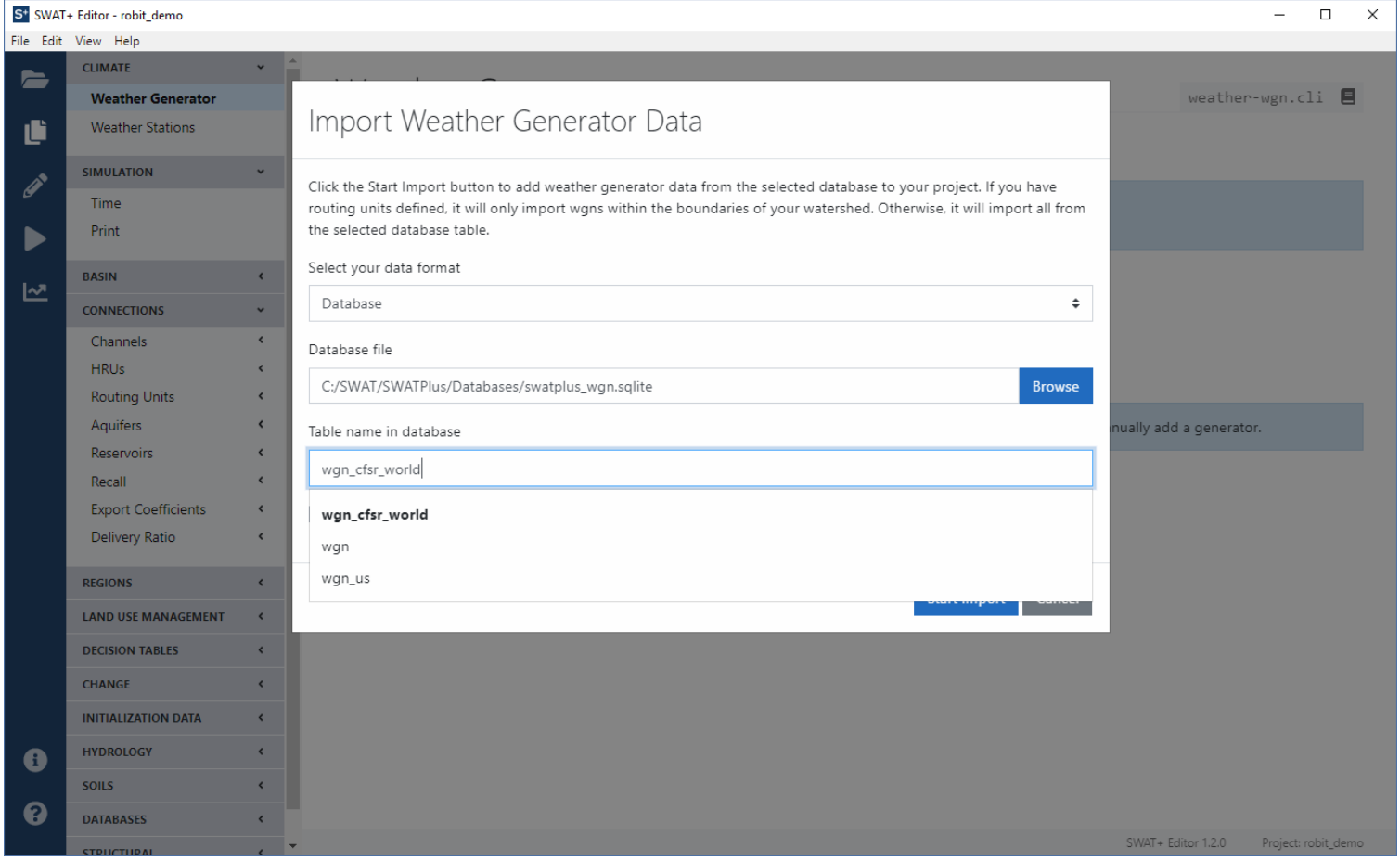



## Edit SWAT+ inputs

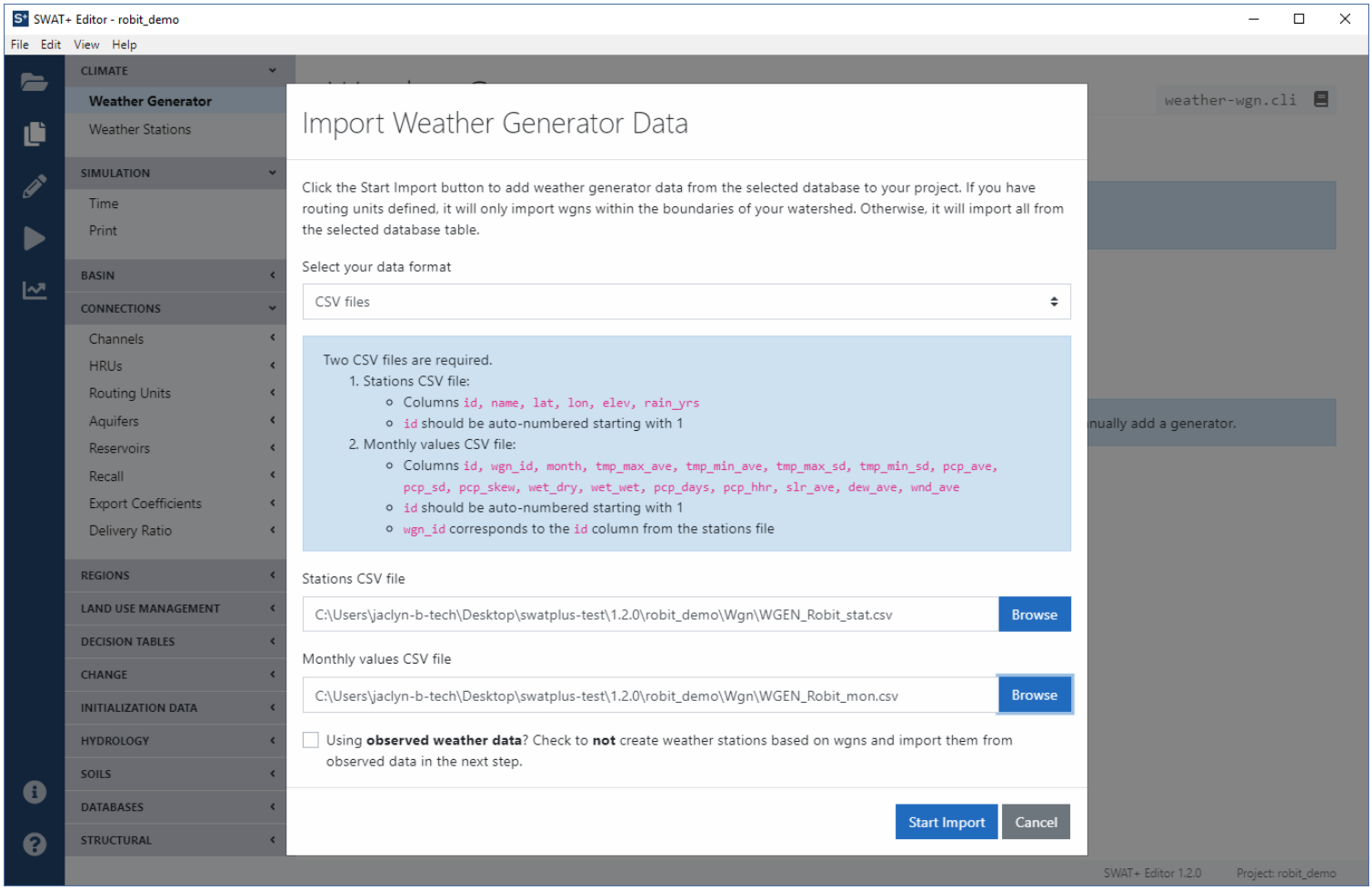
The first step you should take is to add weather generator (WGN) data. It may be imported from the distributed WGN database in the SWAT+ Tools installer, or from CSV files. Go to the weather generator section and click the import data button as shown in the screenshot below.



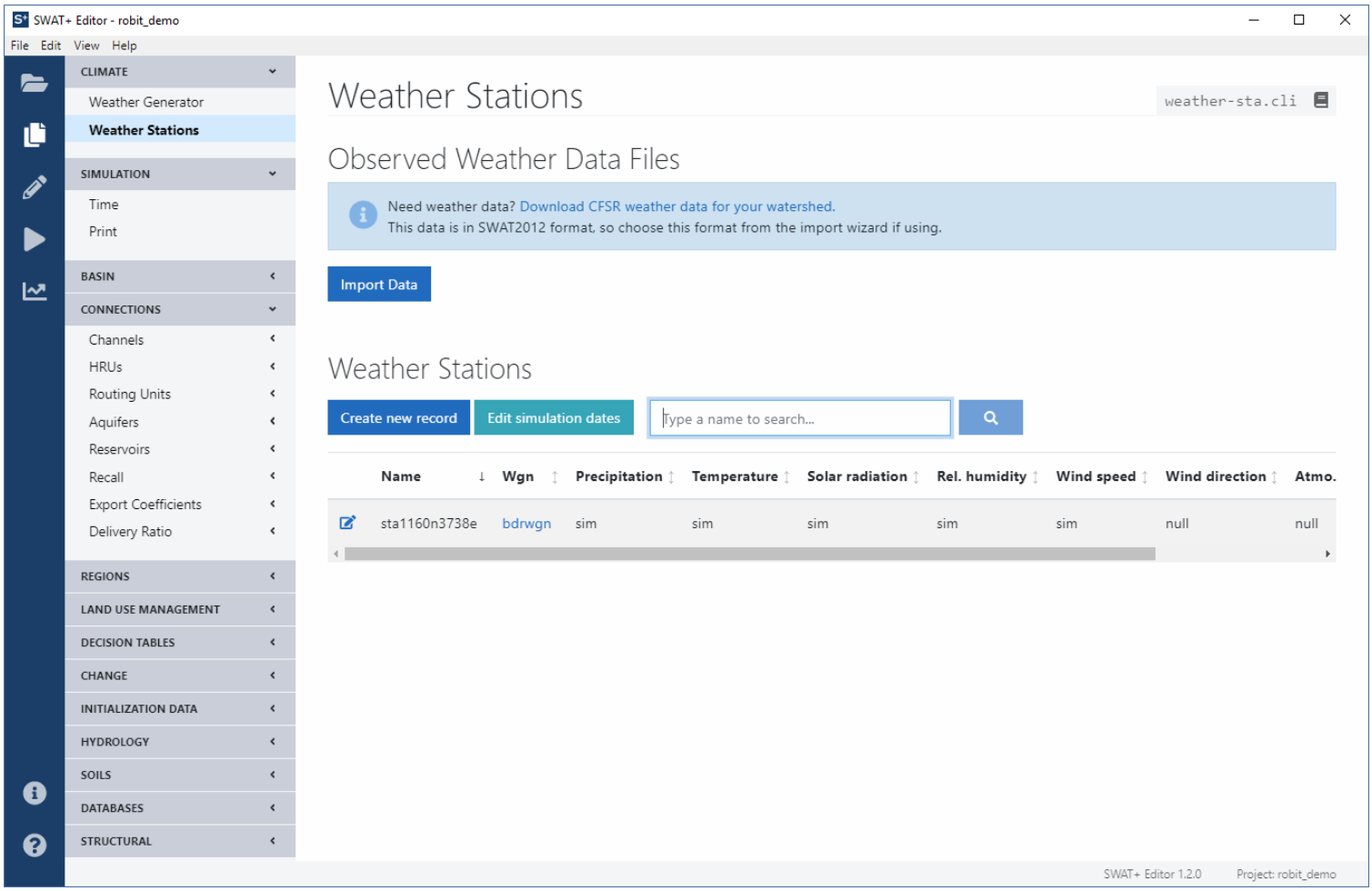

The database import option is selected by default, with the global CSFR weather generator table chosen. Click in the box to see other table options. `wgn_us` is weather generator data for the United States. `wgn` is an empty table you may populate with your own data if desired.



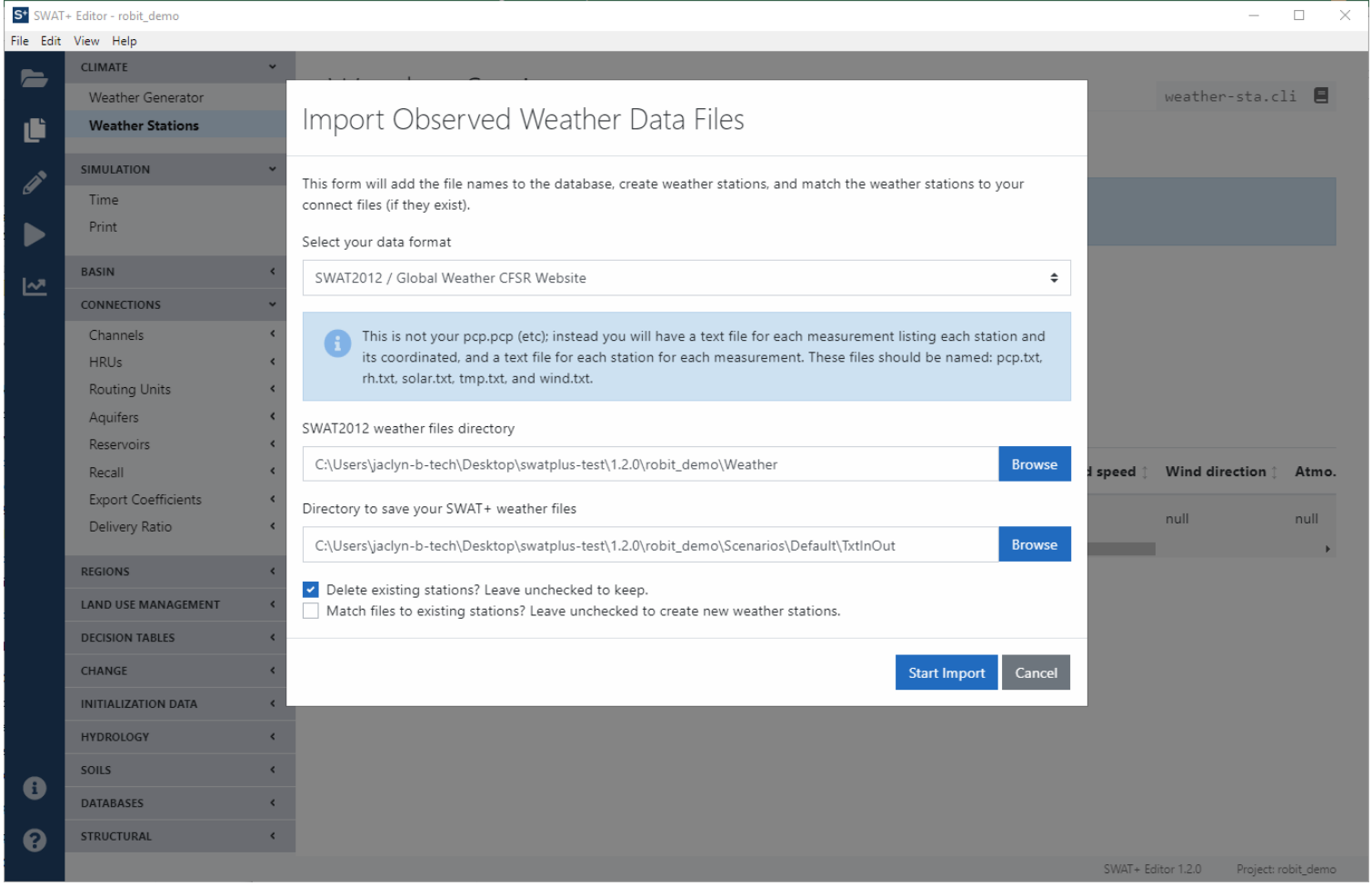

Alternatively, select CSV files from the top pull-down menu. In the Robit demo dataset, we have provided CSV files to use for weather generators, located in the Wgn folder of the sample project linked at the top of this page. Select these files in the editor as shown in the screenshot below, then click the start import button.



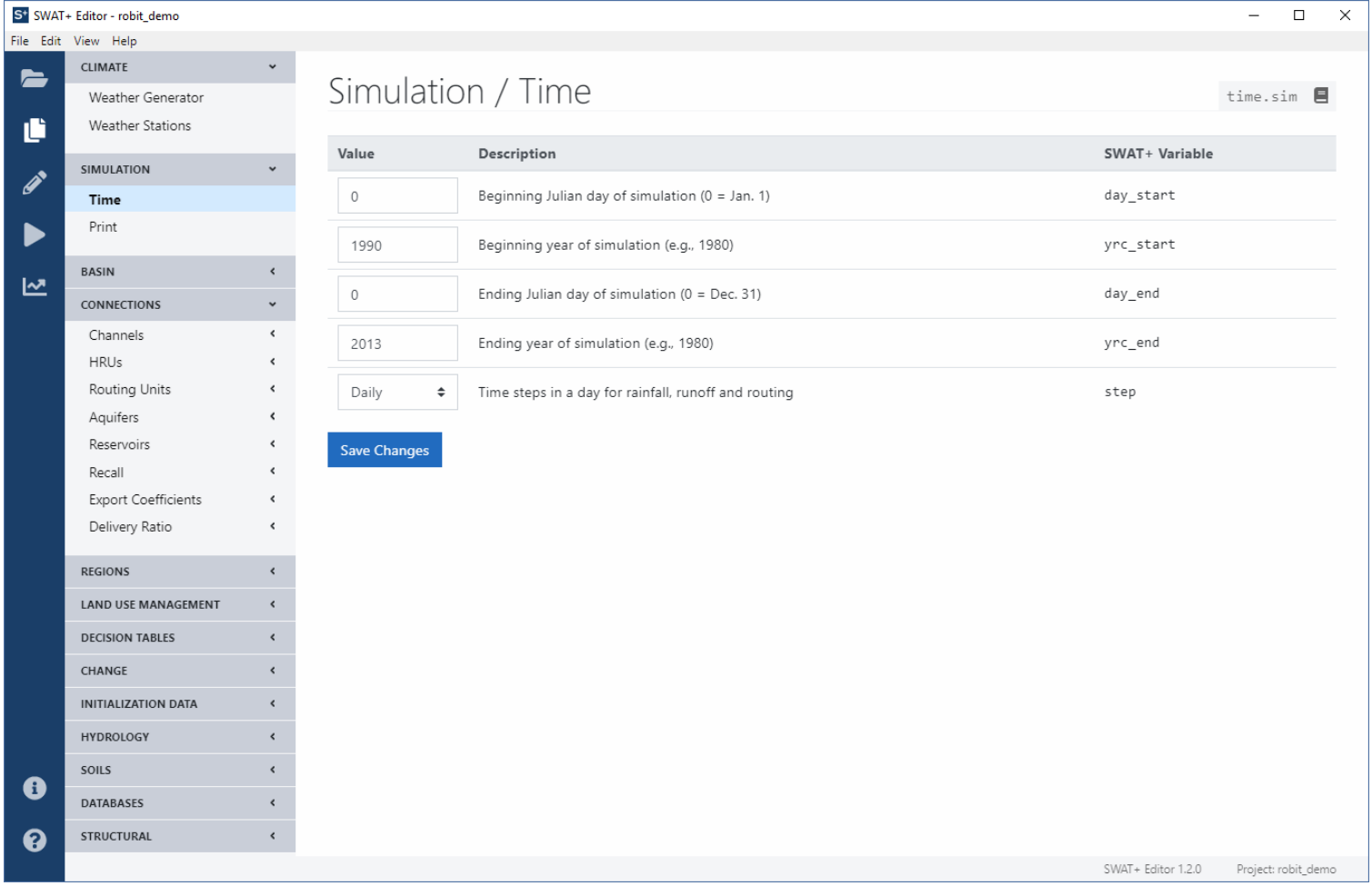
You should now see one weather generator station added. Next, we need to add observed weather data. From the left menu in the editor, click the Weather Stations link.



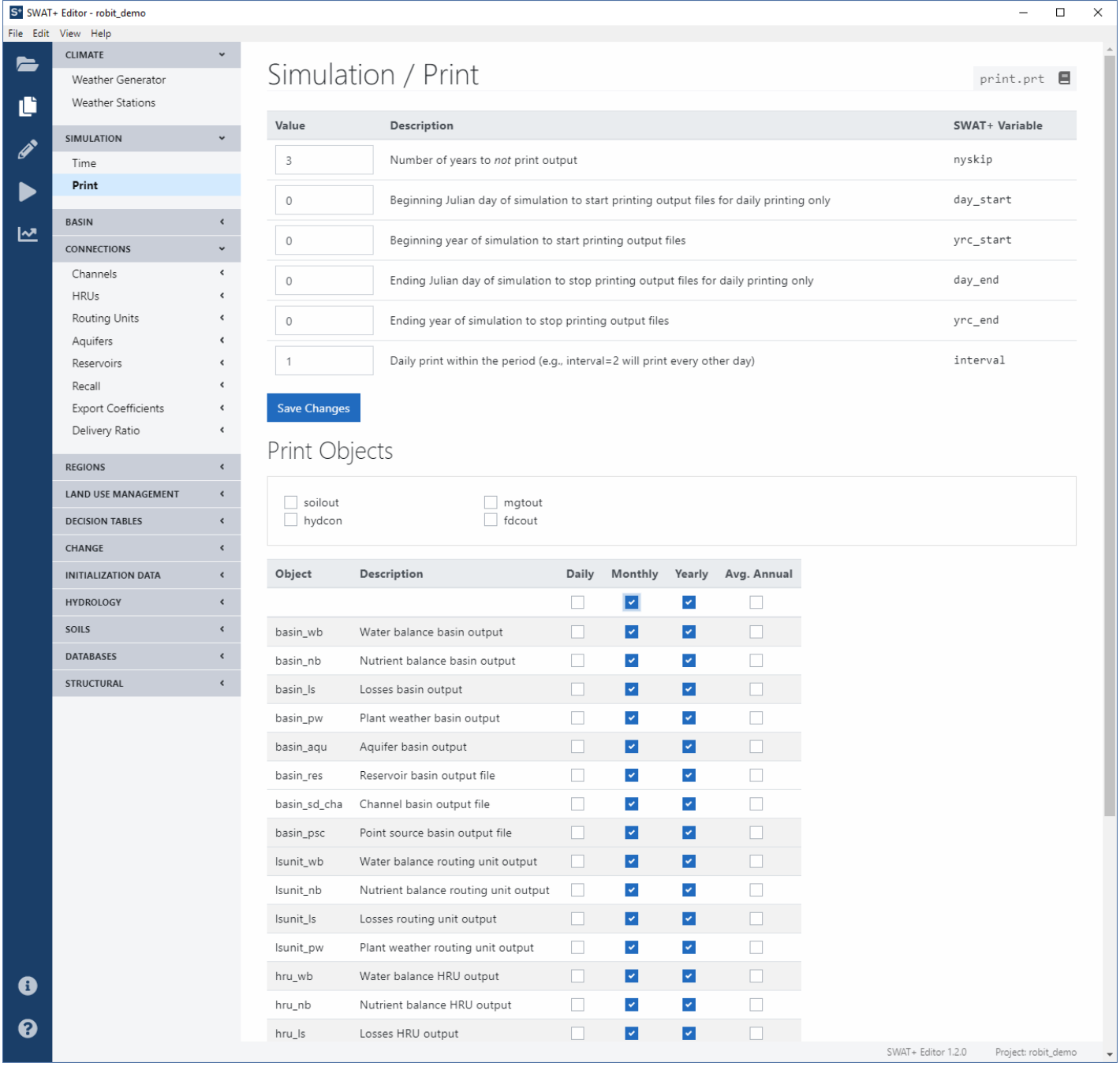

Click the import data button to import your files. In the Robit demo dataset, we have provided weather data in SWAT2012 format in the Weather directory. Select this folder as the SWAT2012 weather files directory in the editor's import form. You may also choose where to save the files when they are converted to SWAT+ format. By default, your TxtInOut folder is selected. Click the start import button to continue.




Once your data has been converted to SWAT+ format and imported to your project, we've also automatically adjusted your simulation dates. From the left menu in the editor, under the Simulation heading, click on Time. You may alter your simulation starting and ending times, however make sure they fall within the range of your observed weather data. If you have made any changes, click the save changes button.



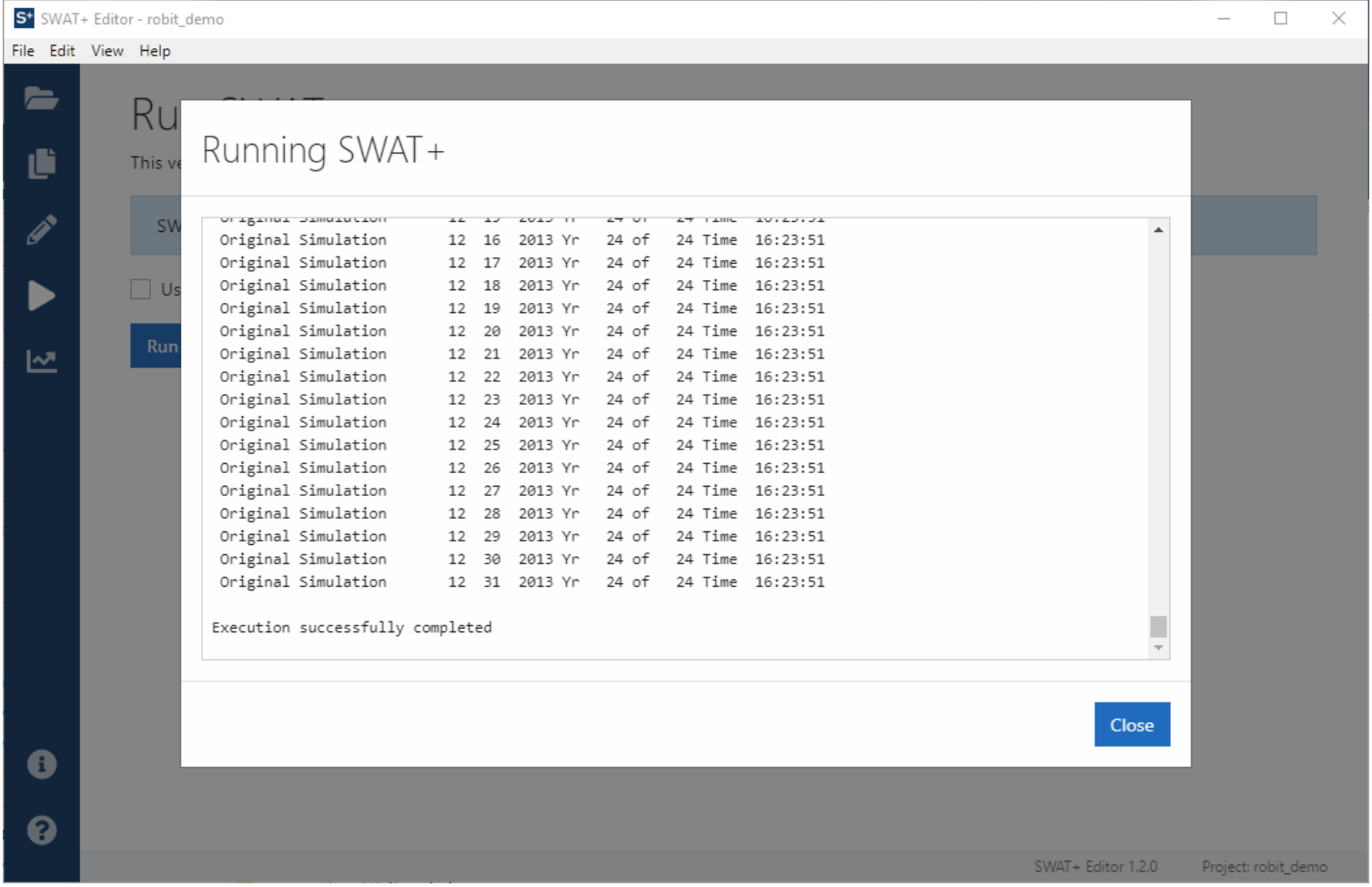

Next, select what data you want to print from your simulation. We advise not printing all daily output files as the file size can be very large and take a long time to import to the database for visualization. For the Robit demo, please change the number of skip years to 3, then check the boxes to print all monthly and annual output. Click the save changes button when done.



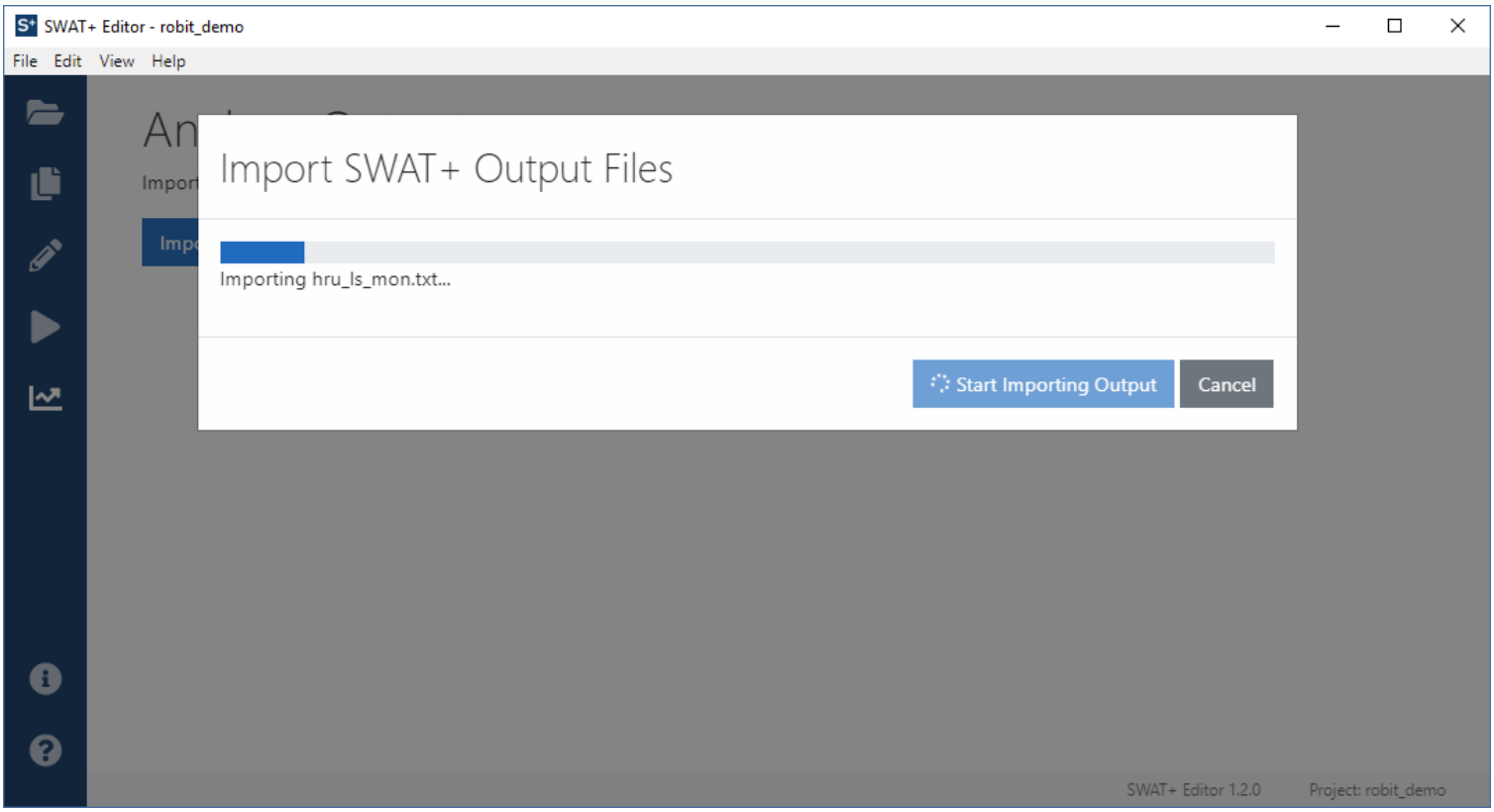

This concludes the last required section for editing SWAT+ inputs before writing files and running the model. The remaining sections are not covered in this walk-through, however you can look through the SWAT+ Editor Documentation section for more information.

In this demo, we'll go ahead and proceed to the next step: writing SWAT+ input files. From the navy ribbon on the left side of the editor, click on the pencil icon. Choose your location to write the files, then click the save and write files button.

When done, from the navy ribbon on the left side of the editor click the next step with the triangle/play icon to run the model. By default, the release version of the model is run. If you encounter an error, you can come back and check the box to run in debug mode to get a detailed error message from the model.




After the model run has finished successfully, from the navy ribbon on the left side of the editor, click the graph icon. From here you can import your model output text files into a SQLite database for use with the QSWAT+ visualization tool. Output may take a long time to import depending on the length of your simulation, the output files selected from the Simulation->Print section, and the size of your watershed.



When the editor finishes importing your output, you may close the editor by clicking the X in the top right of the editor window, or by going to File->Exit from the editor's menu.

You may now proceed to QSWAT+ step 4: visualization. Please refer to the QSWAT+ manual for more information about this process.

# Get Help

Documentation is available in the next section of this website, however please make use of our user groups if you get stuck or run into an issue.

---

## User Groups

Because there are 3 distinct pieces to SWAT+, we have created user groups for each. Please identify which area you are having difficulty with and choose the appropriate group:

**QSWAT+ user group**
For issues defining your watershed in QGIS and using the QSWAT+ plugin.

**SWAT+ Editor user group**
For issues related to the editor interface.

**SWAT+ model user group**
For questions and discussion related to the model itself, not the editor or GIS interfaces.

# Bug/Issue Reporting

For SWAT+ Editor bugs or issues, report to the user groups or the Bitbucket issue tracker.

# Troubleshooting

Before contacting support, please ensure you have the newest version of the tools installed. Check the common situations below, and if your error is not covered, consult the appropriate user group.

---

## I received an error setting up my watershed in QSWAT+

Make sure you have installed the most recent version of QSWAT+ and followed the manual's installation instructions closely.

If you received an error during steps 1, 2, or 4 of the QSWAT+ interface, please consult the QSWAT+ user group. Check existing questions to see if anyone else had the same problem. If not, please post your error and be as descriptive as possible about what you were doing when you received the error message.

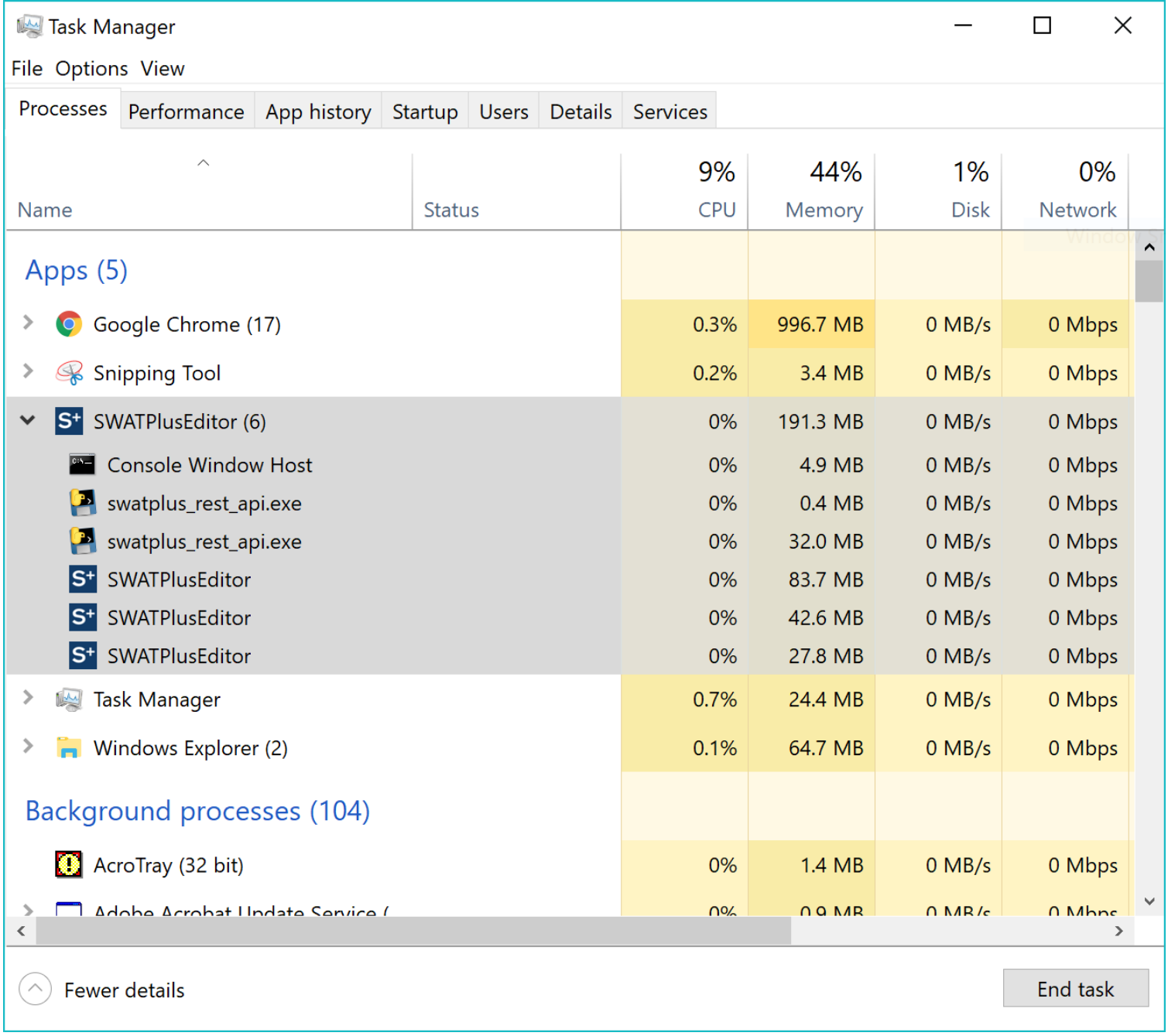
# I received an error in SWAT+ Editor

SWAT+ Editor is most often accessed during step 3 of the QSWAT+ plugin. It may also be launched on its own. Make sure you have installed the most recent version of SWAT+ Editor.

If your error is not covered by the solutions below, please consult the SWAT+ Editor user group.

## There was an error checking your project configuration

If you receive this message when you first launch your project in the editor, it is a sign the editor did not load its services correctly. While SWAT+ Editor is running, please right-click your Windows taskbar and select Task Manager. Expand the arrow next to SWATPlusEditor and ensure you have swatplus_rest_api.exe running as shown in the screenshot below.




If you do not see it running, please open a command prompt window. Make sure you are in the C drive, or the drive where you installed SWAT+. Browse to the location below, then run the swatplus_rest_api.exe file.

```
> cd C:\SWAT\SWATPlus\SWATPlusEditor\resources\app.asar.unpacked\api_dist
> swatplus_rest_api.exe
```

If it is working properly, you should receive something similar to the following:

```
* Serving Flask app "swatplus_rest_api" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

If this is what you received, please simply close SWAT+ Editor and try launching it again. If it still doesn't load, or you received an error message above, please contact Jaclyn directly at jaclynt@tamu.edu. You will receive a response as soon as possible, but please understand it may take several days. This error is possibly due to settings on your computer and you may not be able to get a solution from the Google group.

## I received an error importing GIS data or updating my project

Please post the error message to the SWAT+ Editor user group **AND include your project files**. This error is likely project specific and we cannot help you without seeing your project files.

## Other SWAT+ Editor errors

If you receive any other error messages from SWAT+ Editor, from its menu go to View -> Toggle Developer Tools. In the windows that pops open, toggle the Console tab and take a screenshot or copy an errors received and post to the SWAT+ Editor user group.

# I received an error running the SWAT+ model

If you received an error running the model, first make sure you have run it in debug mode and checked the detailed error received. If you are not able to make sense of the source of the model error, please post to the SWAT+ model user group **AND include your project files**. This error is likely project specific and we cannot help you without seeing your project files.

# Release Notes

SWAT+ Editor 1.2.0 with QSWAT+ 1.2.2 and SWAT+ revision 59.3

---

## SWAT+ revision 59.3

Download the revision notes for the model below.

SWAT+ Revision Notes 55.1-59.3 SWATPlus-Rev-Notes-55.1-59.3.pdf - 619KB

## QSWAT+ revision 1.2.2

- Environmental flows have been added.
- Revised treatment of ponds and reservoirs.

# SWAT+ Editor revision 1.2.0

- Compatible with SWAT+ rev. 59.3
- Fixes default routing in rout_unit_con for upland to floodplain surface runoff. Use fraction of area of upland routing unit surface runoff goes to channel/reservoir, the remaining goes to floodplain (see Bieger et al. JAWRA 2019). New projects only, existing projects should try re-import from GIS option.
- Change aquifer creation. Previously created one aquifer per channel. Changed to two per subbasin (upland/floodplain), and add a deep aquifer for each outlet.
- Fixes default principal/emergency area and volume of reservoirs. Note: new projects / re-import GIS data only. Existing projects should update values manually as needed. New defaults are described below:
  - Principal spillway area ( `area_ps` ) is set from GIS data
  - Emergency spillway area is set to `area_ps * 1.15`
  - Principal spillway volume is set to `area_ps * 10`
  - Emergency spillway volume is set to `area_es * 10`
- Un-managed ponds are now retained as HRUs in QSWAT+. Imported to the editor as HRUs with wetlands inputs (wetlands_wet and hydrology_wet).
- Update output database tables to include revisions from model rev. 59.3: channel and channel morph, reservoir, and wetlands columns.
- Project update function available for the following data changes related to model rev. 59.1-3:
  - Update cal_parm_cal abs_max=10 and units=m for flo_min and revap_min. Add dep_bot.
  - Update aquifer_aqu default values for gw_flo=0.05, dep_wt=10, flo_min=5, revap_min=3.
  - In plant_ini_item, yrs_init changed to fraction (change values to 1 where previously 15), and biomass increased for some plants. Lc_status changed to yes for past and barr plants.
  - Update codes_bsn default values for pet=1, rtu_wq=1, wq_cha=1
- User interface improvements:
  - Add csv import for weather generator data.
  - All related table search boxes return all possible results underneath matches to typed text.
  - Add automatic database rollback when user gets an error importing GIS or updating project.
- Bug fixes:
  - Fixed bug when updating project from 1.0.0, a variable was not declared.
  - Fixed bug where weather stations were created but not always assigned weather data file if one exists.
  - Fixed bug when trying to import weather data located on another hard drive.
  - Fixed bug where swatplus_rest_api.exe wasn't terminating correctly when exiting the editor.

# SWAT+ Editor revision 1.1.1

- Fixes bug when importing GIS data into plant communities not in the datasets database
- Print section usability update
- Update automatic project database backups so multiple failed import/upgrade attempts don't overwrite the original

# SWAT+ Editor revision 1.1.0 changes from 1.0.0

The remainder of this page outlines what changed from the initial release of the editor in fall 2018 to version 1.1.0 released in spring 2019. If you never used version 1.0.0, you can skip the remainder of this page.

- Upgrade function available for projects made with version 1.0.0.
- Compatible with SWAT+ rev. 59.
- Re-designed project setup page. If importing GIS, allow SWAT+ lte option for projects without point source or reservoir data.
- Channels now default to using the channel-lte structure (as per SWAT+ rev. 58). **Please note** that this means your channel input files are different (chandeg.con) and the channel output will be in channel_sd tables.
- New management schedule and decision table defaults determined by your HRU's plant type in plants.plt. It will use an automatic schedule based on corn (warm) or wheat (cold) plants. See the land use management documentation for more information.
- New editor sections for: basin parameters, connections--export coefficients, recall, delivery ratio, landscape unit regions, land use management, calibration, initialization data, soils, databases, and structural.
- Added export/import to and from CSV files for most sections.
- Other miscellaneous usability improvements.
- Automatic updating for more rapid bug fixes and releases.

## Notable limitations

- Constituents (pesticides, pathogens, heavy metals, salts) are not fully available through the editor yet.

## Project database changes

- d_table_dtl - add column file_name, repopulate table based on 4 new decision table files: lum.dtl, res_rel.dtl, scen_lu.dtl, flo_con.dtl
- d_table_dtl_act - add column const2
- d_table_dtl_act - rename columns application->fp and type->option
- recall_dat - drop columns sol_pest, srb_pest, p_bact, lp_bact, metl1, metl2, metl3
- exco_om_exc - drop columns sol_pest, srb_pest, p_bact, lp_bact, metl1, metl2, metl3
- exco_om_exc - rename columns ptl_n->orgn, ptl_p->sedp, no3_n->no3, sol_p->solp, nh3_n->nh3, no2_n->no2, bod->cbod, oxy->dox, sm_agg->sag, lg_agg->lg_agg
- aquifer_aqu - drop columns gw_dp, gw_ht, delay
- aquifer_aqu - add columns dep_bot (default value 10), dep_wt (default value 5), bf_max (default 1)
- aquifer_aqu - change spec_yld value from 0 to 0.05
- aquifer_aqu - add column init_id referencing initial_aqu
- fertilizer_frt - drop columns p_bact, lp_bact, sol_bact
- fertilizer_frt - add column pathogens
- hydrology_hyd - drop column dp_imp
- pesticide_cha - rename column sed_conc->pst_solub
- channel_lte_cha - rename table to hyd_sed_lte_cha
- hru_data_hru - drop column soil_nut_id
- hru_data_hru - add column soil_plant_init_id
- cal_parms_cal - change column type of units from number to text
- initial_cha - drop existing columns, add new columns: org_min_id, pest_id, path_id, hmet_id, salt_id (foreign keys to new tables in init)
- initial_res - drop existing columns, add new columns: org_min_id, pest_id, path_id, hmet_id, salt_id (foreign keys to new tables in init)
- reservoir_res - drop column pest_id
- wetland_wet - drop column pest_id
- sediment_res - add columns carbon and bd
- plants_plt - drop column plnt_hu and add column days_mat
- plant_ini - add column rot_yr_ini
- codes_bsn - change column type of atmo_dep from number to text
- rout_unit_ele - drop column hyd_typ, change foreign key of rtu_id from rout_unit_rtu to rout_unit_con
- constituents_cs - drop and re-create table
- dr_om_del, dr_pest_del, dr_path_del, dr_hmet_del, dr_salt_del, delratio_del - drop and re-create tables
- calibration_cal - drop and re-create table
- pesticide_pst - drop and re-create table
- codes_cal - rename table codes_sft, replace columns landscape and hyd with hyd_hru and hyd_hrulte
- ls_parms_cal - rename table wb_parms_sft
- ch_parms_cal - rename table ch_sed_parms_sft

- pl_parms_cal - rename table plant_parms_sft

## Drop tables

- pest_soil_ini
- pest_soil_ini_item
- path_soil_ini
- hmet_soil_ini
- salt_soil_ini

## Add new tables

- soil_plant_ini
- om_water_ini
- pest_hru_ini
- pest_hru_ini_item
- pest_water_ini
- path_hru_ini
- path_water_ini
- hmet_hru_ini
- hmet_water_ini
- salt_hru_ini
- salt_water_ini
- channel_lte_cha (new structure; not the same as old table renamed to hyd_sed_lte_cha)
- initial_aqu (same structure as initial_cha)
- calibration_cal_cond
- calibration_cal_elem
- water_balance_sft and water_balance_sft_item (replace ls_regions_cal)
- ch_sed_budget_sft and ch_sed_budget_sft_item (replace ch_orders_cal)
- plant_gro_sft and plant_gro_sft_item (replace pl_regions_cal)

# SWAT+ Datasets database changes

- d_table_dtl - add column file_name, repopulate table based on 4 new decision table files: lum.dtl, res_rel.dtl, scen_lu.dtl, flo_con.dtl
- d_table_dtl_act - add column const2
- d_table_dtl_act - rename columns application->fp and type->option
- plants_plt - drop column plnt_hu and add column days_mat
- Replace all decision table data
- Replace all plants_plt data
- Replace all fertilizer_frt data
- Replace all pesticide_pst data (also new table structure)
- Replace all var_range data
- Add new tables: version, tropical_bounds

## Output database changes

- New naming structure based on SWAT+ rev. 56

# Source Code

## Repository Links

- SWAT+ source code repository
- SWAT+ Editor source code repository
- QSWAT+ source code repository

# Run and Compile SWAT+ Editor

## Install Development Tools

- You may use any IDE of your choice, however Visual Studio Code is used by the developer.
- Install Python 3.7
- Install required Python packages. From command prompt, go to source code /api directory and run:

```
pip install -r requirements.txt
```

- Install Node.js
- Install required Node.js packages. From command prompt, go to the root directory of the source code and run:

```
npm install
```

## Run the Source Code

From a command prompt in the root directory (terminal inside Visual Studio Code)

```
npm run dev
```

Open a second command prompt and run

```
npm run electron
```

## Build the Source Code

First, use PyInstaller (included in the Python packages during install) to freeze the Python into executable files. This should be done from the /api directory in the source code. In Windows, run the supplied .bat file from a command prompt:

```
pyinstaller_builds
```

> Note: PyInstaller will create 32 or 64-bit executables depending on the version of Python you have installed.

Next, build the Vue.js code. Open a command prompt and run:

```
npm run build
```

Finally, package the code for distribution using Electron Builder. Configuration is set in the package.json file. Results of the build will be placed in the /release/dist directory.

Create an installer:

```
npm run dist
```

Or, build a portable executable:

```
npm run dist-port
```

Or, pack the files into a directory:

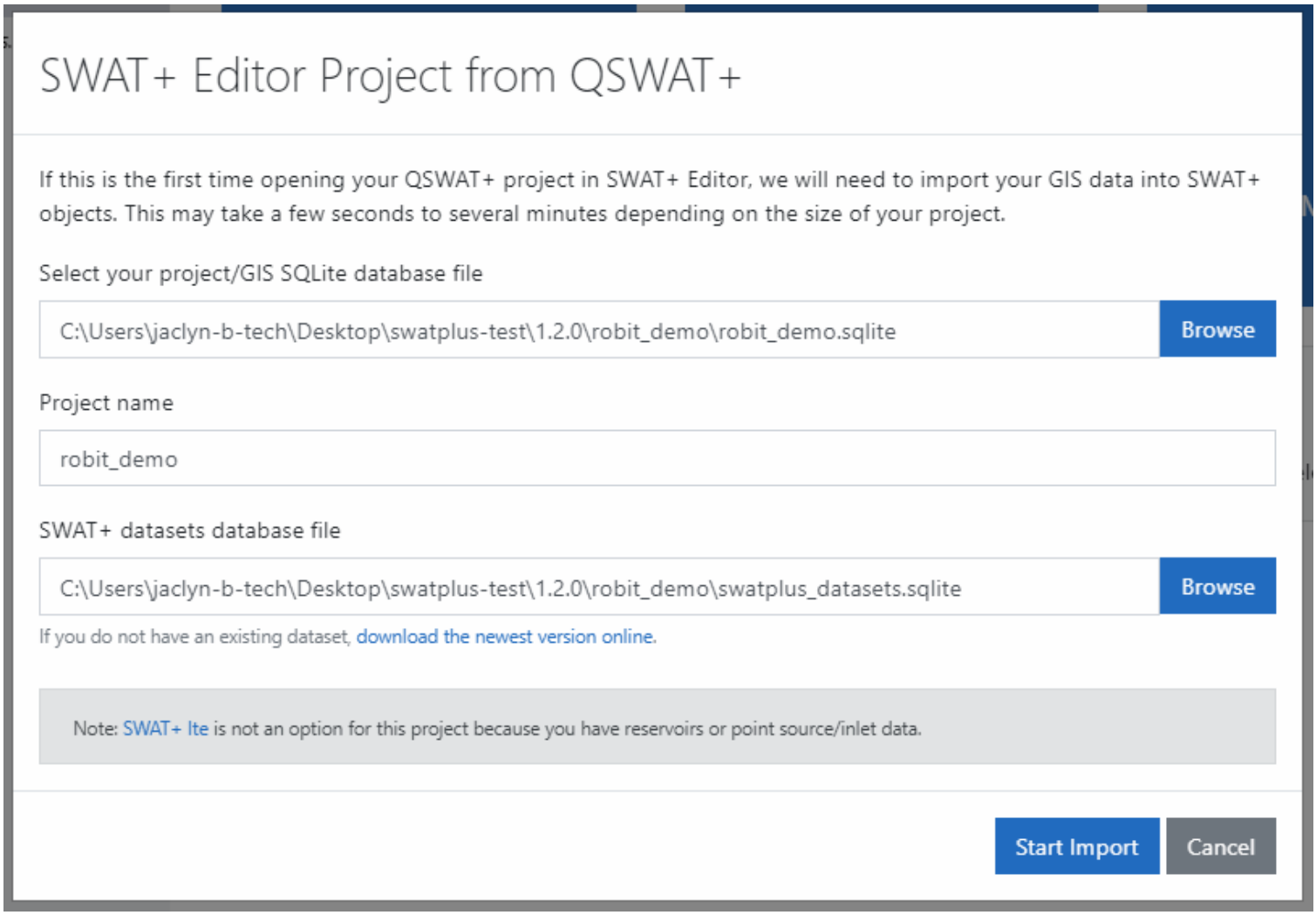
```
npm run pack
```

# SWAT+ Editor Documentation

Here are the articles in this section:

# Project Setup

When you open SWAT+ Editor, you are taken to the project setup screen. If you are coming from QSWAT+, an overlay will appear with the paths to your project databases. You may edit these as needed, and click the start import button.




When your project is done importing from GIS, it will be selected as your current project and displayed in the recent projects sidebar on the left as well as in the center screen.

From here you can start editing your SWAT+ inputs by clicking the button in the middle, or by clicking the paper icon in the far left blue-colored menu.

# Current Project: demo1

C:\Users\jaclyn-b-tech\Desktop\swatplus-test\demo1_0\demo1.json

Start editing SWAT+ inputs | Edit file and database settings...

# SWAT+ lte

SWAT+ lte is a version of the SWAT+ model that greatly simplifies hydrology and plant growth and does not simulate nutrients, concentrating on gully formation and stream degradation. It only uses channel and HRU objects, so this option is not available if you have point source or reservoirs.

If your project in QSWAT+ does not have point source or reservoirs, you will have an option to set up your project for SWAT+ lte by checking the box. Otherwise, this option is hidden.

## Other options for opening a project

If you are not coming from QSWAT+, you may open the editor and choose from one of three options:


Open project allows you to select an existing SWAT+ Editor project settings file and load it into the editor. Please note that this is an existing **editor** project, not any SWAT+ project. You must have an existing .json file for this option to work.

Click QSWAT+ project to select your QSWAT+ project database file. From here the steps are the same as described above when coming directly from QSWAT+.

Click new empty project to start a SWAT+ project from scratch, or if you have an existing SWAT+ project database but no settings file. If creating from scratch, leave the project database field blank and it will be created for you.

# Edit SWAT+ Inputs

Click the paper icon in the leftmost blue toolbar to enter the editing section. Most editors in this section are a literal representation of the SWAT+ input files. The collapsible dark-gray headings on the left correspond to the section lines in the master watershed file (file.cio).

When you click on an editor section from the left menu, you'll find the default SWAT+ file name with which the section corresponds. This enables you to quickly look up further information in the SWAT+ input/output documentation.

# Navigating the Editor

Most data is presented in a tabular format. When you click a row, you're presented with a form where you can make changes and save. The following features are common across many editor sections.

## Action Bar

At the top of most tables, you'll see the following action bar. Not all sections will have each option available.


Click create new record to add an item to the table. Edit multiple records allows you to set a single value for one or more fields across all or selected rows in your table.

In the search box, start typing the name of the object you want to view. Matching options will appear below the text box. Click the one you want, then click the search icon button to the right.

The import/export data button allows you to quickly access your data in CSV (comma-separated values spreadsheet) format, in most cases. We recommend exporting your data (or empty table is okay) first to get a template with the column names. You may then modify the file and import it back into the editor.

## Tables

Sort by a column in the table by clicking on the heading name. It will toggle ascending or descending direction as indicated by the arrows next to the name.

Tables with many records can be scrolled and then paged by clicking the page number or arrow links at the bottom of the table.

Each row may contain an edit/view icon on the far left to access the data in the row, and a delete icon on the far right (may need to scroll to access the far right of the table). We do not recommend deleting rows unless you are absolutely sure they are not used elsewhere in your model. Due to the relationships of data in SWAT+, deleting records could have unintended effects and break your model. Deleting cannot be undone; if in doubt, make a backup of your project SQLite database first.

## Forms

Most objects in SWAT+ have a name field and are identified using this name. Names should be unique and not contain spaces (spaces will be automatically converted to underscores).

Each form will have a save changes button toward the bottom. Be sure to click this button after making any changes and before leaving the form.

**Look-up Fields**

There are a lot of relationships between objects in SWAT+. For example, all fields in your channel properties table link to rows in other tables. In SWAT+ Editor forms, you can easily select these related rows by starting to type an object's name and select it as it pops up. If you accidentally enter an incorrect name, the editor will return an error stating the record does not exist in your database.


Example look-up form field

**Tip:** due to all of these relationships, it may be necessary to add data in a seemingly backwards manner. For example, you will need to add channel hydrology data before adding a row in channel properties, and channel properties will need to be added before a channel connection object. In most cases, these connection objects are imported from GIS so it is not an issue, but it is good to be aware.

## Getting Started

We recommend starting in the climate section, and importing your weather generators and observed weather data. If you're coming from GIS, when you import weather generators or observed data, it will create weather stations and match them to your spatial objects automatically.

# Climate

Weather generator data and weather stations are required for SWAT+ to run.

---

## Weather Stations

Weather stations are linked from all of your connection objects (channels, HRUs, etc.) in SWAT+. If you are coming from QSWAT+, it is much better to import stations either from the weather generator section, or the observed weather file importer than it is to create them manually.

By importing through one of the methods described below, your new stations will be automatically matched your spatial connection objects.

# Import Weather Generator Data

Click the import data button to import weather generator (wgn) data for your project. If you installed the SWAT+ databases, this file will be selected by default along with the CFSR world table. USA wgn data is also available from this database; type wgn_us to use this table.

You may also add your own data to this database using the wgn and corresponding wgn_mon tables.

→ How to Use SQLite /user/how-to-use-sqlite

Below the table name field is a check box asking if you are using observed weather data. By default (unchecked), when you click start import, weather stations will be created based on your wgn locations. If you are using observed weather data and prefer to have weather stations created based on this data, check this box--stations will not be created when you start import, and instead they will be created for you when you import your observed weather data files.

If you are not using observed weather data, it is important to leave the box unchecked so that weather stations are created for you.

## CSV Import

If you do not want to use the SQLite database, you may import CSV files of your weather generator data. Two CSV files are required.

1. Stations CSV file:
    - Columns `id, name, lat, lon, elev, rain_yrs`
    - `id` should be uniquely numbered
2. Monthly values CSV file:
    - Columns

      `id, wgn_id, month, tmp_max_ave, tmp_min_ave, tmp_max_sd, tmp_min_sd, pcp_ave, pcp_sd, pcp_skew, wet_dry, wet_wet, pcp_days, pcp_hhr, slr_ave, dew_ave, wnd_ave`
    - `id` should be uniquely numbered
    - `wgn_id` corresponds to the `id` column from the stations file

# Import Observed Weather Data

Import observed weather data from the top of the weather stations section. The data files may be in one of two formats: SWAT2012/Global Weather Data CFSR website format, or SWAT+.

> After importing observed weather data, be sure to modify your simulation run time to match your weather dates.

## SWAT2012/Global Weather Data CFSR Website Format

Each measurement included in your data must have the following entry file names:

| Measurement | Entry File |
|---|---|
| Precipitation | pcp.txt |
| Temperature | tmp.txt |
| Solar radiation | solar.txt |
| Relative humidity | rh.txt |
| Wind speed | wind.txt |

Each entry file is a comma-separated list of stations. Each station name should have a corresponding .txt file (e.g., name p326-963 should have a p326-963.txt file).

| ID | Name | Latitude | Longitude | Elevation |
|---|---|---|---|---|
| 1 | p326-963 | 32.628 | -96.250 | 142.0 |

Each station file should have the first line as the starting day as YYYYMMDD (e.g., 19790101). The following lines are the measurement for each day, one line per day. For temperature, each line will be max,min (e.g., 10.138,-2.662).

Weather data may be downloaded from the Global Weather CFSR website.

## SWAT+ Format

Each measurement included in your data must have the following entry file names:

| Measurement | Entry File |
|---|---|
| Precipitation | pcp.cli |
| Temperature | tmp.cli |
| Solar radiation | slr.cli |
| Relative humidity | hmd.cli |
| Wind speed | wnd.cli |

Each entry file has a title line (any text allowed), followed by a heading line, followed by a list of filenames for each station. Filenames should be listed alphabetically.

| pcp.cli: precipitation file names |
|---|
| filename |
| p326953.pcp |
| p326956.pcp |

Each station file has a title line, followed by a heading line and data line for time and location. Measurements for each timestep are in the lines to follow. For temperature, the measurements will be listed as max then min.

```
p326953.pcp: Precipitation data - file written by SWAT+ editor
nbyr     tstep       lat       lon      elev
  36         0    32.628   -95.313   121.000
1979    1    5.382
1979    2    0.003
1979    3    0.000
1979    4    3.703
1979    5   12.147
1979    6   27.241
1979    7    0.774
1979    8    0.110
1979    9    0.000
1979   10    4.825
```

# Weather Generator Parameters

| SWAT+ Input File | Database Tables |
|---|---|
| weather-wgn.cli | weather_wgn_cli |
| | weather_wgn_cli_mon |

## weather_wgn_cli

| Field | Type | Description | Units | Range |
|---|---|---|---|---|
| id | int | Auto-assigned identifier | | |
| name | text | Name of weather generator | | |
| lat | real | Latitude of weather station | deg. | +/-90 |
| lon | real | Longitude of weather station | deg. | +/-180 |
| elev | real | Elevation of weather station | m | 0-5000 |
| rain_yrs | int | Number of years of recorded maximum monthly 0.5h rainfall data | | 5-100 |

## weather_wgn_cli_mon

Each entry in weather_wgn_cli will have 12 rows in weather_wgn_cli_mon.

| Field | Type | Description | Units | Range |
|---|---|---|---|---|
| id | int | Auto-assigned identifier | | |
| wgn_id | int | ID of row in weather_wgn_cli | | |
| month | int | Month | | |
| tmp_max_ave | real | Average or mean daily maximum air temperature for month | °C | -30-50 |
| tmp_min_ave | real | Average or mean daily minimum air temperature for month | °C | -40-40 |

| | | | | |
|---|---|---|---|---|
| tmp_max_sd | real | Standard deviation for daily maximum air temperature in month | °C | 0.1-100 |
| tmp_min_sd | real | Standard deviation for daily minimum air temperature in month | °C | 0.1-30 |
| pcp_ave | real | Average or mean total monthly precipitation | mm | 0-600 |
| pcp_sd | real | Standard deviation for the average daily precipitation | mm/day | 0.1-50 |
| pcp_skew | real | Skew coefficient for the average daily precipitation | mm | -50-20 |
| wet_dry | real | Probability of a wet day after a dry day | | 0-0.95 |
| wet_wet | real | Probability of a wet day after a wet day | | 0-0.95 |
| pcp_days | real | Average number of days of precipitation in a month | | 0-31 |
| pcp_hhr | real | Maximum 0.5 hour rainfall in entire period of record for month | mm | 0-125 |
| slr_ave | real | Average daily solar radiation for the month | MJ/m^2/day | 0-750 |
| dew_ave | real | Average daily dew point temperature for each month | °C | -50-25 |
| wnd_ave | real | Average wind speed for the month | m/s | 0-100 |

# Weather Stations Parameters

| SWAT+ Input File | Database Tables |
|---|---|
| weather-sta.cli | weather_sta_cli |
| | weather_file |

## weather_sta_cli

When entering an observed weather file name in the station editor, you may start typing to search for existing weather files adding during the import step. If adding observed files manually, just type the name of the file (e.g., p326953.pcp), and put that file in the directory you plan to write input files (e.g., your TxtInOut). Files must be in SWAT+ format. If your weather data is in SWAT2012 format or from the Global Weather CFSR website, please use the import step to convert them to SWAT+.

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| name | text | Name of the weather station |
| wgn_id | int | ID of row in weather_wgn_cli |
| pcp | text | Precipitation gage filename or "sim" if simulated |
| tmp | text | Temperature gage filename or "sim" if simulated |
| slr | text | Solar radiation gage filename or "sim" if simulated |
| hmd | text | Relative humidity gage filename or "sim" if simulated |
| wnd | text | Wind speed gage filename or "sim" if simulated |
| wnd_dir | text | Wind direction gage filename |
| atmo_dep | text | Atmospheric deposition data file name |
| lat | real | Latitude of weather station |
| lon | real | Longitude of weather station |

## weather_file

This table is only used if you import observed weather data files. If entering stations manually, this table will not be populated.

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| filename | text | Name of the weather data file |
| type | text | Type of weather data: pcp, tmp, slr, hmd, wnd |
| lat | real | Latitude of weather station in file |
| lon | real | Longitude of weather station in file |

# Simulation

## Time

Configure the number or years to run the simulation and time step. If you have observed weather data, make sure your simulation time falls within these dates.

| SWAT+ Input File | Database Table |
| --- | --- |
| time.sim | time_sim |

### time_sim

| Field | Type | Description |
| --- | --- | --- |
| id | int | Auto-assigned identifier |
| day_start | int | Beginning Julian day of simulation<br>If zero, the model starts the simulation on January 1 |
| day_end | int | Ending Julian day of simulation<br>If zero, the model ends the simulation on December 31 |
| step | int | Time steps in a day for rainfall, runoff and routing |

Configure the output files to print.

| SWAT+ Input File | Database Tables |
| --- | --- |
| print.prt | print_prt |
| | print_prt_object |

## print_prt

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| nyskip | int | Number of years to *not* print output |
| day_start | int | Beginning Julian day of simulation to start printing output files for daily printing only |
| yrc_start | int | Beginning year of simulation to start printing output files |
| day_end | int | Ending Julian day of simulation to stop printing output files for daily printing only |
| yrc_end | int | Ending year of simulation to stop printing output files |
| interval | int | Daily print within the period (e.g., interval=2 will print every other day) |
| csvout | bool | Print .csv files in addition to text files |
| dbout | bool | Print database (not currently active) |
| cdfout | bool | Print netcdf (not currently active) |
| soilout | bool | Print soil nutrients carbon output file |
| mgtout | bool | Print management output file |
| hydcon | bool | Print hydrograph connect output file |
| fdcout | bool | Print flow duration curve output file |

## print_prt_object

Each row in print_prt_object represents an output file that can be print daily, monthly, yearly, and average annual output for each.

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| print_prt_id | int | ID of print_prt row |
| name | text | Name of print object |
| daily | bool | Print daily output |
| monthly | bool | Print monthly output |

| | | |
|---|---|---|
| yearly | bool | Print yearly output |
| avann | bool | Print average annual output |

> Daily printing of all files could cause very large output (exceeding hard drive space)

**Print Object Descriptions**

| Object Name | Description |
|---|---|
| basin_wb | Water balance basin output variables |
| basin_nb | Nutrient balance basin output variables |
| basin_ls | Losses basin output variables |
| basin_pw | Plant weather basin output variables |
| basin_aqu | Aquifer basin output variables |
| basin_res | Reservoir basin output file variables |
| basin_cha | Channel basin output file variables |
| basin_sd_cha | CHAN DEG basin output file variables |
| basin_psc | Point source basin output file variables |
| region_wb | Water balance region output variables |
| region_nb | Nutrient balance region output variables |
| region_ls | Losses region output variables |
| region_pw | Plant weather region output variables |
| region_aqu | Aquifer region output variables |
| region_res | Reservoir region output variables |
| region_cha | Channel region output variables |
| region_sd_cha | SWAT DEG Channel region output variables |
| region_psc | Point source region output variables |

| | |
|---|---|
| lsunit_wb | Water balance routing unit output variables |
| lsunit_nb | Nutrient balance routing unit output variables |
| lsunit_ls | Losses routing unit output variables |
| lsunit_pw | Plant weather routing unit output variables |
| hru_wb | Water balance hru output variables |
| hru_nb | Nutrient balance hru output variables |
| hru_ls | Losses hru output variables |
| hru_pw | Plant weather hru output variables |
| hru-lte_wb | Water balance HRU-LTE output variables |
| hru-lte_nb | Nutrient balance HRU-LTE output variables |
| hru-lte_ls | Losses HRU-LTE output variables |
| hru-lte_pw | Plant weather HRU-LTE output variables |
| channel | Channel output variables |
| channel_sd | SWAT DEG (lte) channel output variables |
| aquifer | Aquifer output variables |
| reservoir | Reservoir output variables |
| recall | Recall output variables |
| hyd | Hydin output and hydout_output variables |
| ru | Routing unit output variables |
| pest | Pesticide constituents outputs |

# Basin

General watershed attributes are defined in the basin input files: codes and parameters. These attributes control a diversity of physical processes at the watershed level. The interfaces will automatically set these parameters to the default or recommended values listed in the variable documentation. Users can use the default values or change them to better reflect what is happening in a given watershed. Variables governing bacteria or pesticide transport need to be initialized only if these processes are being modeled in the watershed. Even if nutrients are not being studied in a watershed, some attention must be paid to these variables because nutrient cycling impacts plant growth which in turn affects the hydrologic cycle.

# Codes

| SWAT+ Input File | Database Table |
|---|---|
| codes.bsn | codes_bsn |

## codes_bsn

| Field | Type | Description |
|---|---|---|
| pet_file | text | Potential ET filename |
| wq_file | text | Watershed stream water quality filename |
| pet | int | Potential ET method code |
| event | int | Event code |
| crack | int | Crack flow code |
| rtu_wq | int | Subbasin water quality code |
| sed_det | int | Max half-hour rainfall frac calc |
| rte_cha | int | Water routing method |
| deg_cha | int | Channel degradation code |
| wq_cha | int | Stream water quality code |
| rte_pest | int | Redefined to the sequence number of pest in NPNO(:) to be routed through the watershed |
| cn | int | CN method flag |
| c_fact | int | C-factor |
| carbon | int | Carbon code |
| baseflo | int | Baseflow distribution factor during the day for subdaily runs |
| uhyd | int | Unit hydrograph method |
| sed_cha | int | Instream sediment model |

| | | |
|---|---|---|
| tiledrain | int | Tile drainage EQ code |
| wtable | int | Water table depth algorithms code |
| soil_p | int | Soil phosphorus model |
| abstr_init | int | Initial abstraction on impervious cover |
| atmo_dep | text | Atmospheric deposition code |
| stor_max | int | Max depressional storage selection code |
| headwater | int | Headwater code |

# Parameters

| SWAT+ Input File | Database Table |
|---|---|
| parameters.bsn | parameters_bsn |

## parameters_bsn

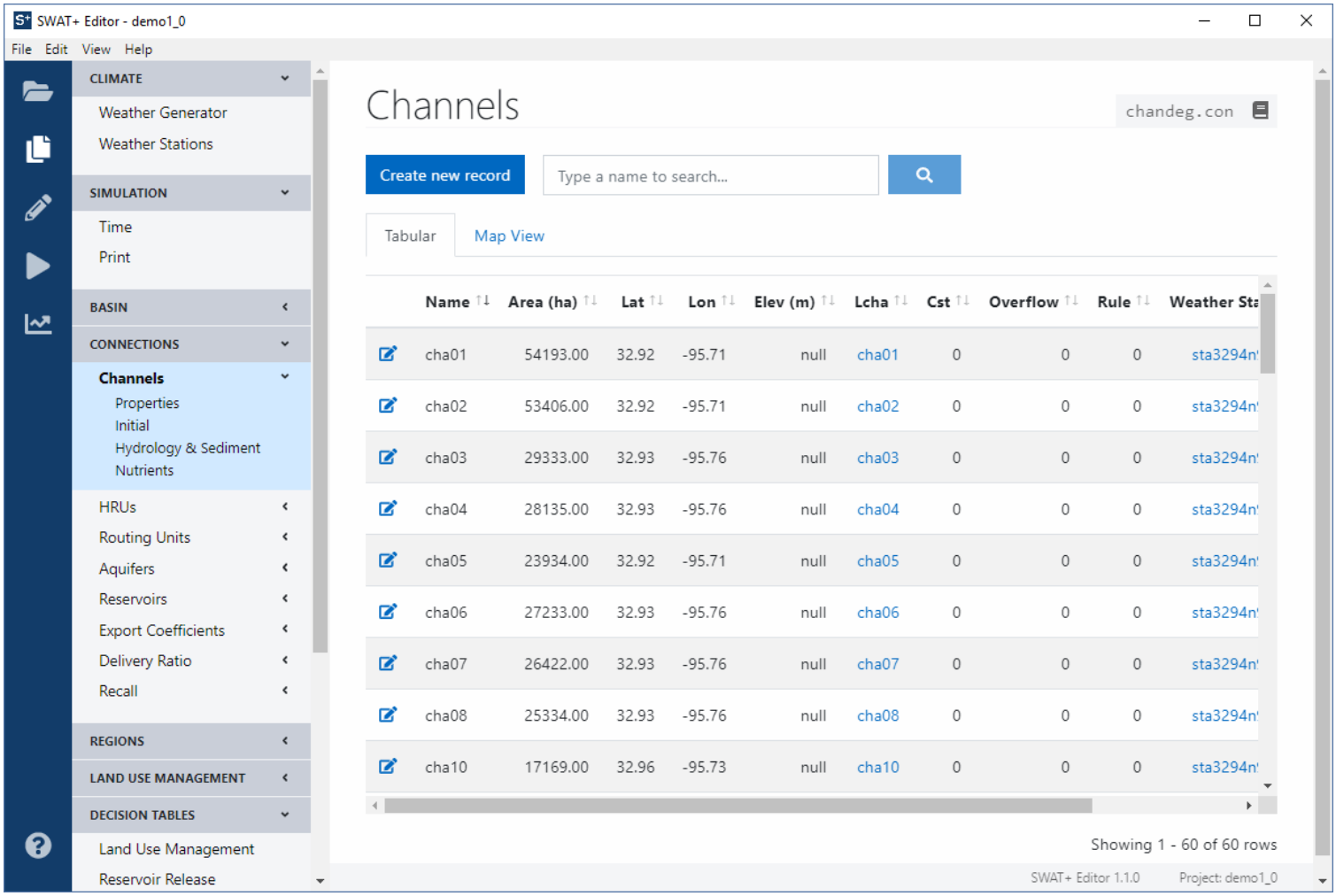
| Field | Type | Description | Units | Default | Range |
|---|---|---|---|---|---|
| lai_noevap | real | Leaf area index at which no evaporation occurs from water surface | | 3 | 0-10 |
| sw_init | real | Initial soil water storage expressed as a fraction of field capacity water content | | 0 | 0-1 |
| surq_lag | real | Surface runoff lag coefficient | | 4 | 1-24 |
| adj_pkrt | real | Peak rate adjustment factor for sediment routing in the subbasin (tributary channels) | | 1 | 0.5-2 |
| adj_pkrt_sed | real | Peak rate adjustment factor for sediment routing in the main channel | | 1 | 0-2 |
| lin_sed | real | Linear parameter for calculating the maximum amount of sediment that can be reentrained during channel sediment routing | | 0.0001 | 0.0001-0.01 |
| exp_sed | real | Exponent parameter for calculating sediment reentrained in channel sediment routing | | 1 | 1-1.5 |
| orgn_min | real | Rate factor for humus mineralization of active organic nutrients (N and P) | | 0.0003 | 0.001-0.003 |
| n_uptake | real | Nitrogen uptake distribution parameter | | 20 | 0-100 |
| p_uptake | real | Phosphorus uptake distribution parameter | | 20 | 0-100 |
| n_perc | real | Nitrate percolation coefficient | | 0.2 | 0-1 |
| p_perc | real | Phosphorus percolation coefficient | 10 m^3/M | 10 | 10-17.5 |

| | | | | | |
|---|---|---|---|---|---|
| p_soil | real | Phosphorus soil partitioning coefficient | m^3/Mg | 175 | 100-200 |
| p_avail | real | Phosphorus availability index | | 0.4 | 0.01-0.7 |
| rsd_decomp | real | Residue decomposition coefficient | | 0.05 | 0.02-0.1 |
| pest_perc | real | Pesticide percolation coefficient | | 0.5 | 0-1 |
| msk_co1 | real | Calibration coefficient to control impact of the storage time constant for the reach at bankfull depth | | 0.75 | 0-10 |
| msk_co2 | real | Calibration coefficient used to control impact of the storage time constant for low flow (where low flow is when river is at 0.1 bankfull depth) upon the km value calculated for the reach | | 0.25 | 0-10 |
| msk_x | real | Weighting factor control relative importance of inflow rate and outflow rate in determining storage on reach | | 0.2 | 0-0.3 |
| trans_loss | real | Fraction of transmission losses from main channel that enter deep aquifer | | 0 | 0-1 |
| evap_adj | real | Reach evaporation adjustment factor | | 0.6 | 0.5-1 |
| cn_co | real | Currently not being used | | | |
| denit_exp | real | Denitrification exponential rate coefficient | | 1.4 | 0-3 |
| denit_frac | real | Denitrification threshold water content | | 1.3 | 0-1 |
| man_bact | real | Fraction of manure applied to land areas that has active colony forming units | | 0.15 | 0-1 |
| adj_uhyd | real | Adjustment factor for subdaily unit hydrograph basetime | | 0 | 0-1 |
| cn_froz | real | Parameter for frozen soil adjustment on infiltration/runoff | | 0.000862 | 0-0 |
| dorm_hr | real | Time threshold used to define dormancy | hrs | 0 | 0-24 |
| s_max | real | Currently not being used | | | |

| | | | | | |
|---|---|---|---|---|---|
| n_fix | real | Nitrogen fixation coefficient | | 0.5 | 0-1 |
| n_fix_max | real | Maximum daily-n fixation | kg/ha | 20 | 1-20 |
| rsd_decay | real | Minimum daily residue decay | | 0.01 | 0-0.05 |
| rsd_cover | real | Residue cover factor for computing fraction of cover | | 0.3 | 0.1-0.5 |
| vel_crit | real | Critical velocity | | 5 | 0-10 |
| res_sed | real | Reservoir sediment settling coefficient | | 0.184 | 0.09-0.27 |
| uhyd_alpha | real | Alpha coefficient for gamma function unit hydrograph | | 5 | 0.5-10 |
| splash | real | Splash erosion coefficient | | 1 | 0.9-3.1 |
| rill | real | Rill erosion coefficient | | 0.7 | 0.5-2 |
| surq_exp | real | Exponential coefficient for overland flow | | 1.2 | 1-3 |
| cov_mgt | real | Scaling parameter for cover and management factor for overland flow erosion | | 0.03 | 0.001-0.45 |
| cha_d50 | real | Median particle diameter of main channel | mm | 50 | 10-100 |
| cha_part_sd | real | Geometric standard deviation of particle size | | 1.57 | 1-5 |
| adj_cn | real | Currently not being used | | | |
| igen | int | Random generator code 0 = use default number; 1 = generate new numbers in every simulation | | 0 | 0-1 |

# Connections

The connections section contains all spatial object connectivity for the simulation run. In SWAT+ Editor, all connection object properties can be set through this section. For example, when you click on channels, you will see additional menu links appear for: properties, initialization, hydrology and sediment, and nutrients.




Example: channel connection object main page

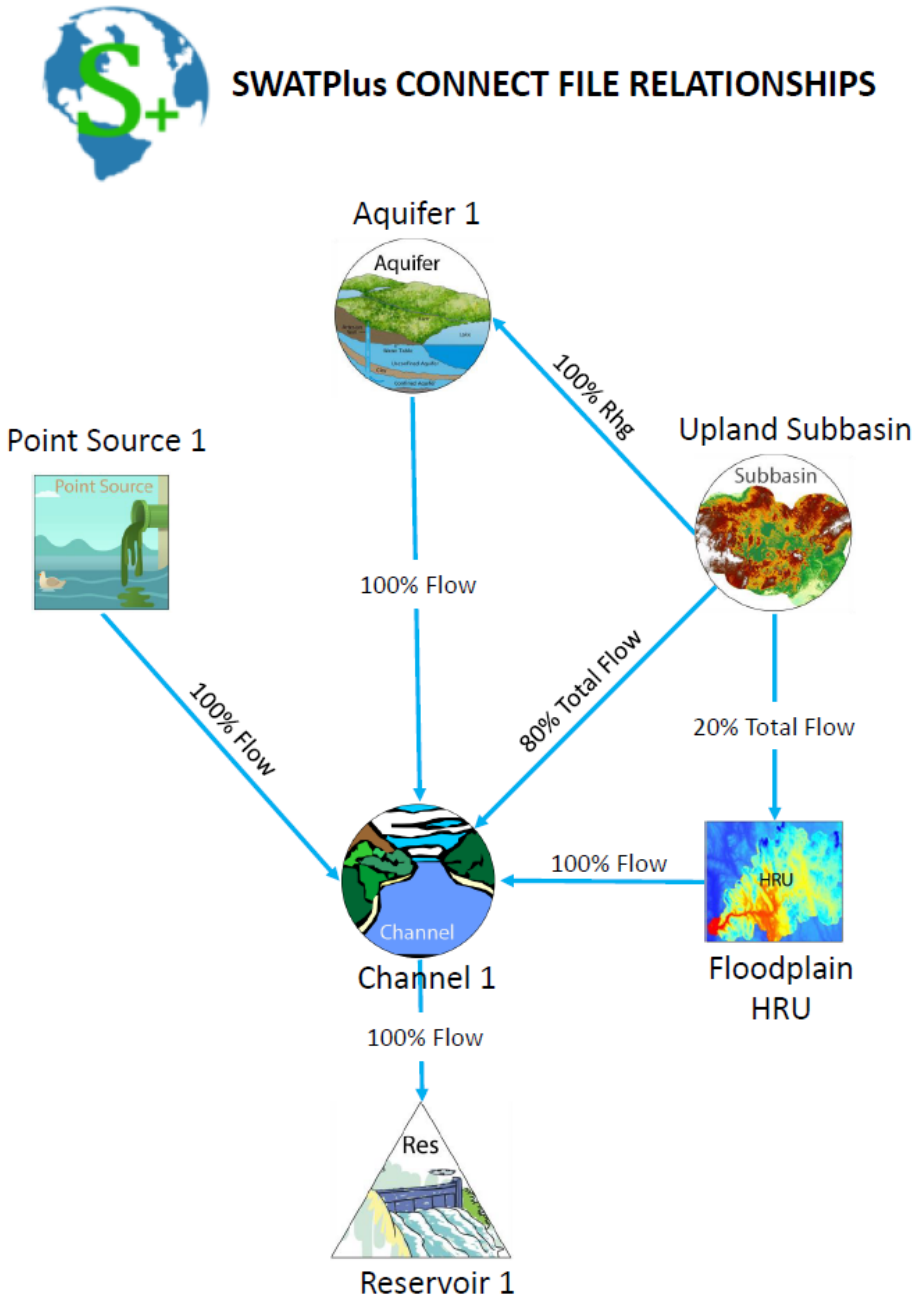
All connection objects have a similar format as seen in the above figure. The tabular view is shown by default. Click the map view tab to see a map with markers for the center coordinates of each object. Click an object marker in the map view, or edit icon on the left side of a row in the table to view or edit the object.

Each connection object will have a main properties object associated with it as well as a weather station. Click on these names in the table, or from the edit view page, click the button next to their names to view information about the properties object or weather station.

Each connection object may have outflow. This can be viewed in the table by clicking the eye icon in the rightmost column, or view from the edit page.

If you imported your project from GIS, your connection objects are populated automatically during project setup.

# Explanation of SWAT+ Spatial Objects




Example connectivity

## Subbasin

The subbasin is defined by the DEM in the GIS interface as it always has been. All flow within the subbasin drains to the subbasin outlet.

## Landscape Unit

A landscape unit (LSU) is defined as a collection of HRUs and can be defined as a subbasin, or it could be a flood plain or upland unit, or it could be a grid cell with multiple HRUs. The landscape unit is not routed, it only used for output. The landscape unit output files (waterbal, nutbal, losses, and plant weather) are output for HRUs, landscape units, and for the basin. Two input files are required: 1) landscape elements and, 2) landscape define. The elements file includes HRUs and their corresponding LSU fraction and basin fractions. The define file specifies which HRUs are contained in each LSU.

## Routing Unit

A routing unit is a collection of hydrographs that can be routed to any spatial object. The routing unit can be configured as a subbasin, then total flow (surface, lateral and tile flow) from the routing unit can be sent to a channel and all recharge from the routing unit sent to an aquifer. This is analogous to the current approach in SWAT. However, SWAT+ gives us much more flexibility in configuring a routing unit. For example, in CEAP, we are routing each HRU (field) through a small channel (gully or grass waterway) before it reaches the main channel. In this case, the routing unit is a collection of flow from the small channels. We also envision simulating multiple representative hillslopes to define a routing unit. Also, we are setting up scenarios that define a routing unit using tile flow from multiple fields and sending that flow to a wetland.

The routing unit is the spatial unit SWAT+ that allows us to lump outputs and route the outputs to any other spatial object. It gives us considerably more flexibility than the old subbasin lumping approach in SWAT, and will continue to be a convenient way of spatial lumping until we can simulate individual fields or cells in each basin.

# Channels

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

# HRUs

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

# Routing Units

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

# Aquifers

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

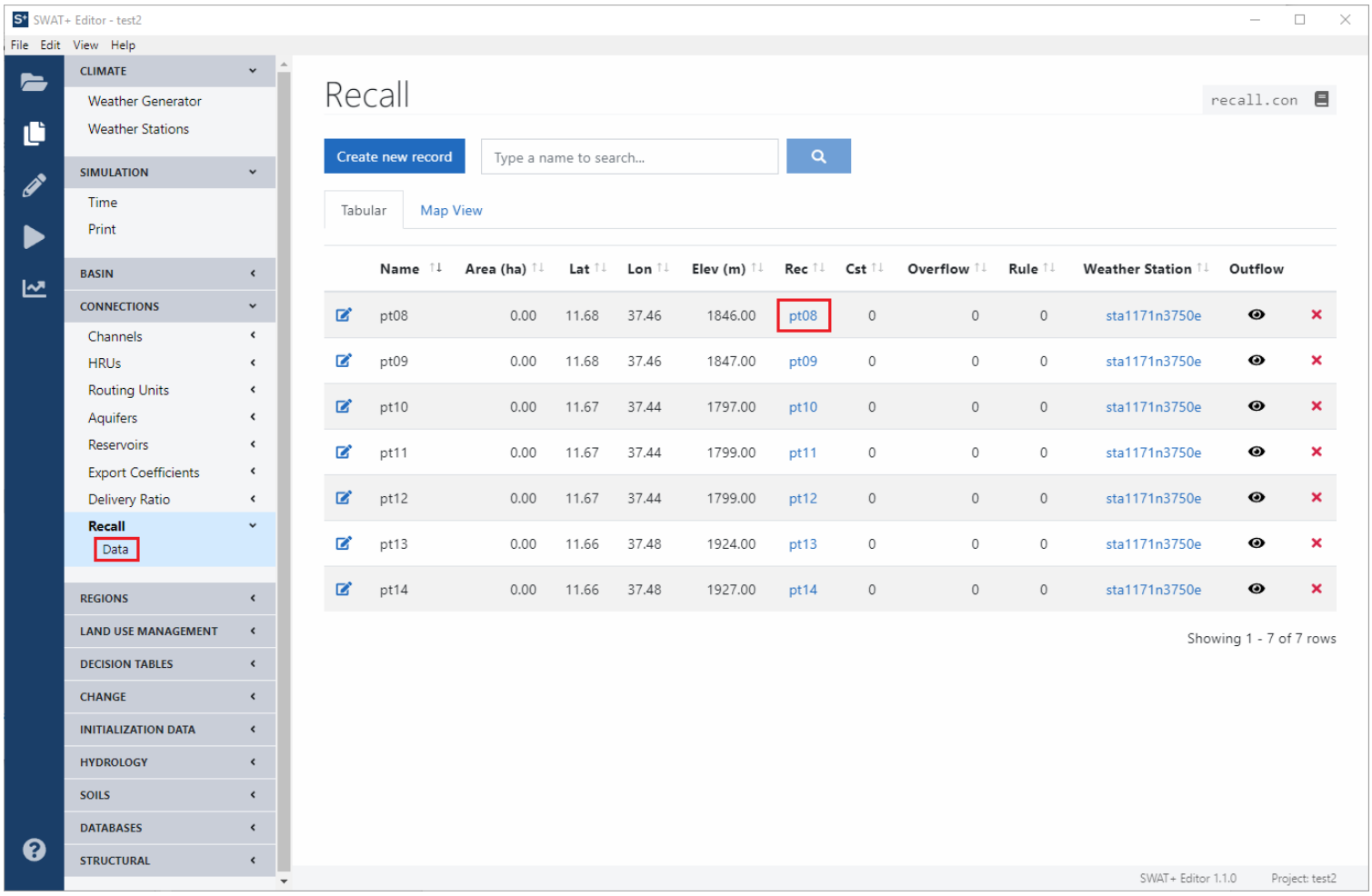
# Reservoirs

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

# Recall (Point Source/Inlet)

Recall objects are used for connecting point source or inlet data to your watershed. If you added point source in QSWAT+, when you import your project into SWAT+ Editor it will be connected via the recall section.

By default, constant data with all zero values during the default simulation period is added. To add your own recall data, click the recall item in the edit menu under connections. Click the item name under the rec column in the connection object table, or click the data item under recall in the edit menu on the left.

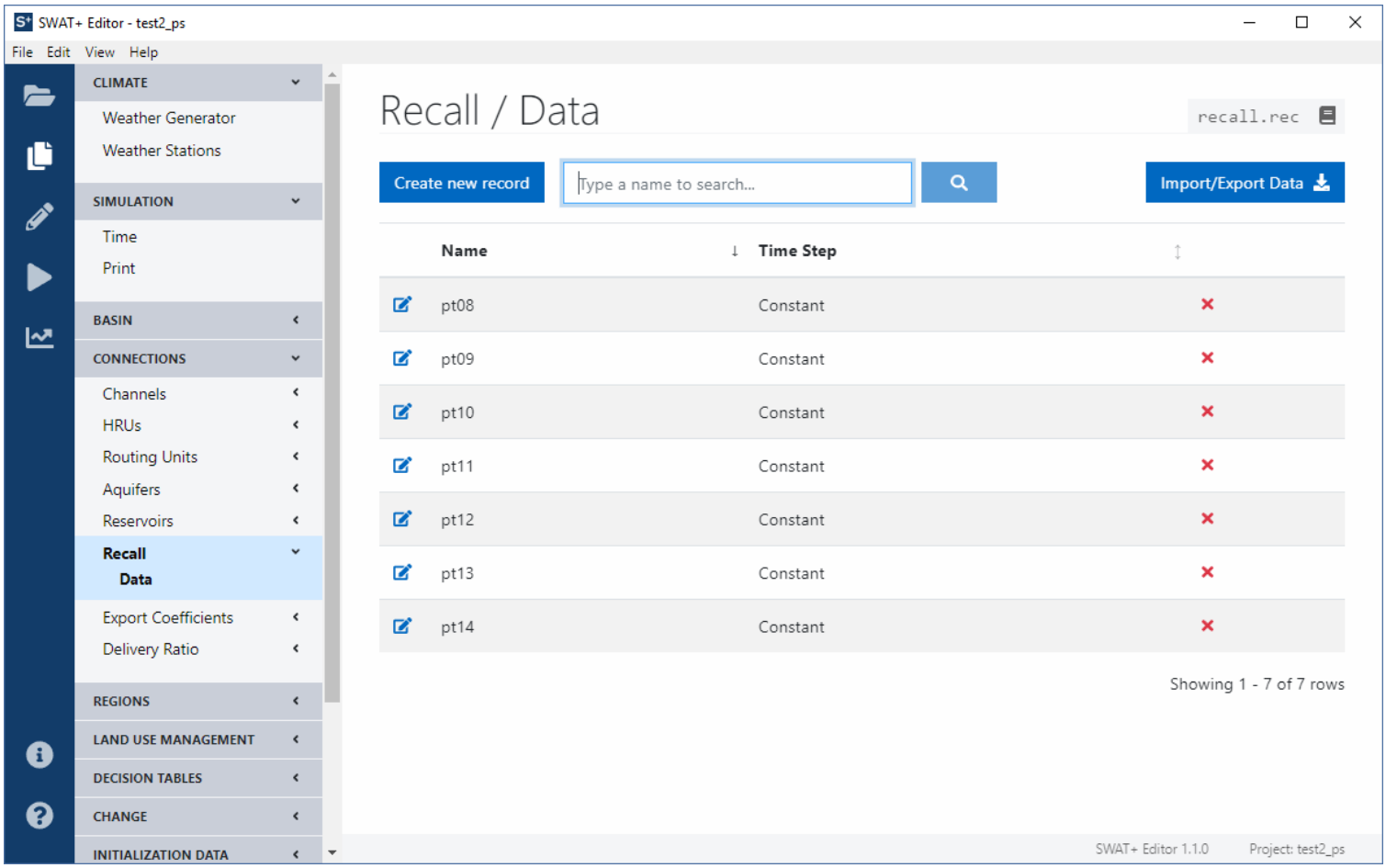



The red boxes highlight where to click to access recall data

# Constant Data

By default, your recall data is imported as constant. To insert your values, you can edit each item individually by clicking the edit button and manually entering each value. Alternatively, you may upload a CSV of your data.

From the recall data section, click the import/export button in the top right corner.




Export is selected by default. Choose a file name, and click the export CSV file button to get a template for your data.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | name | flo | sed | ptl_n | ptl_p | no3_n | sol_p | chla | nh3_n | no2_n | cbn_bod | oxy | sand | silt | clay | sm_agg | lg_agg | gravel | tmp |
| 2 | pt08 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | pt09 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | pt10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | pt11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | pt12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | pt13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | pt14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Example template file after exporting

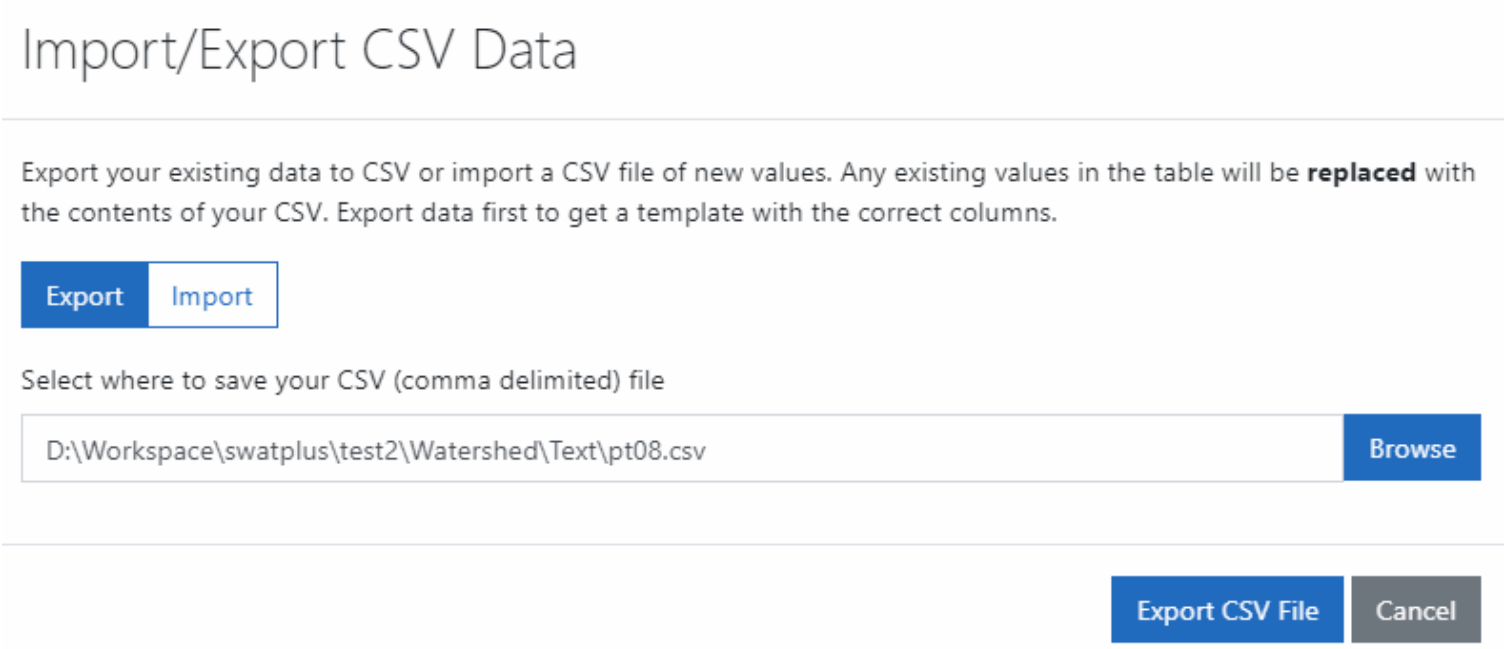
Edit the CSV as needed, save, and then go back to the editor and click the import/output button again. This time toggle the import button. Choose your modified CSV file and click the import CSV data button.

Your updated values will appear in the table.

## Time Series Data

By default recall data is imported as constant, however this can be changed by clicking the edit button next to a row in the recall data table. Select the new time step for your data: daily, monthly, or yearly. Click the save changes button. Next, click the import/export button that appears on the form.

Export is selected by default. Choose a file name, and click the export CSV file button.




Open the file after it is exported to see the template for your data. Modify your data as needed matching the time step you selected previously. Be sure the years match your simulation run time. In a yearly time step, t_step equals 1 through number of years. For monthly data, t_step equals the number of the month, and for daily it is the number of the day of the year.

To import your data, click the import/export data button again and this time click to toggle import. Choose your file and click import CSV file. Your new data will appear in the table.

# Recall Table Definitions

| SWAT+ Input File | Database Table |
|---|---|
| recall.rec | recall_rec |

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| name | text | Name of recall object |
| rec_typ | int | Time step for recall object (1-daily, 2-monthly, 3-yearly) |

Each record in recall_rec will have a data file named {name}.rec. All of this data is stored in a single recall_dat table in the database.

| SWAT+ Input File | Database Table |
|---|---|
| {name}.rec | recall_dat |

| Field | Type | Description | Units |
|---|---|---|---|
| yr | int | Year | |
| t_step | int | Timestep | |
| flo | real | Volume of water | m^3 |
| sed | real | Sediment | metric ton |
| ptl_n | real | Organic nitrogen | kg N |
| ptl_p | real | Organic phosphorus | kg P |
| no3_n | real | Nitrate | kg N |
| sol_p | real | Mineral (soluble P) | kg P |
| chla | real | Chlorophyll-a | kg |
| nh3_n | real | Ammonia | kg N |

| | | | |
|---|---|---|---|
| no2_n | real | Nitrogen dioxide | kg N |
| cbn_bod | real | Carbonaceous biological oxygen demand | kg |
| oxy | real | Dissolved oxygen | kg |
| sand | real | Detached sand | |
| silt | real | Detached silt | |
| clay | real | Detached clay | |
| sm_agg | real | Detached small ag | |
| lg_agg | real | Detached large ag | |
| gravel | real | Gravel | |
| tmp | real | Temperature | deg c |

# Export Coefficients

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

## Special note about using export coefficients with constant point source/inlet data

In SWAT+, constant values for point sources and inlets are stored in the export coefficients properties file, exco.exc, while time series data are stored entirely in the recall section.

However, in the editor, we keep both constant and time series point sources and inlets in the **recall section**. When you write input files, the editor will write to the exco.exc and exco_om.exc files appropriately.

# Delivery Ratio

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

# Regions

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

SWAT+ IO Documentation /user/io

# Land Use Management

A primary goal of environmental modeling is to assess the impact of human activities on a given system. Central to this assessment is the itemization of the land and water management practices taking place within the system. This section contains input data for planting, harvest, irrigation applications, nutrient applications, pesticide applications, and tillage operations. Information regarding tile drains and urban areas is also stored in this file.

| SWAT+ Input File | Database Table |
|---|---|
| landuse.lum | landuse_lum |
| management.sch | management_sch |
| | management_sch_auto |
| | management_sch_op |
| cntable.lum | cntable_lum |
| ovn_table.lum | ovn_table_lum |
| cons_practice.lum | cons_practice_lum |

In addition to the above, SWAT+ Editor groups the operations databases in this section of the editor. However, within the SWAT+ master watershed file (file.cio), these are listed under the ops section.

| SWAT+ Input File | Database Table |
|---|---|
| graze.ops | graze_ops |
| harv.ops | harv_ops |
| irr.ops | irr_ops |
| sweep.ops | sweep_ops |
| fire.ops | fire_ops |
| chem_app.ops | chem_app_ops |

# Land Use Management

This section is the entry point for management data in SWAT+. It comprises cross-walks to several other sections of data.

This data is accessed from the HRU properties section (hru-data.hru).

## landuse_lum

| Field | Type | Description | Related Table |
|---|---|---|---|
| id | int | Auto-assigned identifier | |
| name | text | Name of the land use properties | |
| cal_group | text | Calibration group | |
| plnt_com_id | int | Plant community | plant_ini |
| mgt_id | int | Management schedule | management_sch |
| cn2_id | int | Curve number | cntable_lum |
| cons_prac_id | int | Conservation practices | cons_prac_lum |
| urban_id | int | Urban land use | urban_urb |
| urb_ro | text | Urban runoff | |
| ov_mann_id | int | Overland flow Manning's n | ovn_table_lum |
| tile_id | int | Tile drain | tiledrain_str |
| sep_id | int | Septic tank | septic_str |
| vfs_id | int | Filter strip | filterstrip_str |
| grww_id | int | Grassed waterway | grassedww_str |
| bmp_id | int | Best management practices | bmpuser_str |
| description | text | Optional description of the row | |

# Management Schedules

Management schedules comprise auto-schedules (decision tables) and/or operations schedules.

When you import your project from GIS, SWAT+ assigns auto-schedules for management based on your crop land use.

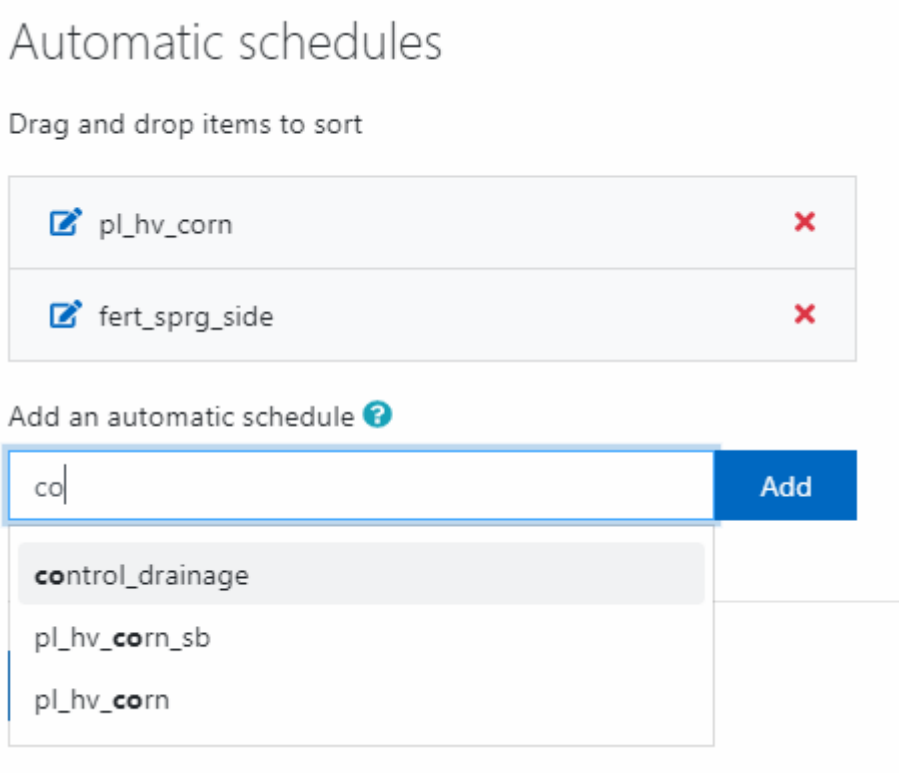
| Plant Type (in plants_plt) | Decision Table Template |
|---|---|
| warm_annual | pl_hv_corn |
| cold_annual | pl_hv_wwht |
| perennial | no management schedule |

For example, oats is a cold annual crop. If this crop is in your HRUs, a decision table named pl_hv_oats will be created based on the template of pl_hv_wwht when you import your data from GIS.

## Adding/Editing a Schedule

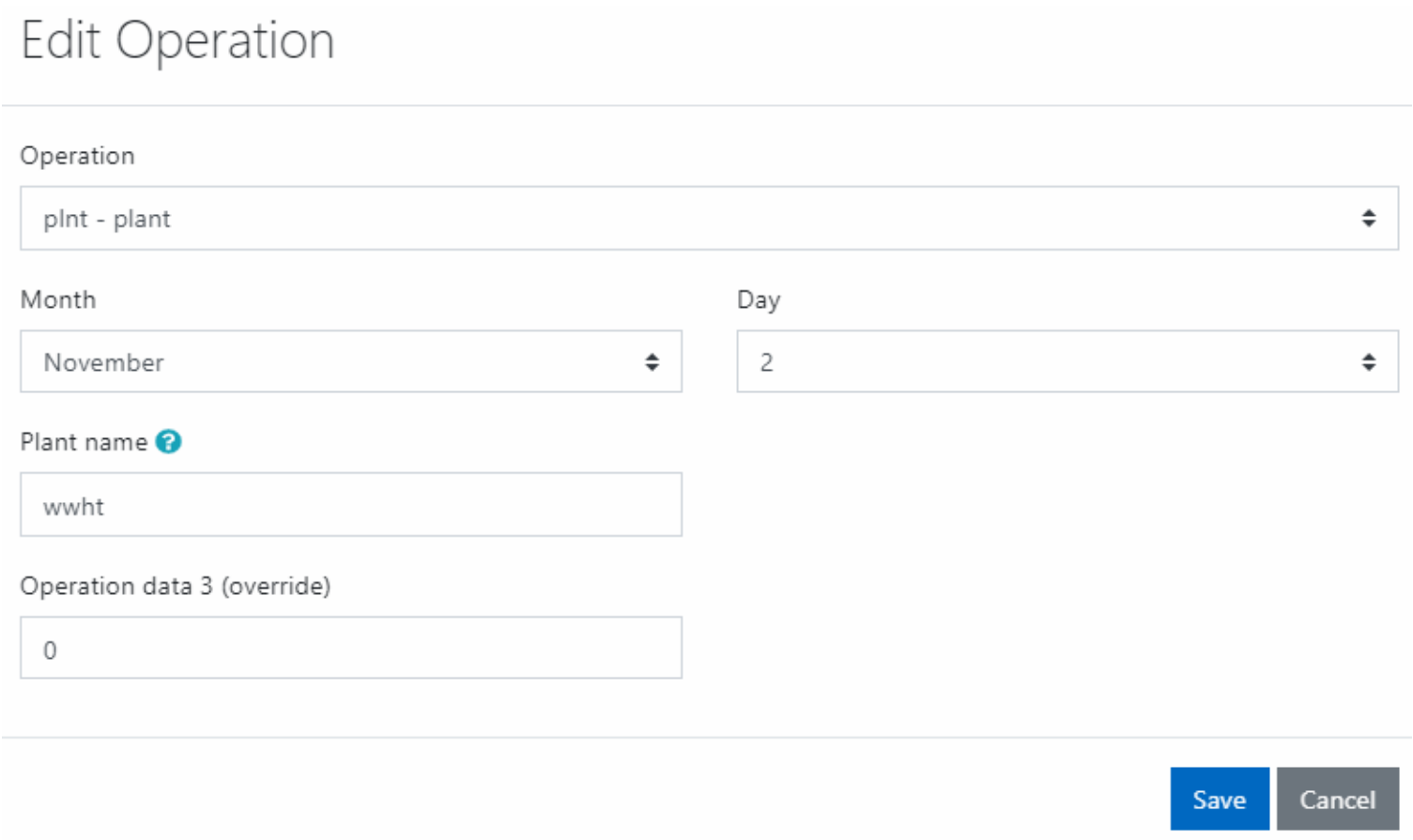
From the management schedules section, click create a new record or click edit on a row in the table. Give your schedule a unique name.

To add an automatic schedule, start typing a decision table name in the box provided. Click the desired result from the list of matches that pops up, and click the add button. If you enter more than one schedule, you can drag and drop to sort.

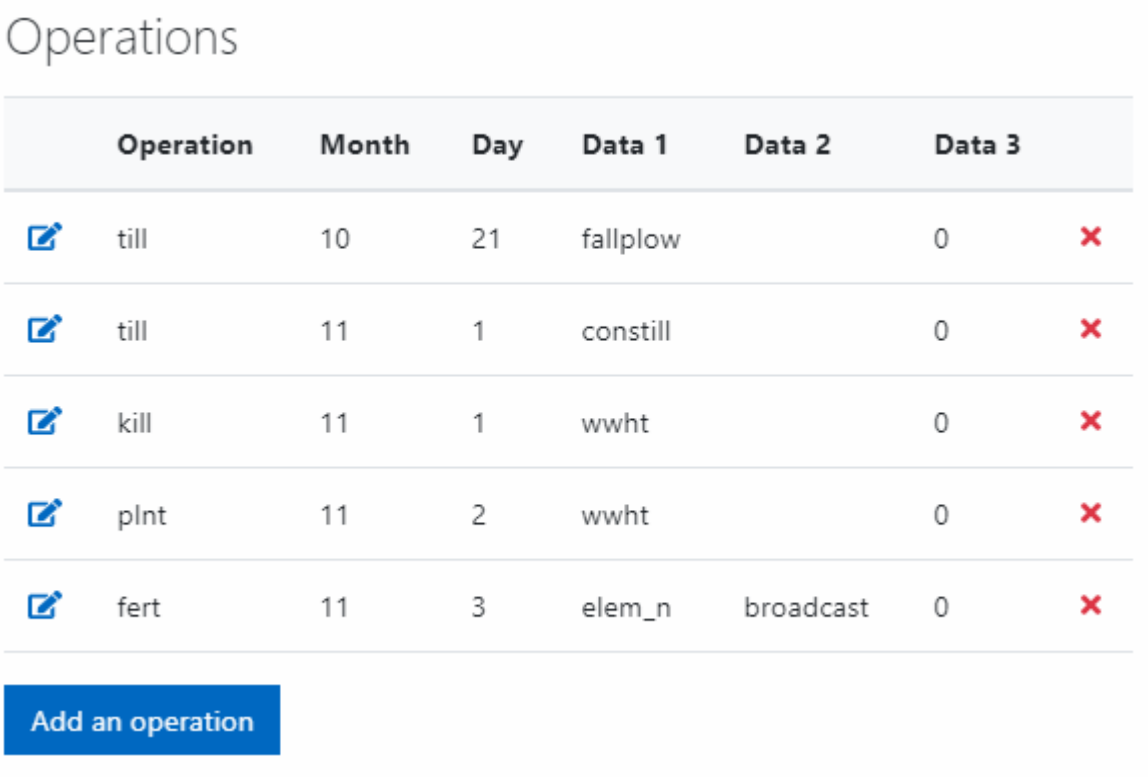



Example of adding automatic schedules

To add an operation, click the add operation button. Select your operation type from the form that pops up and complete the remaining fields. Click save when done to add the operation to your table.




Example of adding a plant operation




Example operations table

When you're done adding automatic schedules and operations, **click the save changes button** to save your management schedule.

## Table Parameters

**management_sch**

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| name | text | Name of the schedule |

**management_sch_auto**

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| management_sch_id | int | ID of management schedule |
| d_table_id | int | ID of decision table |

**management_sch_op**

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| management_sch_id | int | ID of management schedule |
| op_typ | text | Type of operation (see options below) |
| mon | int | Month operation takes place |
| day | int | Day operation takes place |
| op_data1 | text | Dependent on op_typ (see options below) |
| op_data2 | text | |
| op_data3 | real | Override value |

**Operations Types**

| Code | Description |
|---|---|
| plnt | plant |
| harv | harvest only |
| kill | kill |
| hvkl | harvest and kill |
| till | tillage |
| irrm | irrigation |
| fert | fertilizer |
| pest | pesticide application |
| graz | grazing |
| burn | burn |
| swep | street sweep |
| prtp | print plant vars |
| skip | skip to end of the year |

**Operation Data 1 Values**

| Code | Value | Look-up Table |
|---|---|---|
| plnt | plant name | plants_plt |
| harv | plant name | plants_plt |
| kill | plant name | plants_plt |
| hvkl | plant name | plants_plt |
| till | tillage name | tillage_til |
| irrm | irrigation operation name | irr_ops |
| fert | fertilizer name | fertilizer_frt |
| pest | pesticide name | pesticide_pst |
| graz | graze operation name | graze_ops |
| burn | fire operation name | fire_ops |
| swep | street sweep operation name | sweep_ops |
| prtp | none | |
| skip | none | |

**Operation Data 2 Values**

| Code | Value | Look-up Table |
|---|---|---|
| plnt | none | |
| harv | harvest operation name | harv_ops |
| kill | none | |
| hvkl | harvest operation name | harv_ops |
| till | none | |
| irrm | none | |
| fert | chemical application operation name | chem_app_ops |
| pest | chemical application operation name | chem_app_ops |

| | |
|---|---|
| graz | none |
| burn | none |
| swep | none |
| prtp | none |
| skip | none |

# Operations Databases

Values in the operations tables are provided in the SWAT+ datasets database and copied to your project database during project setup. You may modify them or add new rows as needed in the editor.

## Harvest

| SWAT+ Input File | Database Table |
|---|---|
| harv.ops | harv_ops |

| Field | Type | Description | Units |
|---|---|---|---|
| id | int | Auto-assigned identifier | |
| name | text | Name of operation | |
| harv_typ | text | Harvest type: grain, biomass, residue, tree, or tuber | |
| harv_idx | real | Harvest index target specified at harvest | |
| harv_eff | real | Harvest efficiency | |
| harv_bm_min | real | Minimum biomass to allow harvest | kg/ha |
| description | text | Optional description | |

## Graze

| SWAT+ Input File | Database Table |
|---|---|
| graze.ops | graze_ops |

| Field | Type | Description | Units | Range |
|---|---|---|---|---|
| id | int | Auto-assigned identifier | | |
| name | text | Name of operation | | |
| fert_id | int | ID of fertilizer from fertilizer_frt | | |

| | | | | |
|---|---|---|---|---|
| bm_eat | real | Dry weight of biomass removed by grazing daily | kg/ha | 0-500 |
| bm_tramp | real | Dry weight of biomass removed by trampling daily | kg/ha | 0-500 |
| man_amt | real | Dry weight of manure deposited | kg/ha | 0-500 |
| grz_bm_min | real | Minimum plant biomass for grazing to occur | kg/ha | 0-5000 |
| description | text | Optional description | | |

## Irrigation

| SWAT+ Input File | Database Table |
|---|---|
| irr.ops | irr_ops |

| Field | Type | Description | Units | Range |
|---|---|---|---|---|
| id | int | Auto-assigned identifier | | |
| name | text | Name of operation | | |
| irr_eff | real | Irrigation efficiency | | 0-1 |
| surq_rto | real | Surface runoff ratio | | 0-1 |
| irr_amt | real | Depth of application for subsurface | mm | 0-100 |
| irr_salt | real | Concentration of salt in irrigation water | mg/l | |
| irr_no3n | real | Concentration of nitrate in irrigation water | mg/l | |
| irr_po4n | real | Concentration of phosphate in irrigation water | mg/l | |
| description | text | Optional description | | |

## Chemical Application

| SWAT+ Input File | Database Table |
|---|---|
| chem_app.ops | chem_app_ops |

| Field | Type | Description | Units |
|---|---|---|---|

| | | | |
|---|---|---|---|
| id | int | Auto-assigned identifier | |
| name | text | Name of operation | |
| chem_form | text | Chemical form: liquid or solid | |
| app_typ | text | Application type: spread, spray, inject, direct | |
| app_eff | real | Application efficiency | |
| foliar_eff | real | Foliar efficiency | |
| inject_dp | real | Injection depth | mm |
| surf_frac | real | Surface fraction amount in upper 10mm | |
| drift_pot | real | Drift potential | |
| aerial_unif | real | Aerial uniformity | |
| description | text | Optional description | |

## Fire

| SWAT+ Input File | Database Table |
|---|---|
| fire.ops | fire_ops |

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| name | text | Name of operation |
| chg_cn2 | real | Change in SCS curve number II value |
| frac_burn | real | Fraction burned |
| description | text | Optional description |

## Sweep

| SWAT+ Input File | Database Table |
|---|---|
| sweep.ops | sweep_ops |

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| name | text | Name of operation |
| swp_eff | real | Removal efficiency of sweeping operation |
| frac_curb | real | Fraction of the curb length that is sweep-able |
| description | text | Optional description |

# Curve Number Table

Values in this table are provided in the SWAT+ datasets database and copied to your project database during project setup. You may modify them or add new rows as needed in the editor.

| SWAT+ Input File | Database Table |
|---|---|
| cntable.lum | cntable_lum |

| Field | Type | Description | Range |
|---|---|---|---|
| id | int | Auto-assigned identifier | |
| name | text | Name of curve number entry | |
| cn_a | real | Curve number for hydrologic soil group A | 30-100 |
| cn_b | real | Curve number for hydrologic soil group B | 30-100 |
| cn_c | real | Curve number for hydrologic soil group C | 30-100 |
| cn_d | real | Curve number for hydrologic soil group D | 30-100 |
| description | text | Optional description | |
| treat | text | Treatment/Practice | |
| cond_cov | text | Condition of cover | |

## Conservation Practices

Values in this table are provided in the SWAT+ datasets database and copied to your project database during project setup. You may modify them or add new rows as needed in the editor.

| SWAT+ Input File | Database Table |
|---|---|
| cons_practice.lum | cons_practice_lum |

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| name | text | Name of curve number entry |
| usle_p | real | Usle p factor |
| slp_len_max | real | Maximum slope length |
| description | text | Optional description |

# Overland Flow Manning's n

Values in this table are provided in the SWAT+ datasets database and copied to your project database during project setup. You may modify them or add new rows as needed in the editor.

| SWAT+ Input File | Database Table |
|---|---|
| ovn_table.lum | ovn_table_lum |

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| name | text | Name of curve number entry |
| ovn_mean | real | Overland flow Manning's n = mean |
| ovn_min | real | Overland flow Manning's n = min |
| ovn_max | real | Overland flow Manning's n = max |
| description | text | Optional description |

# Decision Tables

Decision tables are a precise yet compact way to model complex rule sets and their corresponding actions. Decision tables, like flowcharts and if-then-else and switch-case statements, associate conditions with actions to perform, but in many cases do so in a more elegant way (see Wikipedia article on decision tables).

Structure of decision tables:

1. Conditions
2. Condition alternatives
3. Actions
4. Action entries

Each decision corresponds to a variable, relation or predicate whose possible values are listed among the condition alternatives. Each action is a procedure or operation to perform, and the entries specify whether (or in what order) the action is to be performed for the set of condition alternatives the entry corresponds to. Many decision tables include in their condition alternatives the "don't care" symbol, a hyphen. Using "don't cares" can simplify decision tables, especially when a given condition has little influence on the actions to be performed. In some cases, entire conditions thought to be important initially are found to be irrelevant when none of the conditions influence which actions are performed.

## Usage in SWAT+

There are four decision table sections in SWAT+: land use management, reservoir release, scenario land use, and flow conditions.

See the land use management documentation, under management schedules for how to choose a decision table for your HRUs.

→ Land Use Management /user/editor/inputs/land-use-management

Reservoir release decision tables are assigned from the reservoir properties section under connections in SWAT+ Editor.

# Modifying Decision Tables in SWAT+ Editor

In the current version of SWAT+ editor, we do not have a GUI available for editing decision tables, however we have provided the ability to export the decision table file. You may then make changes to it in a text editor, and upload back into SWAT+ Editor.

# Table Definitions

| SWAT+ Input File | Database Table |
|---|---|
| lum.dtl, res_rel.dtl, scen_lu.dtl, flo_con.dtl | d_table_dtl |
| | d_table_dtl_cond |
| | d_table_dtl_cond_alt |
| | d_table_dtl_act |
| | d_table_dtl_act_out |

## d_table_dtl

| Field | Type | Description |
|---|---|---|
| id | int | Auto-assigned identifier |
| name | text | Name of the decision table |
| file_name | text | File name denoting type of decision table:  lum.dtl, res_rel.dtl, scen_lu.dtl, flo_con.dtl |

## d_table_dtl_cond

| Field | Type | Description | Related Table |
|---|---|---|---|
| id | int | Auto-assigned identifier | |
| d_table_id | int | ID of decision table | d_table_dtl |
| var | text | Condition variable | |
| obj | text | Object variable (res, hru, etc) | |
| obj_num | int | Object number | |
| lim_var | text | Limit variable (evol, pvol, fc, etc) | |
| lim_op | text | Limit operator (*, +, -) | |
| lim_const | real | Limit constant | |

## d_table_dtl_cond_alt

| Field | Type | Description | Related Table |
|---|---|---|---|
| id | int | Auto-assigned identifier | |
| cond_id | int | ID of condition | d_table_dtl_cond |
| alt | text | Condition alternatives (>, <, =) | |

## d_table_dtl_act

| Field | Type | Description | Related Table |
|---|---|---|---|
| id | int | Auto-assigned identifier | |
| d_table_id | int | ID of decision table | d_table_dtl |
| act_typ | text | Type of action (reservoir, irrigate, etc) | |
| obj | text | Object variable (res, hru, etc) | |
| obj_num | int | Object number | |
| name | text | Name of action | |
| option | text | Action option-specific to type of action (e.g., for reservoir, option to input rate, days of draw-down, weir equation pointer, etc) | |
| const | real | Constant used for rate, days, etc | |
| const2 | real | | |
| fp | text | Pointer for option (e.g., weir equation pointer) | |

## d_table_dtl_act_out

| Field | Type | Description | Related Table |
|---|---|---|---|
| id | int | Auto-assigned identifier | |
| act_id | int | ID of action | d_table_dtl_act |
| outcome | bool | Perform action (1 or true), or don't perform action (0 or false) | |

# Text File Example and Explanation

The following is an example of a decision table in the lum.dtl input file. It is a table for warm season annual crops, using continuous corn.

| name | conds | alts | acts | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| pl_hv_corn | 6 | 4 | 3 | | | | | | | | |
| var | obj | obj_num | lim_var | lim_op | lim_const | alt1 | alt2 | alt3 | alt4 | | |
| soil_water | hru | 0 | fc | * | 1.05000 | < | < | - | - | | |
| phu_base0 | hru | 0 | null | - | 0.15000 | > | - | - | - | | |
| phu_plant | hru | 0 | phu_mat | - | 1.15000 | - | > | - | - | | |
| year_rot | hru | 0 | null | - | 1.00000 | = | = | = | - | | |
| jday | hru | 0 | null | - | 350.00000 | - | - | = | - | | |
| year_rot | hru | 0 | null | - | 1.00000 | - | - | - | > | | |
| act_typ | obj | obj_num | name | option | const | const2 | | fp | outcome | | |
| plant | hru | 0 | plant_corn | corn | 0.00000 | 1.00000 | | null | y | n | n | n |
| harvest_kill | hru | 0 | grain_harv | corn | 0.00000 | 1.00000 | | grain | n | y | y | n |
| rot_reset | hru | 0 | reset_1 | null | 1.00000 | 0.00000 | | null | n | n | n | y |

In the above table, there are 6 conditions, 4 alternatives and 3 actions.

## Description of the conditions

1. soil_water – if soil water is too high (> 1.50*field capacity), it will be too wet to operate machinery
2. plant_gro – (“n”) Planting allowed if plant is not growing.
3. phu_base0 – (0.15) when the sum base zero heat units for the year (starting Jan 1) exceeds 0.15, indicating it’s warm enough to plant
4. phu_plant – (1.15) harvest is scheduled when the sum of the heat units for the crop exceed 1.15 (if the heat units to maturity for the crop = 1500, then harvest would be scheduled at 1.15 * 1500 = 1725).
5. year_rot – needed to identify the current year of rotation.  In this example, corn is grown in year 1.
6. days_plant – days since last plant (200) to ensure harvest occurs before next crop is planted.

## Description of the alternatives

If all of the conditions for each alternative are met, outcomes are checked for ‘y’ to take action. Alternatives with dash (‘-‘) are not checked.

1. plant corn based on heat units: if soil water < 1.50*fc and  if phubase0 > 0.15*phu_mat and if year_rot = 1  then check outcomes for ‘y’ and if ‘y’, take that action (plant)
2. Harvest corn based on crop accumulated heat units: if soil_water < 1.50*fc and if phu_plant > 1.15*phu_mat and if year_rot = 1 and then check outcomes for ‘y’ and if ‘y’, take that action (plant)
3. Harvest corn based on days since planting: if year_rot = 1 and if days_plant =200 then check outcomes for ‘y’ and if ‘y’, take that action (harvest)
4. Reset rotation year: if year_rot > 1 then check outcomes for ‘y’ and if ‘y’, take that action (rot_reset)

## Description of the actions

1. plant: corn – cross walked to plant name in plants.plt file
2. harvest_kill:  corn – cross walked to plant name in plants.plt file grain – relates to harvest type in harv.ops file
3. rot_reset: rotation reset – for continuous corn (1 year rotation). The rotation year is reset to 1 at the end of every year.

# Change/Calibration

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

SWAT+ IO Documentation /user/io

# Initialization Data

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

SWAT+ IO Documentation /user/io

# Hydrology

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

SWAT+ IO Documentation /user/io

# Soils

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

SWAT+ IO Documentation /user/io

# Databases

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

SWAT+ IO Documentation /user/io

# Structural

Documentation for this section is not available yet. For now, please refer to the SWAT+ input/output documentation PDF for parameter definitions.

SWAT+ IO Documentation /user/io

# Write Input Files

When you are done editing your input data, click the pencil icon in the leftmost blue toolbar to go to the write input files section. Select the directory to save your files (if coming from GIS, your default scenario TxtInOut directory is chosen by default), and click the save and write files button.

A progress bar will pop up showing the files being written. When complete, you can proceed to the next step: running the model. Click the arrow button in the leftmost blue toolbar.

> You must have weather stations before being allowed to write input files.

If you need to make additional changes to your SWAT+ input data, be sure to come back to this step and re-write your files before running the model again.

# Run SWAT+

After writing your input files, click the arrow button in the leftmost blue toolbar to go to the run SWAT+ section. Click the button to run the model. A progress window will pop up displaying the results from the model executable program.

If you encounter an error, check the box to run the debug version and run the model again. Copy the contents of the output error and see our SWAT+ model user group for help diagnosing the problem.

# Analyze Output

After successfully running the model, click the graph icon in the leftmost blue toolbar, then click the import output button to read your SWAT+ output files into a SQLite database. When you are done, close the SWAT+ Editor window to return to QSWAT+. From here, the step 4 button should be enabled in QSWAT+ and you can plot or map your output.

To change the output printed in your model run, go to the edit inputs section of SWAT+ Editor, and click print under the simulation section. Choose the data you want to print, save changes, then re-write input files and re-run the model.

See the simulation section documentation linked below for help with print options.

→ Simulation /user/editor/inputs/simulation

# QSWAT+ Manual

QSWAT+ 1.2.2 Manual

QSWATPlus Manual_v1.2.2.pdf - 4MB

# SWAT+ IO Documentation

The SWAT+ input documentation is available as a PDF file here. However, you may also follow the SWAT+ Editor documentation through this website, as it will describe the SWAT+ input files and fields.

**SWAT+ Revision 59.2/3 Input Documentation** inputs_swatplus_rev59_2.pdf - 2MB

**SWAT+ Editor Documentation** /user/editor

SWAT+ output documentation is not yet available. However, units are available under the heading line in the output files, and also in the `column_description` table in the output SQLite database.

# How to Use SQLite

QSWAT+ and SWAT+ Editor use a SQLite database to hold model input data to allow easy manipulation by the user. The database is structured to closely resemble the SWAT+ input text files in order to keep a clean link between the model and editor.

## Opening the SQLite Database

We recommend using the SWAT+ Editor program provided to browse and edit SWAT+ input data. However, if you need to access the database, we recommend using:

- SQLite Studio

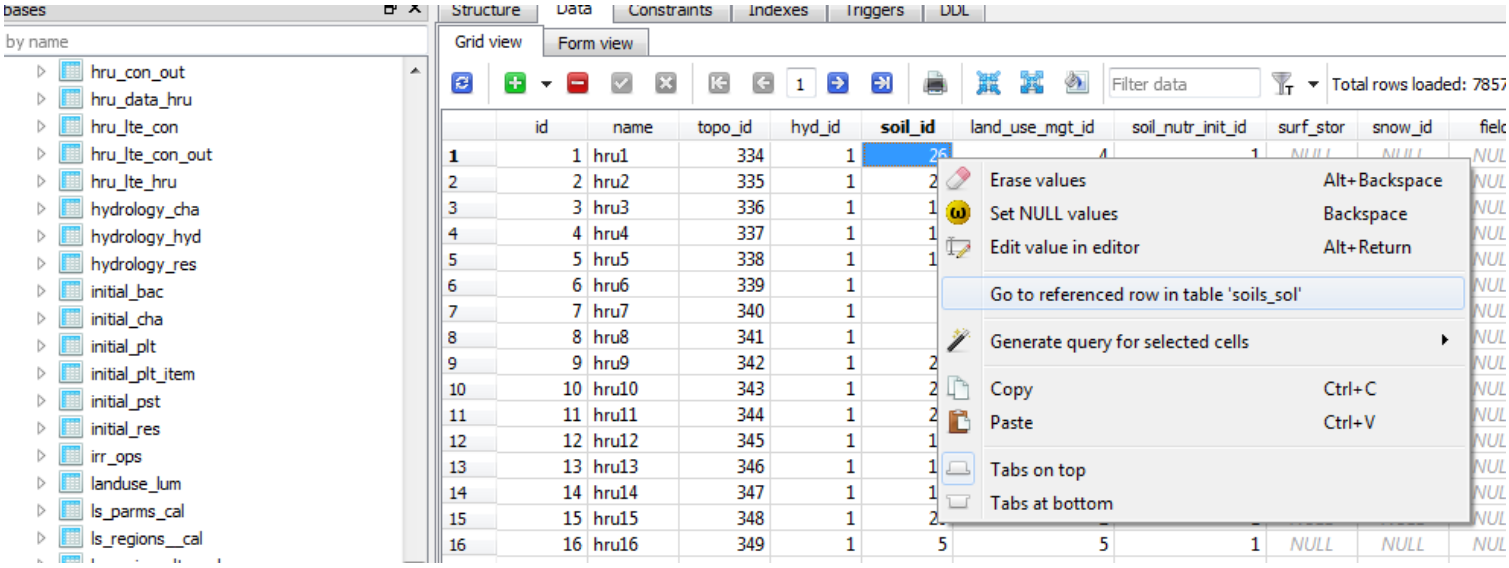
There are many other alternatives out there. A few of them are:

- DB Browser for SQLite
- SQLite Manager, Firefox Add-On

# Understanding Table Relationships

SWAT+ contains many cross-walks between files, and the database follows suit by creating foreign key relationships where applicable. In the SWAT+ text files, you will see files reference object names from another file. In the database however, these are done with an integer id. Relational databases make it easy to view the referenced row.

In SQLite Studio, right-click a foreign key id in a row of data, and select “Go to referenced row in table …” as shown in the image below. This will open the referenced row of data in a new tab.




Results in:

| | id | name | hydgrp | zmx | anion_excl | crk | texture | desc |
|---|---|---|---|---|---|---|---|---|
| 1 | 26 | Podzols | C | 570 | 0.5 | 0.5 | SANDY_CLAY_LOAM | NULL |

# SWAT+ Editor Design

SWAT+ Editor is a program that allows users to modify SWAT+ inputs easily without having to touch the SWAT+ input text files directly. The editor will import a watershed created in QSWAT+, or allow the user to create a SWAT+ project from scratch. The user may write input files and run the SWAT+ model through the editor.

# Technologies

The following software is used to create and build SWAT+ Editor:

- Node.js
- Electron
- Vue.js 2.x
- Bootstrap 4
- Python 3.x
- PyInstaller
- SQLite
- Peewee ORM

# Database Design

SWAT+ Editor uses a SQLite database to hold model input data to allow easy manipulation by the user. The database is structured to closely resemble the SWAT+ ASCII text files in order to keep a clean link between the model and editor. The following conventions are used in the project database:

- The table names will match the text file names, replacing any "." or "-" with an underscore "_".
- The table column names will match the model's variable names. All names use lowercase and underscores.
- Any text file with a variable number of repetitive columns will use a related table in the database. For example, many of the connection files contain a variable number of repeated outflow connection columns (obtyp_out, obtyno_out, hytyp_out, frac_out). In the database, we represent these in a separate table, basically transposing a potentially long horizontal file to columns.
- All tables will use a numeric "id" as the primary key, and foreign key relationships will use these integer ids instead of a text name. This will allow for easier modification of these object names by the user and help keep the database size down for large projects.

A separate SQLite database containing common datasets and input metadata will be provided with SWAT+ Editor. (This is a replacement for the SWAT2012.mdb packaged with SWAT2012 versions of ArcSWAT, QSWAT, and SWATeditor.)

In addition, reformatted SSURGO and STATSGO soils databases is available for download. The structure of the soils database has been split into two tables: a soil table and soil_layer table.

Similarly, the global weather weather generator database is available for download in SQLite format. The structure of the wgn database has been split into two tables: a wgn table and wgn_monthly_value table.

## Database Access in the Python API

SWAT+ Editor uses the Peewee ORM (object-relational mapping) to represent and work with the tables in Python. The use of an ORM provides a layer of abstraction and portability in hopes of streamlining future SWAT+ development projects.

Relationships are defined in a Peewee ORM python class as a `ForeignKeyField`. In the python class, the field will be named after the object it is referencing. In the database, this name will automatically be appended by the referencing table's column name, which is usually `id`.

For example, we have two tables representing soils: soils ( `soils_sol` ) and layers ( `soils_sol_layer` ). The layer table has a foreign key to the main soils table, so we know to which soil the layer belongs. In the

python class, this field is named `soil`, and in the database it is called `soil_id`.

# Source Code

See the link below for information about accessing and running the source code.

→ Source Code /source-code