2024

# CyberWater Manual

## BETA Release

By Ranran Chen[1], Drew A. Bieger[1], Daniel Luna[2], Sherif Aly[2], Ryan Young[2], Yao Liang[1], and Xu Liang[2]

[1] Indiana University-Purdue University Indianapolis

[2] University of Pittsburgh

# TABLE OF CONTENTS

# CYBERWATER MANUAL (BETA RELEASE)

By Ranran Chen[1], Drew A. Bieger[1], Daniel Luna[2], Sherif Aly[2], Ryan Young[2], Yao Liang[1], and Xu Liang[2]

This manual is designed to guide users through the installation and uses of *CyberWater* through a number of exercises. For questions, please contact Dr. Xu Liang (xuliang@pitt.edu) or Dr. Yao Liang (yaoliang@iupui.edu) of this project principal investigators.

*CyberWater* [1] is an open and sustainable modeling framework software system that enables easy and incremental integration of diverse data sources and models for knowledge discovery and interdisciplinary teamwork, reproducible computing, and seamless, on-demand access to various HPC resources. CyberWater utilizes the visual interface of *VisTrails*, a scientific workflow management system that offers a friendly and convenient graphical workflow interface in addition to its unique capabilities such as provenance and data visualization. It allows executing complex workflows with comprehensive interactions between remotely accessible heterogeneous data sources and diverse models and model couplings. CyberWater offers a cyberinfrastructure that enables seamlessly automated machine-to-machine access from external heterogeneous data sources to diverse models. It also allows users to easily add different data sources and models into the CyberWater system through its Open Data and Open Modeling architecture, via Data Agents, Model Agents or generic model agent tools, and Workflow, all in one place. The CyberWater project is supported by NSF awards. This CyberWater user manual focuses on how to use/operate CyberWater modeling software through use-case examples and also in a step-by-step fashion via examples provided in it. For users who want to better understand the underlying ideas and principles of the CyberWater system, please refer to reference [1] and other references listed below. If you make any use of this CyberWater software system, please acknowledge the appropriate references listed in the references below. These should include Salas et al. (2020), Chen et al. (2022a), and any references relevant to the features you are using as illustrated in this manual and/or in the CyberWater website under the Overview page.

**REFERENCES**:

(1) Salas, D., X. Liang, M. Navarro, Y. Liang, and D. Luna, 2020. An open-data open-model framework for hydrological models' integration, evaluation and application. *Environmental Modelling & Software,* Volume 126, 104622, 1-20, https://doi.org/10.1016/j.envsoft.2020.104622, 2020.

(2) Li, F., R. Chen, T. Fu, F. Song, Y. Liang, I. Ranawaka, S. Pamidighantam, D. Luna, and X. Liang, 2021. Accelerating complex modeling workflows in CyberWater using on-demand HPC/Cloud resources. The

[1] Indiana University-Purdue University Indianapolis
[2] University of Pittsburgh

17th IEEE eScience Conference, 196–205, DOI:https://doi.org/10.1109/eScience51609.2021.00030, 2021.

(3) Chen, R., D. Luna, Y. Cao, Y. Liang, and X. Liang, 2022a. Open data and model integration through generic model agent toolkit in CyberWater framework, Environmental Modelling and Software, 152, 105384, https://doi.org/10.1016/j.envsoft.2022.105384, 1-16, 2022a.

(4) Chen, R., D. Luna, F. Li, R. Young, D. Bieger, F. Song, S. Pamidighantam, Y. Liang and X. Liang. 2022b. CyberWater: An Open Framework for Data and Model Integration in Water Science and Engineering. In Proceedings of the 31st ACM Int'l Conference on Information and Knowledge Management (CIKM'22), Oct. 17-21, 2022, Atlanta, GA. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557186

(5) Luna, D., R. Chen, R. Young, Y. Liang, and X. Liang, Towards flexible, rich, and easy model coupling environments via CyberWater's open data and open model framework, American Geophysical Union Fall Meeting (Hybrid), Chicago, IL, Dec. 12-16, 2022. Website at: https://agu2022fallmeeting-agu.ipostersessions.com/default.aspx?s=DF-BE-CC-07-14-C5-1F-01-F2-24-10-BB-4A-64-04-8C.

(6) Chen, R., D. Luna, A. Castronova, X. Liang, and Y. Liang, Composable National Water Model simulations using the CyberWater framework, American Geophysical Union Fall Meeting (Hybrid), Chicago, IL, Dec. 12-16, 2022.

**SYSTEM REQUIREMENTS:**

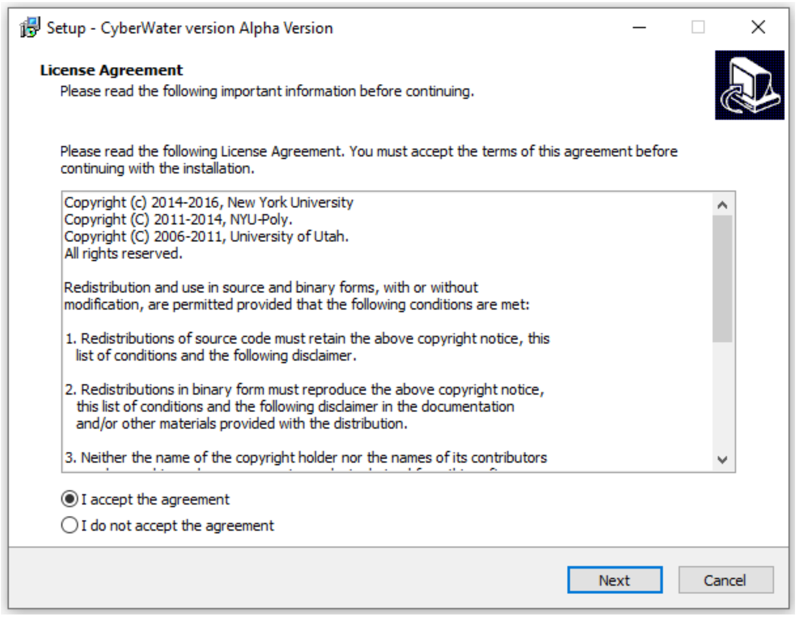
- OS: 64-bit Windows 10 system.
- RAM: Minimum of 6GB. 8 GB desirable.
- Processor: Minimum of 2.1 GHz per core (4 or more are desirable).

# 1. Installation

Installing CyberWater is done through the official *CyberWater* installer. The installer will also lead the user through installing *VisTrails*, *GRASS GIS, Java, and Docker for WRF-Hydro*. The entire installation takes about 12 minutes. You can get the CyberWater installer from the download webpage of CyberWater website https://cyber-water.cs.iupui.edu/ after a quick registration.

## 1.1 CyberWater installer

When the user opens the installer, they may be prompted by their antimalware to confirm they wish to proceed with the execution of the program service (by default "Windows SmartScreen", go to More Info > Run Anyway). The user must approve the program to begin the installation. The first step of the installer is shown below in Figure 1. The user must accept the license agreement to proceed.



Setup - CyberWater version Alpha Version

**License Agreement**

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Copyright (c) 2014-2016, New York University
Copyright (C) 2011-2014, NYU-Poly.
Copyright (C) 2006-2011, University of Utah.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

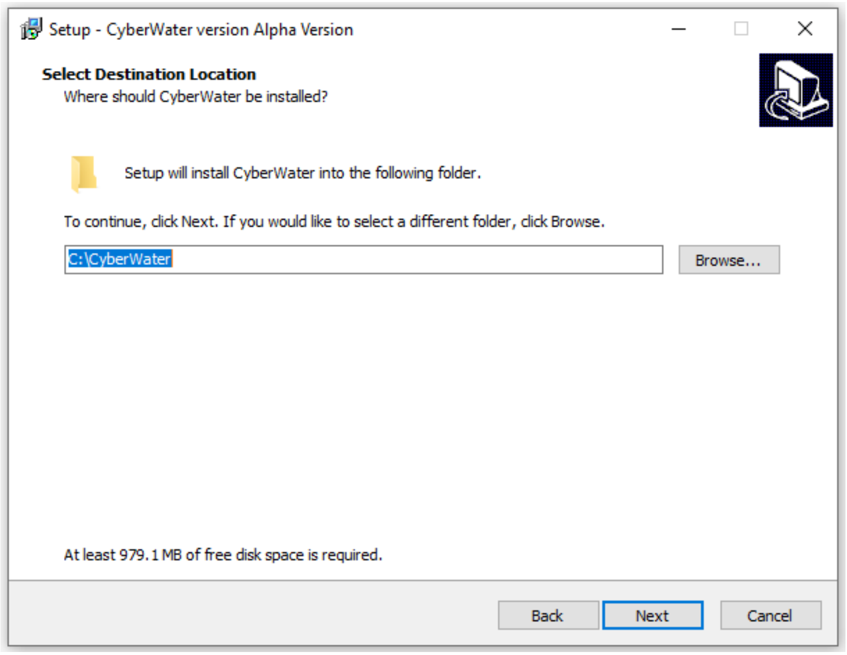
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors

[✓ I accept the agreement] I do not accept the agreement

*Figure 1.The first step of the official CyberWater installer.*

Next, the user must specify the location they would like *CyberWater* to be installed. It is recommended that the user uses the default '*C:\CyberWater*' installation location suggested by the installer. If the user choses to select a different location, they must keep in mind that since *CyberWater* creates cache files for some of its modules when used, it cannot be installed inside operating system folders or any folders that require administrative privileges. This step is shown below in Figure 2.



Setup - CyberWater version Alpha Version

**Select Destination Location**

Where should CyberWater be installed?

Setup will install CyberWater into the following folder.

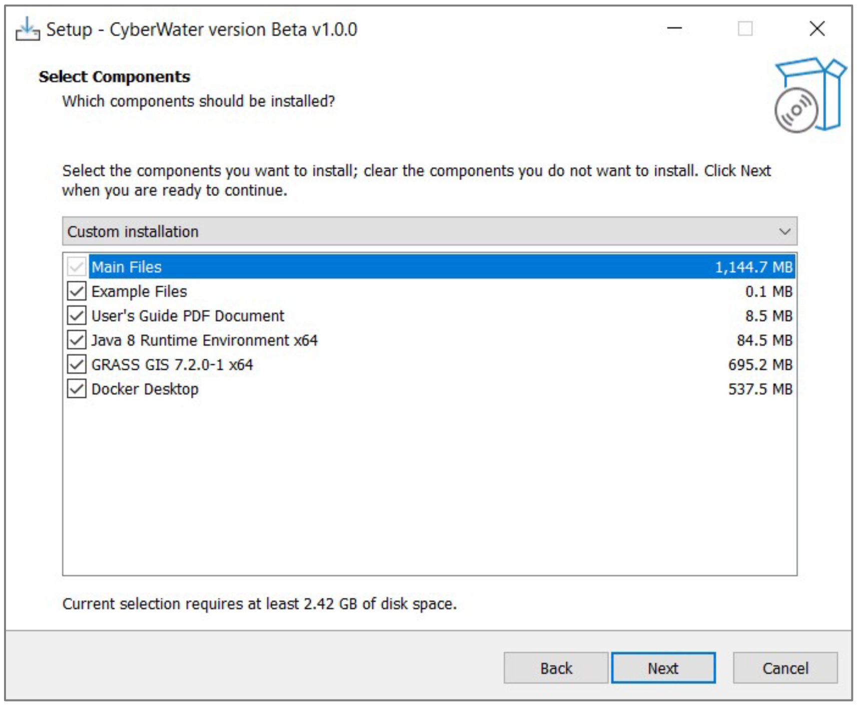
To continue, click Next. If you would like to select a different folder, click Browse.

C:\CyberWater

At least 979.1 MB of free disk space is required.

*Figure 2. The user must specify a location to install CyberWater.*

After selecting the installation destination location, the user is prompted to choose what components they would like to install as shown in Figure 3. It is highly recommended that the user completes a full installation, and if the components listed have been installed before, users can uncheck them. The Docker Desktop is not essential to use CyberWater but it is used to illustrate how some models (e.g., WRF-Hydro) can be integrated into CyberWater via Docker as demonstrated in Section 7. Next, the users are asked if they would like to add a *CyberWater* shortcut to their desktop. This is not required but is recommended. Once the user is happy with their choices, they should begin the installation by clicking the "Install" button.


Setup - CyberWater version Beta v1.0.0

**Select Components**

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

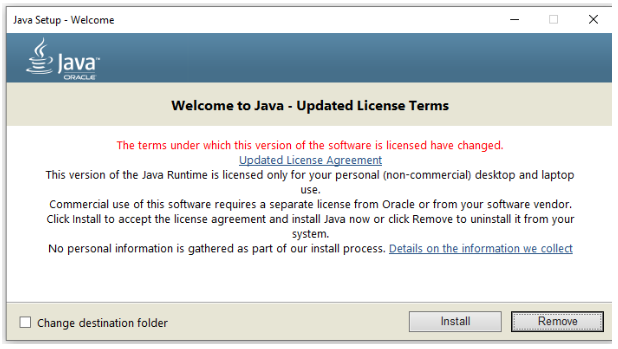
Custom installation

| | |
|---|---|
| ✓ Main Files | 1,144.7 MB |
| ✓ Example Files | 0.1 MB |
| ✓ User's Guide PDF Document | 8.5 MB |
| ✓ Java 8 Runtime Environment x64 | 84.5 MB |
| ✓ GRASS GIS 7.2.0-1 x64 | 695.2 MB |
| ✓ Docker Desktop | 537.5 MB |

Current selection requires at least 2.42 GB of disk space.

*Figure 3. Components to be installed with CyberWater*

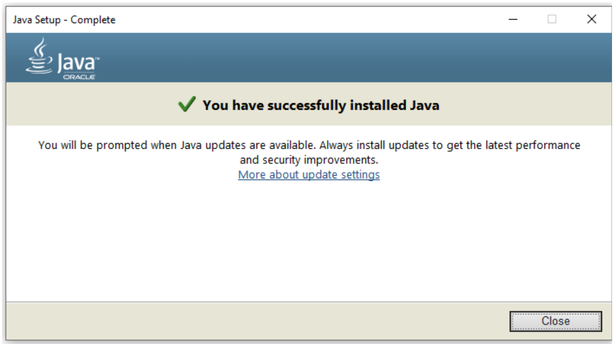
After extracting some files, the installer will begin installing *Java*. This *Java* installation will install alongside any other *Java* installations the user has, so it is highly recommended that the user completes the installation even if they already have a version of *Java* installed. The user must accept the license by clicking "Install" in the Java Setup window to continue as shown in Figure 4. This is also the point in the installation where the user can change where Java is installed by checking "Change destination folder", but that is not recommended. Once the Java Setup finishes installing *Java*, the user will be presented with a window informing them of the success of their *Java* installation. The user must click "close" on this window for the *CyberWater* installation to proceed. This window is shown in Figure 5.


Java Setup - Welcome

Welcome to Java - Updated License Terms

The terms under which this version of the software is licensed have changed.
Updated License Agreement
This version of the Java Runtime is licensed only for your personal (non-commercial) desktop and laptop use.
Commercial use of this software requires a separate license from Oracle or from your software vendor.
Click Install to accept the license agreement and install Java now or click Remove to uninstall it from your system.
No personal information is gathered as part of our install process. Details on the information we collect

*Figure 4. The user must accept the Java license.*


Java Setup - Complete

You have successfully installed Java

You will be prompted when Java updates are available. Always install updates to get the latest performance and security improvements.
More about update settings

*Figure 5. The user must close the final window to continue the installation.*

The installer will continue to extract more files and automatically install Grass GIS 7.2 and Docker Desktop if they are selected from the installation package as shown in Figure 3. After finishing the installation process, the installer may ask for a system reboot depending on the installed components chosen by the user. It is recommended to open *GRASS GIS* after finishing installation (or rebooting the system) and confirm seeing '*demolocation'* in panel '2. Select GRASS Location' as shown in Figure 6. If the user is missing this, they should follow the directions described in Section 1.2.2. Finally, to complete the Docker Desktop installation if it is selected from the installation package, follow the instructions in Section 1.2.3.

Many of the following examples will assume files located in "*C:\temp\CYBERWATER*". Please create the two folders if they are not already found on your local machine. It is recommended that the user downloads the available data required to test the manual examples from the provided source to this directory and that the user works within the directory for the examples, although this is not required.

## 1.2 Additional Installation Instructions

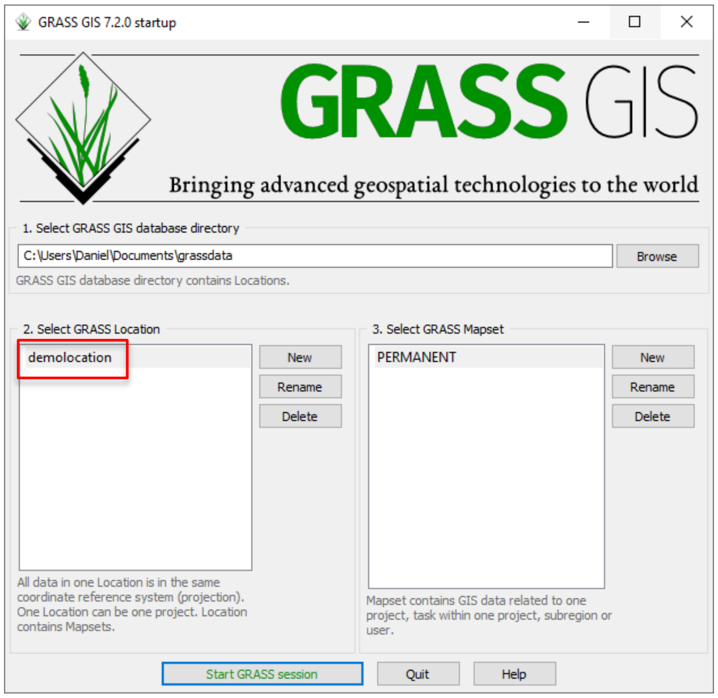
### 1.2.1 *Learn more about CyberWater*

The user is welcomed to watch or review the videos related to *CyberWater* features and its four short demos from the CyberWater project page in the section "Learn More About CyberWater" under CyberWater at https://cyber-water.cs.iupui.edu/lectures/. The five videos are:

(1) Overview of the CyberWater Project, Platform and Features (~30 minutes)
(2) The CyberWater Interface and Modeling Workflow (~14 minutes)
(3) Using the Generic Model Agent Toolkit to Port Models into CyberWater (~14 minutes)
(4) Leveraging High Performance Computing in CyberWater (~14 minutes)
(5) CyberWater's Geographic Information System Capabilities (~10 minutes)

### 1.2.2 *Setting up GRASS GIS*

After the installation, it is important to check where the "grassdata" folder is installed in your computer. This is because sometimes, the "grassdata" folder related to the GRASS GIS is NOT automatically put in the "documents" folder under one's "C:/" drive, rather, it is put in the "documents" folder of a different drive due to the setup of individual's computer. In such a case, the user needs to copy the entire "grassdata" folder and put it at: "C:/[username]/Documents/". After this action, it should look like: "C:/[username]/Documents/grassdata". After this check, the user should open GRASS GIS to check if they see "demolocation" in panel 2. If yes, the user could escape the remaining part of this section and move to the next section. If not, please follow through the remaining description here. In the "grassdata" folder, it should have the "demolocation" folder in which it has "PERMANENT" folder and a file called ".grassrc72" whose file type is: GRASSRC72 File". If one does not see “demolocation” in panel “2. Select GRASS Location”, as shown in Figure 6, then , one needs to go through the following steps. Otherwise, please proceed to Section 1.2.3.



C:\Users\Daniel\Documents\grassdata

GRASS GIS database directory contains Locations.

demolocation

PERMANENT

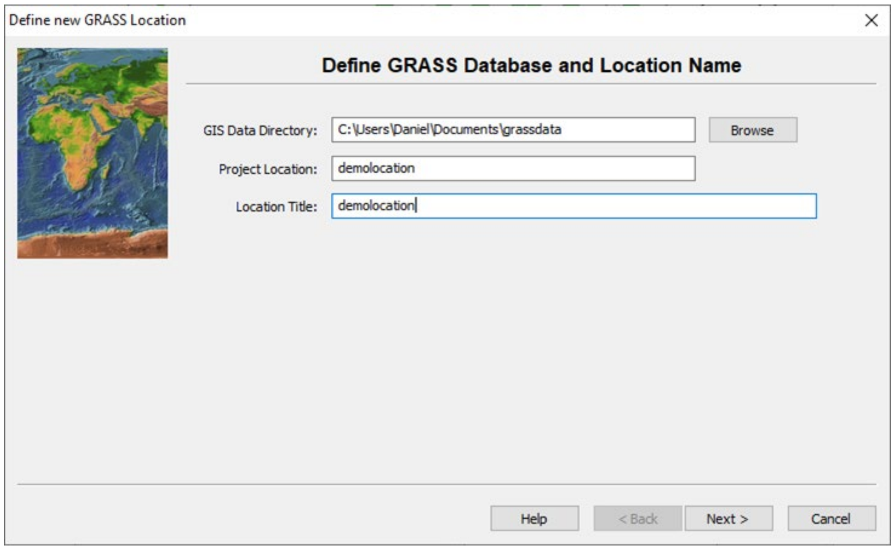
All data in one Location is in the same coordinate reference system (projection). One Location can be one project. Location contains Mapsets.

Mapset contains GIS data related to one project, task within one project, subregion or user.

*Figure 6. Check if the word “demolocation” is available when opening Grass GIS for the first time*

If such a situation happens, create a folder named ‘*grassdata’* inside your Documents [1] folder. Now click on the ‘**Browse’** button in the menu above and select the newly created ‘*grassdata’* folder. Then, click on the ‘**New’** button of the left panel ‘2. Select GRASS Location’. Then rename the ‘**Project location**’ and ‘**Loaction Title**’ to “*demolocation*” as shown in Figure 7.

[1] Path should be*: [Disc]:/Users/[your-username]/Documents/grassdata*


Define new GRASS Location

Define GRASS Database and Location Name

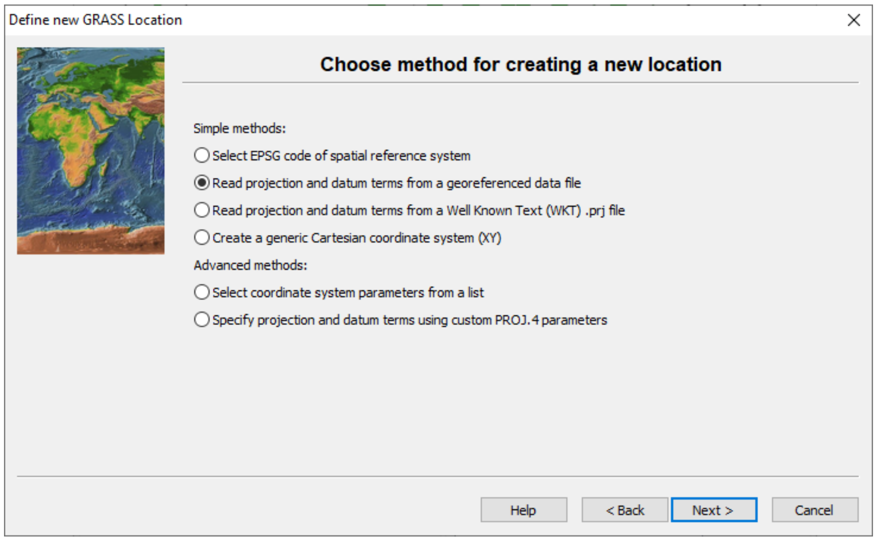
GIS Data Directory: C:\Users\Daniel\Documents\grassdata

Project Location: demolocation

Location Title: demolocation

*Figure 7. Rename the Project Location and Location Title to "demolocation" in the screen page of defining new Grass GIS database*

Click "Next" then select the second option "Read projection and datum terms from a georeferenced data file" as shown in Figure 8, then click "Next".


Define new GRASS Location

Choose method for creating a new location

Simple methods:

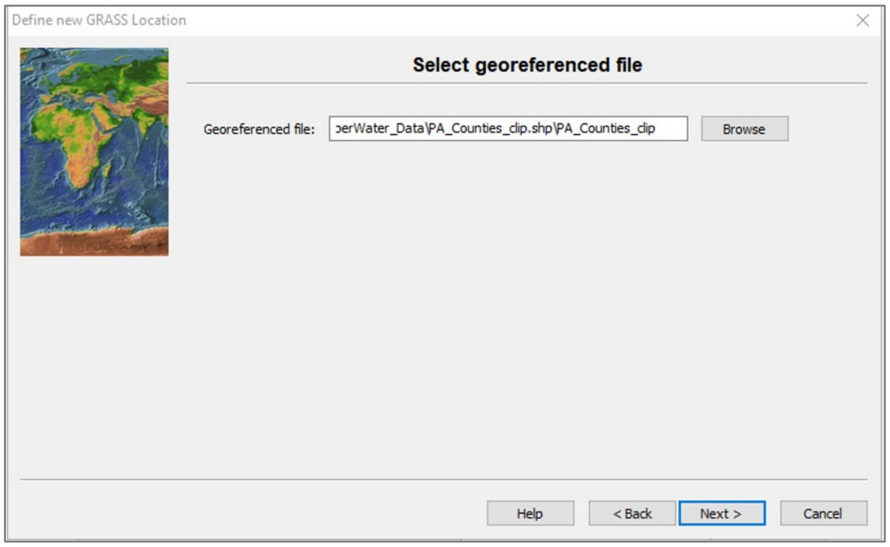
- ( ) Select EPSG code of spatial reference system
- (●) Read projection and datum terms from a georeferenced data file
- ( ) Read projection and datum terms from a Well Known Text (WKT) .prj file
- ( ) Create a generic Cartesian coordinate system (XY)

Advanced methods:

- ( ) Select coordinate system parameters from a list
- ( ) Specify projection and datum terms using custom PROJ.4 parameters

*Figure 8. Choose method for creating a new location in Grass GIS*

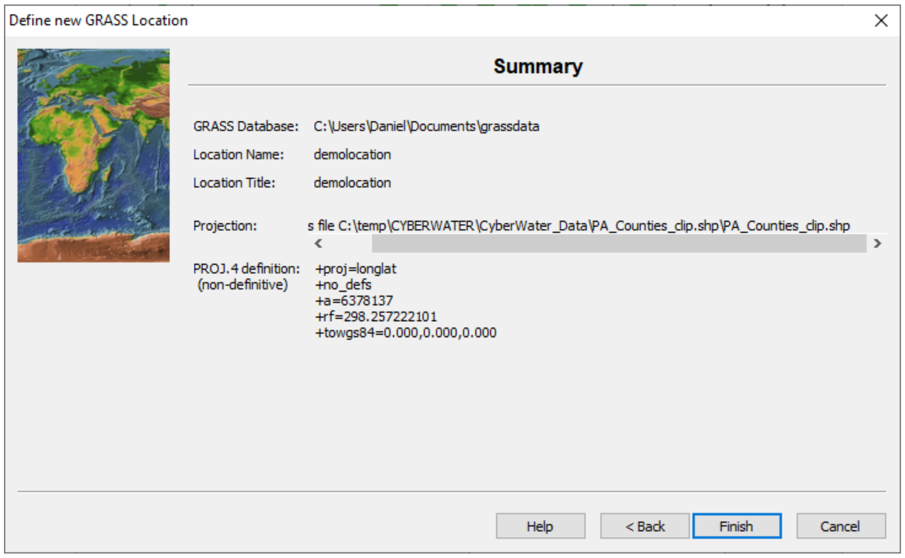
Download the zip file <Examples Data/PA_Counties_clip.shp/PA_Counties_clip.shp.zip> available on HydroShare CyberWater shared folder, or user can download the zip file by clicking and following the hypertext link directly. Make sure to download the whole compressed file and unzip it into the local machine, so all the metadata is loaded appropriately. Browse to the shape source file "PA_Counties_clip" as shown in Figure 9 (file size is around 2762 KB).



Georeferenced file: ɔerWater_Data\PA_Counties_clip.shp\PA_Counties_clip

*Figure 9. Select georeferenced file in Grass GIS*

Click ”Next” and a last display should show a summary as shown in Figure 10. Click “Finish” on the summary display page.

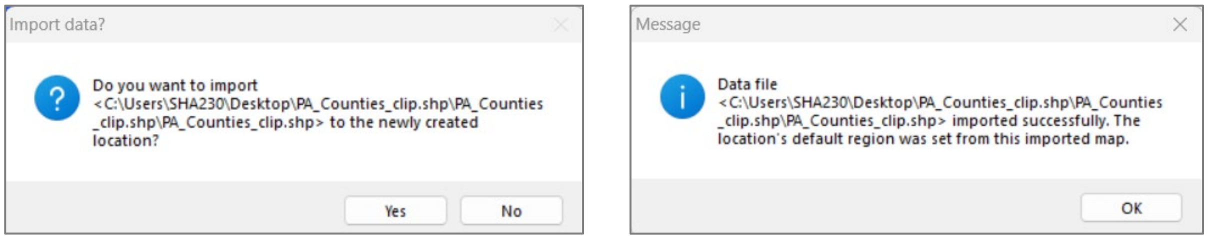


GRASS Database: C:\Users\Daniel\Documents\grassdata
Location Name: demolocation
Location Title: demolocation
Projection: s file C:\temp\CYBERWATER\CyberWater_Data\PA_Counties_clip.shp\PA_Counties_clip.shp
PROJ.4 definition: (non-definitive)

```
+proj=longlat
+no_defs
+a=6378137
+rf=298.257222101
+towgs84=0.000,0.000,0.000
```

*Figure 10. Summary for defining Grass GIS database and location*

Finally, the user will be asked to accept the message shown in Figure 11 (left) to import the data to the newly created location. Then click “OK” to the confirmation message shown in Figure 11 (right).



Import data?

Do you want to import <C:\Users\SHA230\Desktop\PA_Counties_clip.shp\PA_Counties_clip.shp\PA_Counties_clip.shp> to the newly created location?

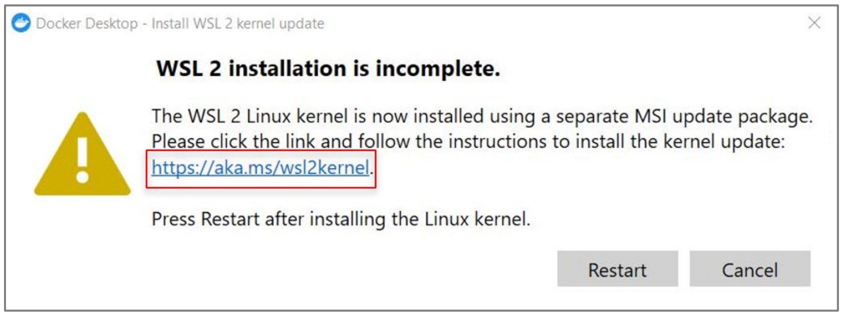
Message

Data file <C:\Users\SHA230\Desktop\PA_Counties_clip.shp\PA_Counties_clip.shp\PA_Counties_clip.shp> imported successfully. The location's default region was set from this imported map.

*Figure 11. Importing data and confirmation message after defining the new database location in Grass GIS*

### 1.2.3 *Setting up Docker Desktop*

This section is only necessary if the user will use the Docker Module in CyberWater discussed in Section 7 in this manual. To complete the Docker Desktop installation, open it. If the message in Figure 12 pops out, then WSL 2 requires an important update. Click on the link in Figure 12 which will direct you to a Microsoft web page.



Docker Desktop - Install WSL 2 kernel update

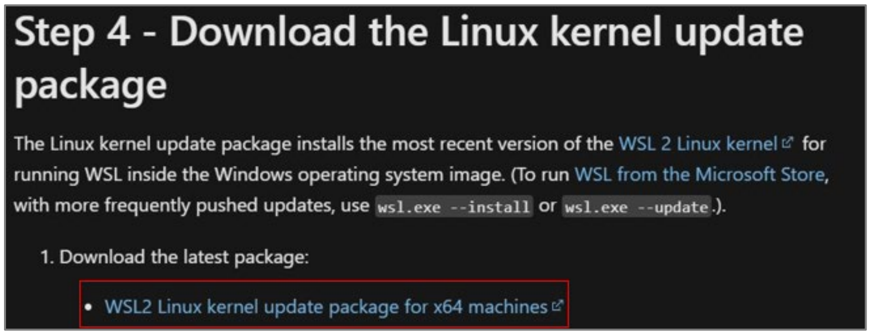
**WSL 2 installation is incomplete.**

The WSL 2 Linux kernel is now installed using a separate MSI update package. Please click the link and follow the instructions to install the kernel update: https://aka.ms/wsl2kernel.

Press Restart after installing the Linux kernel.

*Figure 12. WSL 2 missing updates when opening the Docker Desktop for the first time*

Click on the link shown in Figure 13 to download "WSL2 Linux kernel update package for x64 machine".



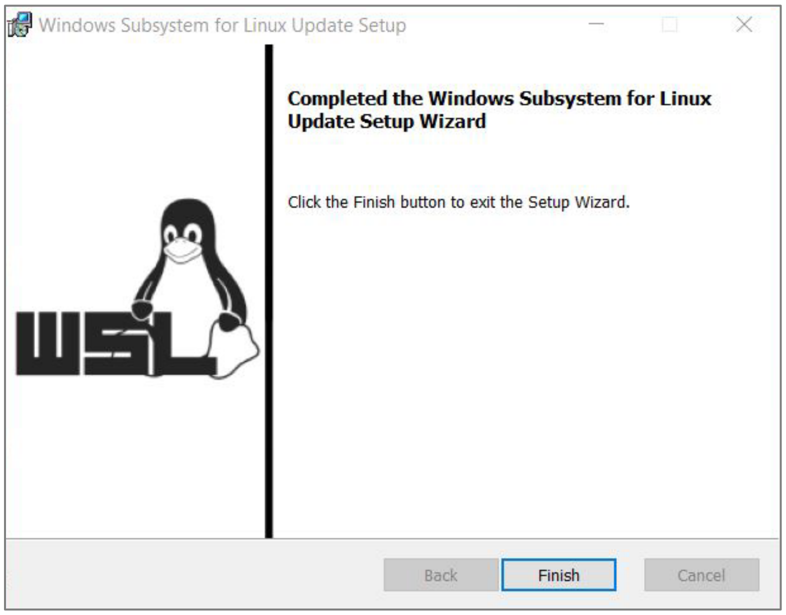
Step 4 - Download the Linux kernel update package

The Linux kernel update package installs the most recent version of the WSL 2 Linux kernel for running WSL inside the Windows operating system image. (To run WSL from the Microsoft Store, with more frequently pushed updates, use `wsl.exe --install` or `wsl.exe --update`.).

1. Download the latest package:
   - WSL2 Linux kernel update package for x64 machines

*Figure 13. Link to install WSL2 update from Microsoft website*

Then install the update following the steps by just clicking "Next" until you finish the installation as shown in Figure 14. Once you finished the installation, reopen the Docker Desktop to make sure that the message has disappeared.


Windows Subsystem for Linux Update Setup

**Completed the Windows Subsystem for Linux Update Setup Wizard**

Click the Finish button to exit the Setup Wizard.

*Figure 14. Finish installing WSL2 update*

### 1.2.4 *Downloading data*

*CyberWater* constitutes various modules that help users automatically access various data sources in real time from data source providers. Downloading data from data providers in real time can sometimes be time consuming. However, it is important to note that the *CyberWater* system ***does not slow down*** the data downloading process. It is the Internet communication bandwidth and the data servers altogether that affect the online retrieval time of the data when executing a model running in *CyberWater* in real time. For most of the exercises included in this manual, direct downloads of the forcing data are provided via HydroShare to skip this time-consuming step. A brief summary is given below which lists the download time between downloads directly from the NASA data sources used by *CyberWater* and a download from our pre-downloaded data stored in HydroShare. Please go to HydroShare and download the data needed for exercises described in this manual from the folder named <Examples Data\Forcing Data>.

*Table 1. A brief summary of the data downloading time and model run time with different specs*

| | PC1: 16GB memory and i7 processor | PC2: 8GB memory and 2.1 GHz processor |
|---|---|---|
| Time to directly download from NASA data source | **> 1 hour** (with a relative faster network) | **> 2 hours** (with a slow network) |
| **Time to directly download from HydroShare where the data are pre-downloaded and stored** | **< 1 minute** (with the Same Internet as above) | **< 5 minutes** (with the same Internet as above) |
| Time to run VIC or DHSVM examples in ***CyberWater**** | A few minutes | A few minutes |

*Note: One needs a PC with at least a memory of 6GB for our modeling examples. For a PC with 6GB, it takes about 10 minutes to run VIC5.0 with the water and energy balance mode for the West-Branch Susquehanna (WBS) river basin (17,700 km$^2$).

### 1.2.5 *Setting up NASA GESDISC DATA ARCHIVE*

*CyberWater* provides Data Agents capable of accessing NASA Earth database information in real time. However, the user needs to set up an account at https://urs.earthdata.nasa.gov/. After setting up the account, follow the procedures indicated at https://disc.gsfc.nasa.gov/earthdata-login to enable the download of NASA GESDISC DATA for your account. **This authorization step is crucial and commonly missed by users. Following the steps in the previous link, ensure that NASA GESDISC DATA ARCHIVE appears within the list of Approved Applications.** After this authorization step, all the Data Agents that access NASA services can be used in CyberWater.

# 2. Getting Familiar with CyberWater VisTrails

## 2.1 Visual Interface

Open *CyberWater*. This process may take a few minutes while *CyberWater* sets up its working environment. A starting console window will be displayed. Here, the user will be able to see the progress of the *CyberWater* Agents when they are executed in *VisTrails*. After the console is displayed, a splash screen of *CyberWater* will be prompted, showing various initialization steps. At the end of this process a *CyberWater* window will appear.

Figure 15 displays a standard *CyberWater VisTrails* view. The right panel shows information about the workflow, or the modules selected; the top left panel lists the current and previously saved workflows, and the bottom left panel holds the hierarchy of packages available in *CyberWater VisTrails*. Please note the search bar at the top of this panel, where the user can type the name of any available module for easy access. From here, any module can be dragged and dropped into the central area where the workflows are built, as shown in Figure 16.

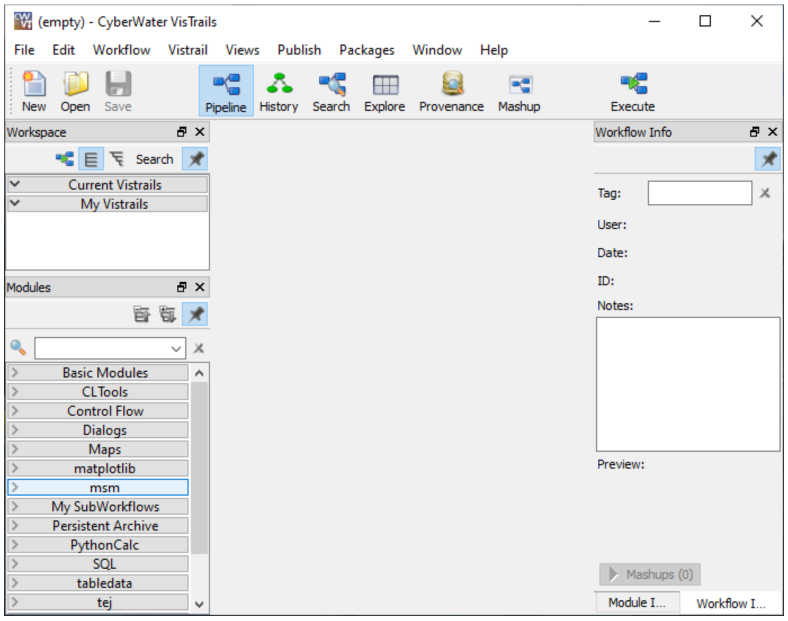


*Figure 15. CyberWater VisTrails view with the newly added 'msm' package.*

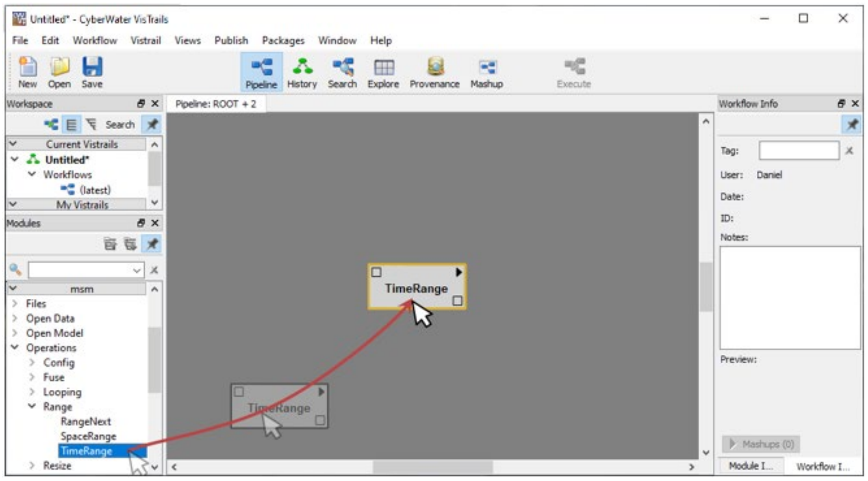


*Figure 16. Drag-and-drop feature of VisTrails, to bring modules into the current workflow.*

## 2.2 CyberWater folder structure

This section briefly explains locations of important user-related files inside *CyberWater*. The user might need to change, delete, or make a copy of these files. For more details about *CyberWater*, the user is referred to Salas et al. (2020) and Chen et al. (2022a, 2022b).

### 2.2.1 *CyberWater cache folder*

Every time a dataset is created in *CyberWater*, a cache file is created with a unique ID name. The program always checks if such dataset already exists in this folder before performing any operation (downloading data, executing a model, transforming a dataset, etc.). If it exists, *CyberWater* loads the cached file. If not, the execution continues normally.

**`CyberWater\VisTrails\vistrails\packages\msm\msm_core\msm_dao\helper\cache_files`**

### 2.2.2 *Data Agent independent files*

Sometimes users need to download large numbers of data files for simulations. If no partial information is stored when such an operation gets interrupted, the whole process would have to be re-started from the beginning. To prevent this, *CyberWater* stores a file at every time-step requested. This way, the program can continue downloading datasets just by continuingly bringing the files from where it was previously interrupted. These data files are located at:

**`DataAgentFolder\downloads\west_north_east_south\`**

These Data Agent Folders can be found at:

**`CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_data_agents`**

This location contains one folder per Data Agent installed in *CyberWater*.

### 2.2.3 *Model Agent simulation files*

*CyberWater* users are free to provide a local folder where model simulations are performed. However, if the user does not provide a folder, a temporary one will be created at the Model Agents Folder:

**`CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_model_agents`**

## 2.3 Debugging and Developing in CyberWater

An essential skill to have while using CyberWater is the ability to debug commonplace issues when executing a workflow. CyberWater offers a “Provenance” tool that allows the user to see the status of each module in their workflow and to read errors from modules that fail. Some modules also produce info, warning, and error messages in the CyberWater console, so checking for any hints there is essential to any productive workflow debugging.

While most errors are uncommon, it would be impossible to categorize all possible issues here. However, there are a few common errors that are worth understanding as you begin to become comfortable with CyberWater. One example that may happen time to time with data agents is a failure to reach a data source at a certain URL. Unfortunately, data sources have a habit of modifying their location and structure without warning, so CyberWater data agents may temporarily fall out of date with the source it is supposed to sample from. To fix this issue, simply open up the module and update the URL path encoded within the module. The CyberWater team will also periodically update these modules to guarantee they remain synchronized with the many data sources offered. Another common issue is “Module return none” when using certain modules that interface with GRASS GIS. Because of the way the GRASS API is developed, it is not easy to report out the details of an error that occurs in GRASS GIS. “Module return none” is diagnostic of an issue within GRASS itself when a command is executed. If you run into this issue, check your GRASS installation and your module inputs to make sure that the GRASS command is executing successfully.

Notes:

- To ensure that any changes you make to the workflow take effect, it's essential to clear the cache before re-executing the model. This is because the program prioritizes cached files, which can prevent new updates from being applied. You can easily clear the cache from the workflow menu shown in Figure 17.
- In CyberWater, a cache file for each execution is always saved at this folder “C:\CyberWater\VisTrails\vistrails\packages\msm\msm_core\msm_dao\helper\cache_files”. Users can delete the data in this folder periodically to save space in their local machine.

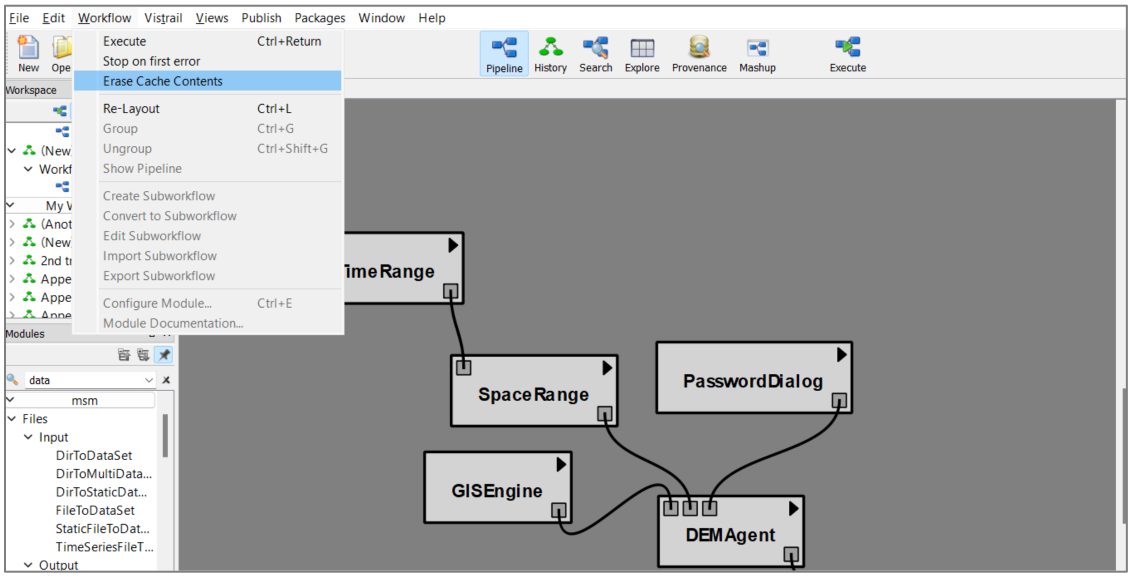


*Figure 17. Screen shot showing how to clear the cache.*

### 2.3.1 *PythonSource*

One of the important modules in CyberWater is the *PythonSource* module, which allows users to implement small chunks of Python code directly into their workflow. While CyberWater works hard to reduce any need for programming from the user, sometimes, a few lines of simple Python codes may be needed from the user to accommodate some models' unique requirements, such as conducting simple calculations to provide model needed information based on data given by the providers. In such situations, the *PythonSource* module can be utilized to fill in the "gap" between modules provided by CyberWater to complete the user's workflow task. It is very useful as a sort of "glue" for CyberWater workflows, sticking together parts of a workflow that otherwise would be difficult to combine for the current use case. They are also extremely useful in module development and debugging, as you can test new code without needing to recompile a module each time. The possibilities with this module are endless!

To use a *PythonSource* module, add one to your workflow and click "Configure" in the Module Info panel on the side of CyberWater to get the configuration panel as shown in Figure 18 (a). The user will need to add inputs and outputs to use the *PythonSource*. The first column is used to name the input/output, and the second is to define its type. If the user does not know what type to use, they should use "Variant", which is a generic class that works with all types. To access the input data, simply access the variable with the name of the input you defined. For example, if the user put "input1" as the *Input Port Name* in the first column for a certain input and its type to be "Variant (org.vistrails.vistrails.basic)". Let's say this input equals to "1" and we want to print it. Write the following code inside the source text area (the blank space in the lower half of the configuration window):

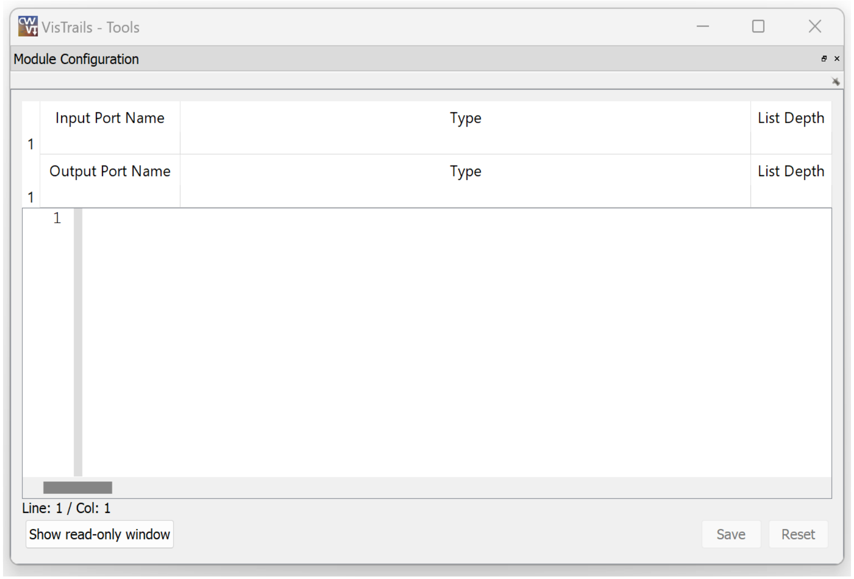
```
input1=1
print input1
```

Click save to the configuration window and execute the workflow from the execution button Execute. You will see the value “1” printed in CyberWater console. Similarly, to set an output, simply set the variable with the name of the output you defined. For example, if the user put “output1” as the *Output Port Name* in the first column for a certain output and its type to be “Variant (org.vistrails.vistrails.basic)”. Then added the following code inside the source text area to output “ Helloworld!”:

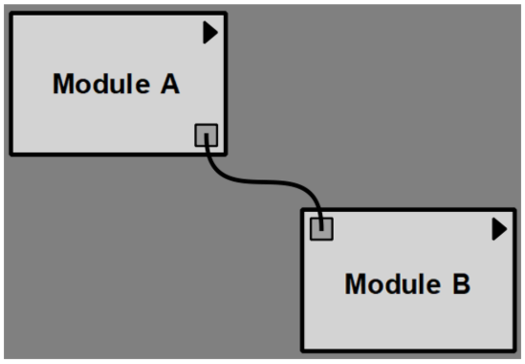
```
output1=”Hello world!”
print output1
```

After saving and executing it you will see ”Hello world!” on CyberWater console.

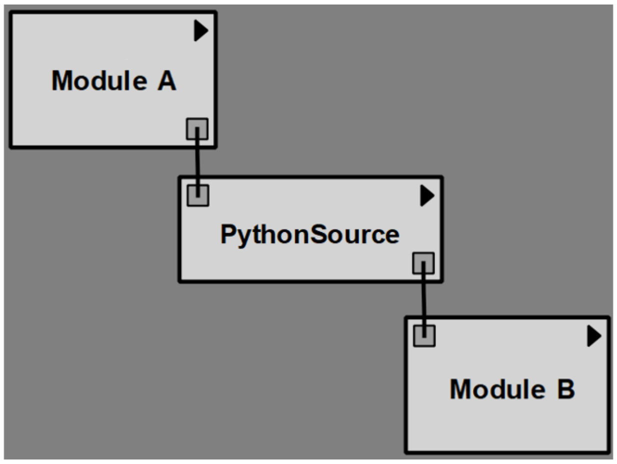
As explained before, the main idea of using the *PythonSource* module is to perform certain tasks between different modules (that’s why it has inputs and outputs). Let’s assume that we have two modules: “*Module A*” and “*Module B*”. *Module A* outputs data which is then used as an input to *Module B* as shown in Figure 18 (b). The *PythonSource* can be used to perform special operations on the data from *Module A* before adding it to *Module B* as shown in Figure 18 (c). Configure the *PythonSource* module by adding a new name to *Input Port Name* (e.g. Outputs from Module A) and select the data type (e.g. Variant). Then add a new name to the *Output Port Name* (e.g. Inputs to Module B) and select the data type (e.g. Variant). Finally, add the python code for your special operation inside the source text area. The previous steps show how to write Python statements to be executed as part of a workflow. Note that, more than one input or output modules can be added to the *PythonSource* depending on the application needs.



*(a)*



(b)



(c)

*Figure 18. (a) The configuration panel for PythonSource. (b) Output data from Module A used as inputs to Module B. (c) Adding a PythonSource module in between the two modules to perform certain task on the data.*

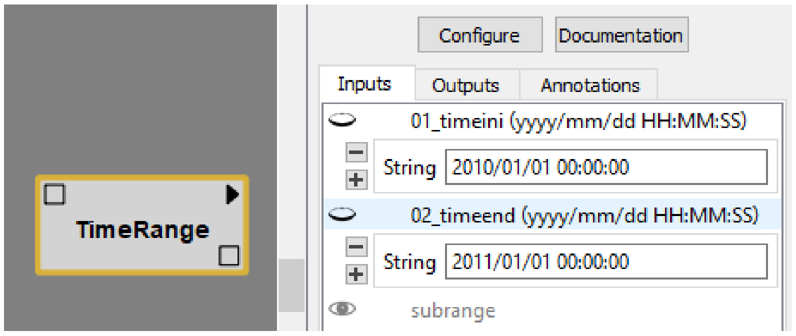
## 2.4 Illustration of workflow with a simple CyberWater example

In this section, a simple example is used to demonstrate the basic *CybeWater* workflow concept and its working environment via some of the functionalities developed in the *CyberWater* system. Specifically, a streamflow case study that can automatically access the USGS streamflow data in real time is used. This automatic data retrieval process is one of the many features *CyberWater* has.

### 2.4.1 *Define the study area*

All simulations require the user to specify a *time-window* and the *spatial extent* where the study case will be conducted. This will guide all downloads of hydro-meteorological information, as well as the creation of most of the parameter-files required for the models to run.

The *TimeRange*[1] module allows the user to define the starting and ending dates of the simulation (see Figure 19). The user needs to be aware that all the individual Data Agents will bring the information available within that time-span with whichever temporal resolution available.

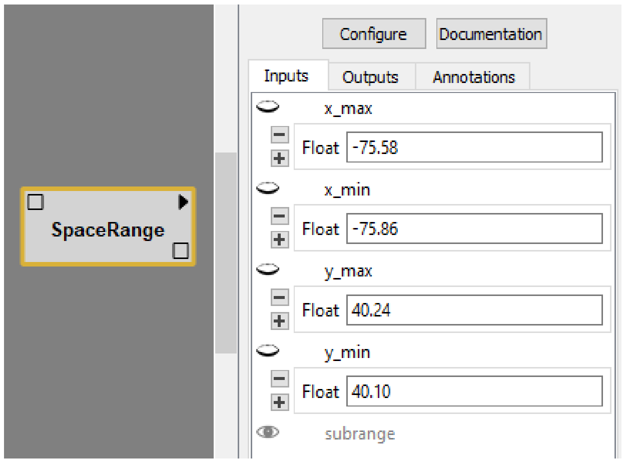


01_timeini (yyyy/mm/dd HH:MM:SS)
String 2010/01/01 00:00:00
02_timeend (yyyy/mm/dd HH:MM:SS)
String 2011/01/01 00:00:00

*Figure 19. TimeRange module of CyberWater, defining a time window for a simulation.*

[1] Located at msm>Operations>Range>TimeRange

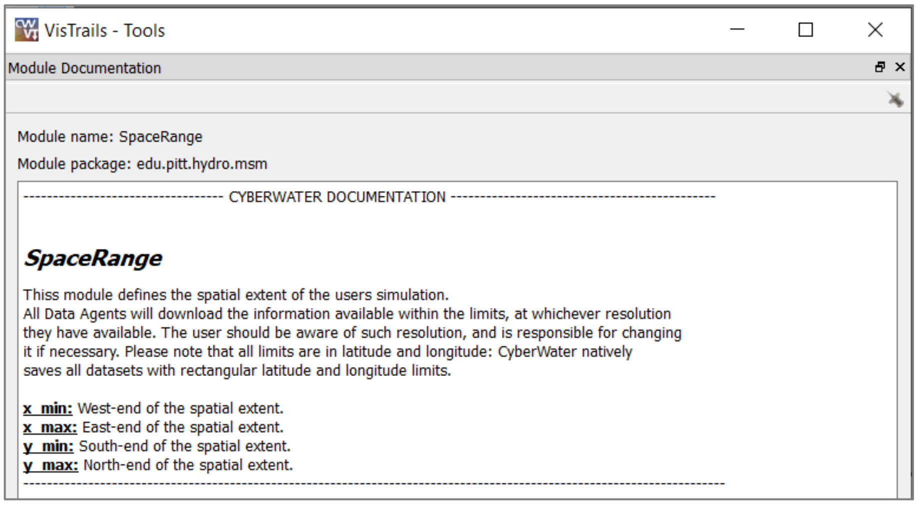
The *SpaceRange*[1] module allows the user to define the spatial extent of a simulation (see Figure 20). The boundaries of such space can be specified as the right, left, top, and bottom coordinates (e.g., if the projection of the datasets is latitude-longitude, such coordinates would be: east, west, north, and south, respectively). The coordinates that are shown in Figure 20 covering the area of the French Creek basin in southeast Pennsylvania.[2]



x_max: Float -75.58
x_min: Float -75.86
y_max: Float 40.24
y_min: Float 40.10

*Figure 20. SpaceRange module of CyberWater, defining a spatial extent for a simulation.*

All modules in *CyberWater* come with dedicated documentation that explains their purpose and uses in detail. Also, such documentation includes the requirements on each of the inputs, as well as the properties of the outputs. For Data Agents, the documentation further includes information about spatio-temporal coverage and resolutions. This documentation can be accessed by selecting the Documentation button shown in Figure 20 (see top row in the right configure panel), which is displayed in Figure 21.



Module name: SpaceRange

Module package: edu.pitt.hydro.msm

---------------------------------- CYBERWATER DOCUMENTATION ----------------------------------------------

***SpaceRange***

Thiss module defines the spatial extent of the users simulation.
All Data Agents will download the information available within the limits, at whichever resolution they have available. The user should be aware of such resolution, and is responsible for changing it if necessary. Please note that all limits are in latitude and longitude: CyberWater natively saves all datasets with rectangular latitude and longitude limits.

**x_min:** West-end of the spatial extent.
**x_max:** East-end of the spatial extent.
**y_min:** South-end of the spatial extent.
**y_max:** North-end of the spatial extent.

------------------------------------------------------------------------------------------------------------

*Figure 21. Documentation of the SpaceRange module in CyberWater.*

[1] Located at msm>Operations>Range>SpaceRange

[2] https://waterdata.usgs.gov/nwis/uv?01472157

Connect the *TimeRange* output with the *SpaceRange* 'subrange' input by drawing a line between the two ports. Note that the label of each input will be prompted if the mouse pointer is left on top of the port for a brief period of time. This is shown in Figure 22. Please note that sometimes *CyberWater VisTrails* automatically connects previously existing modules with newly dropped ones. While the system tries to do this intelligently, it does not always connect them to the right ports, so make sure to double check the name of the ports connected afterwards. This tuple *SpaceRange-TimeRange* constitutes a case study in *CyberWater*.

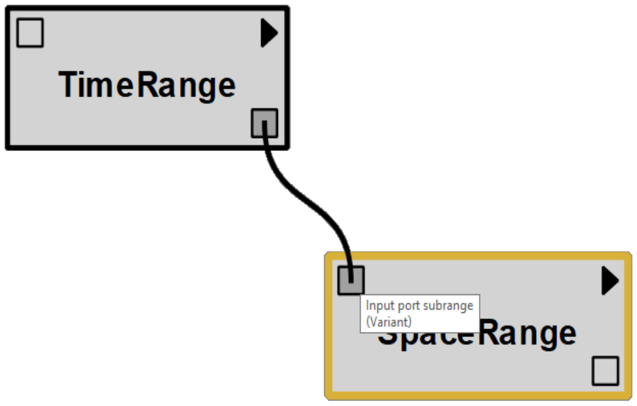


*Figure 22. Display of the name of the input port.*

*CyberWater* provides users with various so-called Data Agents to directly access various hydro-meteorological data from diverse online data sources. All Data Agents require, for any case study, the information of the spatial and temporal extents of the downloaded files. These files are the forcing inputs required by the selected hydrological models.

*CyberWater* comes with a set of visualization tools that allow the user to explore the data visually, by either displaying time-series, static spatial charts, or dynamic animations of the spatial-temporal information.

For example, if the user uses the *UsgsAgent*[1], a Data Agent for accessing USGS online data sources, and the *msmShowChart*[2], a module for displaying time series, to build the workflow as shown in Figure 23 with default values, the chart depicted in Figure 24 will automatically be prompted after clicking the *Execute* icon on the top toolbar of *CyberWater VisTrails*. The example workflow can be found either under the folder **C:\CyberWater\VisTrails\examples**, or on CyberWater online webpage <1. USGS Data Retrieval and Visualization>.

[1] Located at msm>Open Data>UsgsAgent

[2] Located at msm>Visualization>msmShowChart

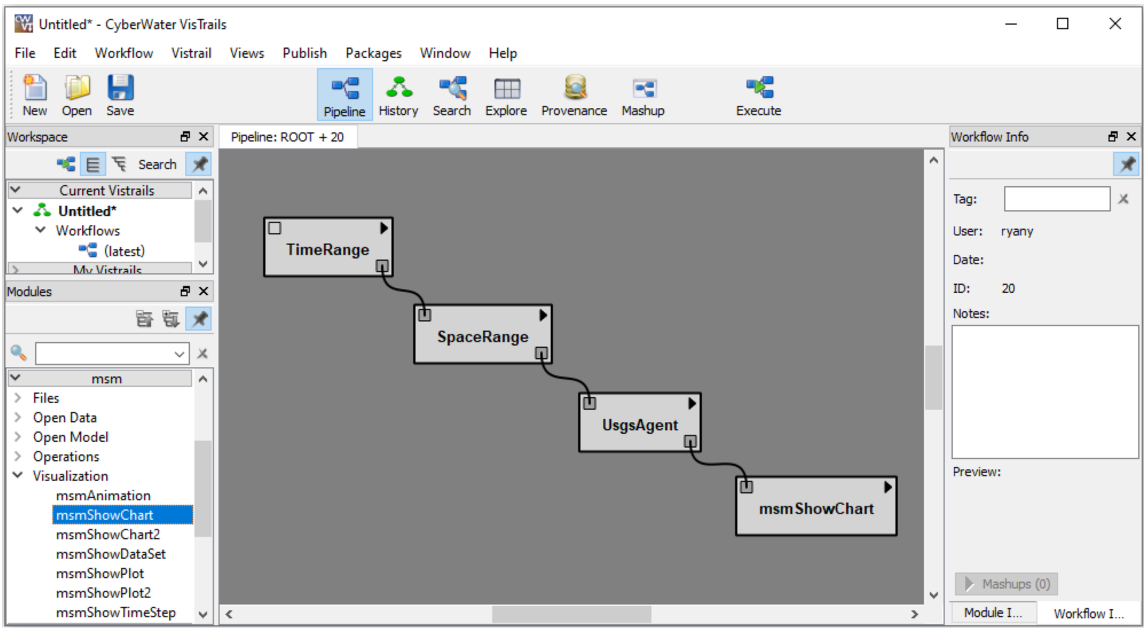


*Figure 23. Workflow depicting a case study to download and plot streamflow values of the French Creek basin.*

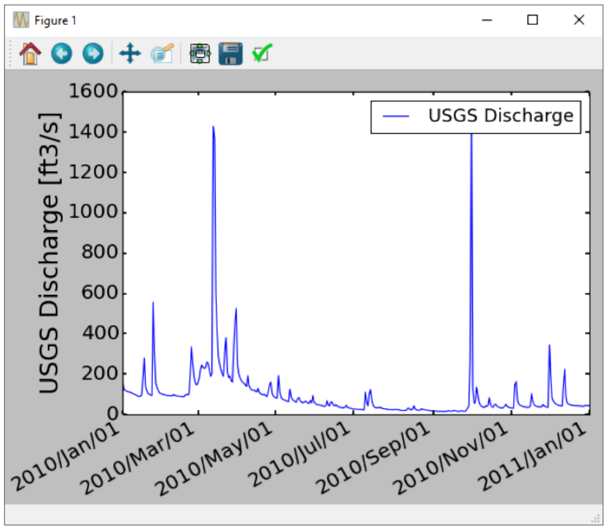


*Figure 24. A plot of the French Creek streamflow time series retrieved by CyberWater's USGS Agent.*

At this point, *CyberWater VisTrails* colors the modules of a workflow, following the color standard below:

| | | |
|---|---|---|
| Green | 🟩 | Successfully executed |
| Yellow | 🟨 | Cached in memory |
| Red | 🟥 | Errors found |
| Orange | 🟧 | Suspended execution |
| Blue | 🟦 | Not executed |

Please note that some of the workflows for the exercises illustrated in this manual can be downloaded from the HydroShare <CyberWater_Beta_Examples>. However, these workflows cannot be directly executed in general unless the users enter their own credentials for some data agents and update the file paths according to their own local machine. It is highly recommended that new users try to follow the manual to build their own workflows rather than using the pre-built ones to gain deeper understanding of CyberWater. Also, the users are recommended to get their forcing data directly from the data providers using the CyberWater's data agents in the workflow to gain the experience on how the forcing data are directly "flown" into the model in the CyberWater system. For cases where a larger amount of forcing data is required, the users can get these forcing data provided by us from the HydroShare <Examples Data\Forcing Data> to save time.

# 3. Case Study

*CyberWater* constitutes various modules that help users automatically access various data in real time from diverse data source providers (e.g., forcing data and USGS streamflow data), execute models, and produce model simulation results. In this section, modeling examples and procedures on how to use the *CyberWater* system to achieve these tasks are illustrated through a step-by-step instruction.

## 3.1 VIC (v4.0) and Routing Model simulations using the VIC Model Agent and Routing Agent with provided data

This section shows how to build a full example of a streamflow simulation workflow using the Variable Infiltration Capacity (VIC) model via the *CyberWater* VIC Agent. The first example is a water balance simulation of the French Creek basin[1] using VIC version 4.0 (v4.0), located in the southeast of Pennsylvania, United States. This is a small watershed with a drainage area of 153 km$^2$ (59.1 mi$^2$). Please note that all the parameter values in this simulation are default values that can be modified in the future by the user through a parameter calibration process.

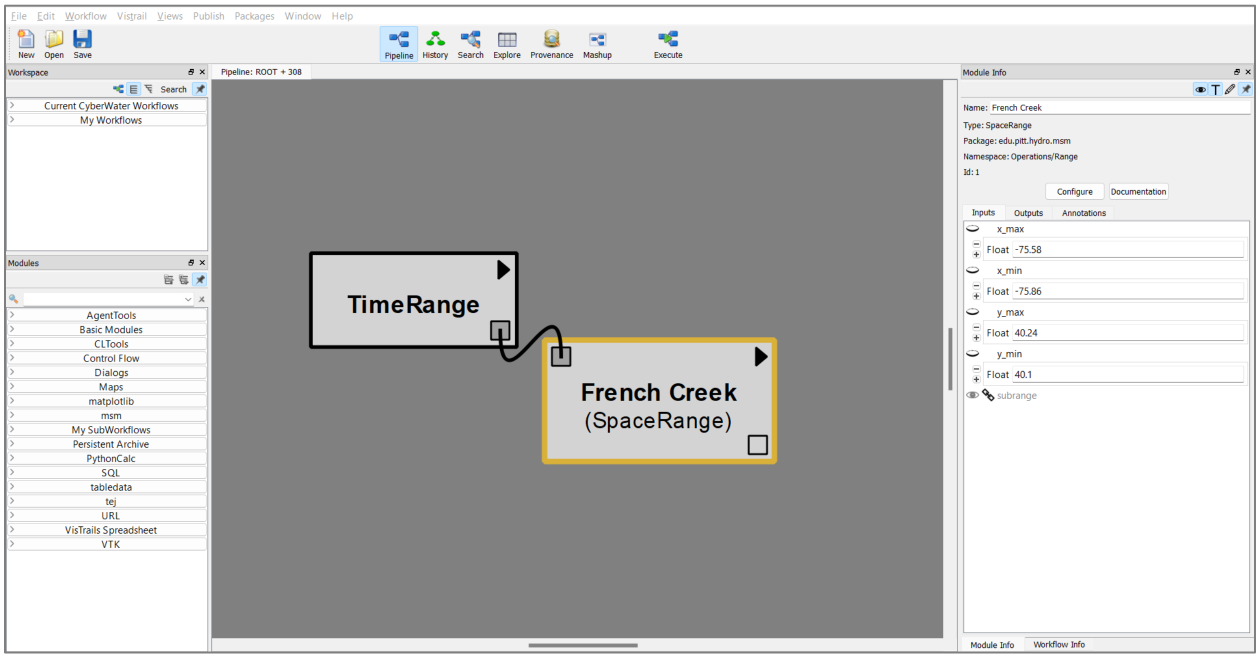
### 3.1.1 *Creating a simulation with the VIC model Agent*

- Add a *TimeRange*[2] box for a simulation between 2010/02/01 00:00:00 (timeini) and 2010/05/01 00:00:00 (timeend).
- Add a *SpaceRange*[3] box with the same limits shown in Figure 20: -75.58,-75.86,40.24, 40.10 (x_max, x_min, y_max, y_min respectively). Connect the '*timerange*' output of the *TimeRange* box with the '*subrange*' input of the *SpaceRange* box. Note that the *Module Info* panel (to the right) allows the user to assign an alias to the box. For clarity, this module will be named '*French Creek*', as shown in Figure 25.

[1] https://waterdata.usgs.gov/nwis/uv?01472157

[2] Located at msm>Operations>Range>TimeRange

[3] Located at msm>Operations>Range>SpaceRange



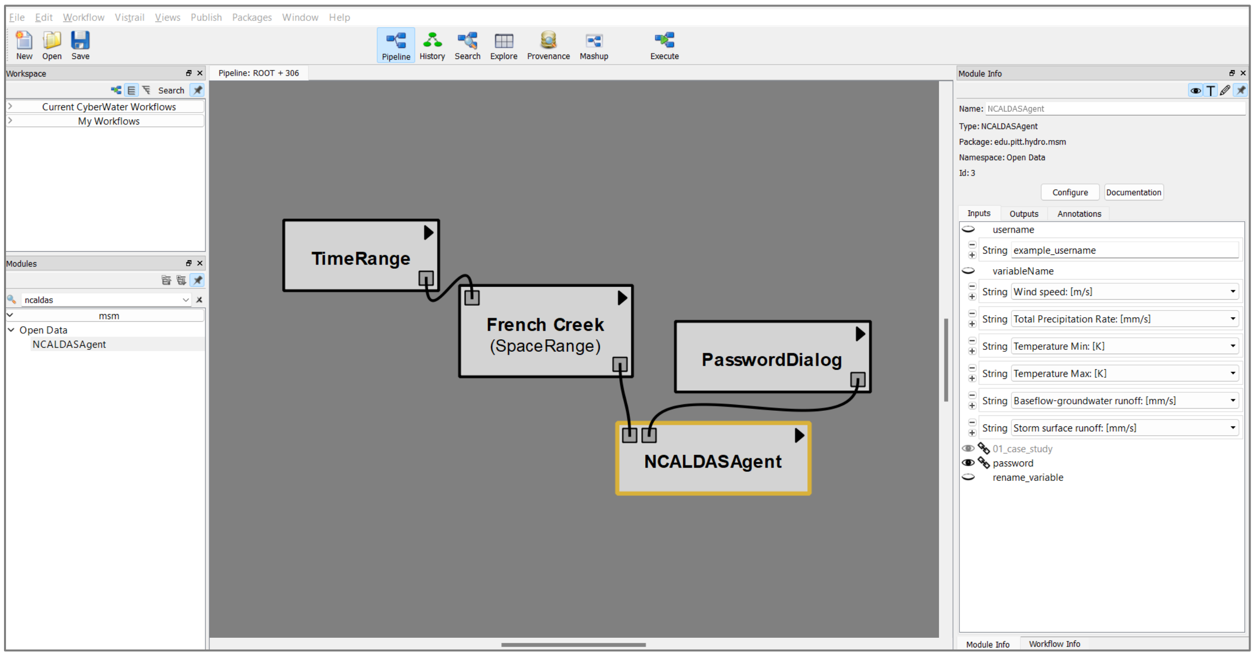
Name: French Creek
Type: SpaceRange
Package: edu.pitt.hydro.msm
Namespace: Operations/Range
Id: 1

x_max Float -75.58
x_min Float -75.86
y_max Float 40.24
y_min Float 40.1
subrange

*Figure 25. Case study definition with Temporal and Spatial specification in CyberWater.*

- Now, to access the forcing information required for the simulation, the NCALDAS Data Agent will be used. This agent accesses the NASA Earth Data database to bring NASA's spatially organized hydrometric data information to the local machine automatically.
- Add an *NCALDASAgent*[1] module and fill it with your NASA Earth Data credentials. The user is also able to use the *PasswordDialog*[2] module to collect password information at runtime instead of saving the information in the data agent. To use *PasswordDialog*, first click the icon ( 👁 ) to the left of the *'password'* variable in the *NCALDASAgent* to make the password input visible, then connect the *PasswordDialog* output to the new input square. In *NCALDASAgent*, the input '*variableName*' is a list-box with all the variables available. Select '*Wind speed: [m/s]*', '*Total Precipitation Rate: [mm/s]*', '*Temperature Min: [K]*' and '*Temperature Max: [K]*' as shown in Figure 26. These are the forcing data required by VIC4 model. Then activate the output ports corresponding to the previous four forcing data from the *Outputs* panel as shown in Figure 27. It takes about 7 to 10 minutes to download these forcing data from NASA database, depending on the internet speed of the local machine. Note that the relevant information is stored locally at ('CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_data_agents\NCALDASAgent\downloads') and will be used again if the user requests the same spatial and temporal extent for that specific NCALDAS variable. This prevents redownloading the same information in the future. The user is free to delete these files afterwards. At this point, the user is encouraged to use the '*msmShowChart'* (as displayed in Figure 23) to plot a time series of the daily *Wind Speed* (for example), averaged around the area defined in the case study by connecting the *msmShowChart* with the output port of the *Wind Speed [m/s]*. The plot obtained is an area-average of the spatial data of Wind Speed.

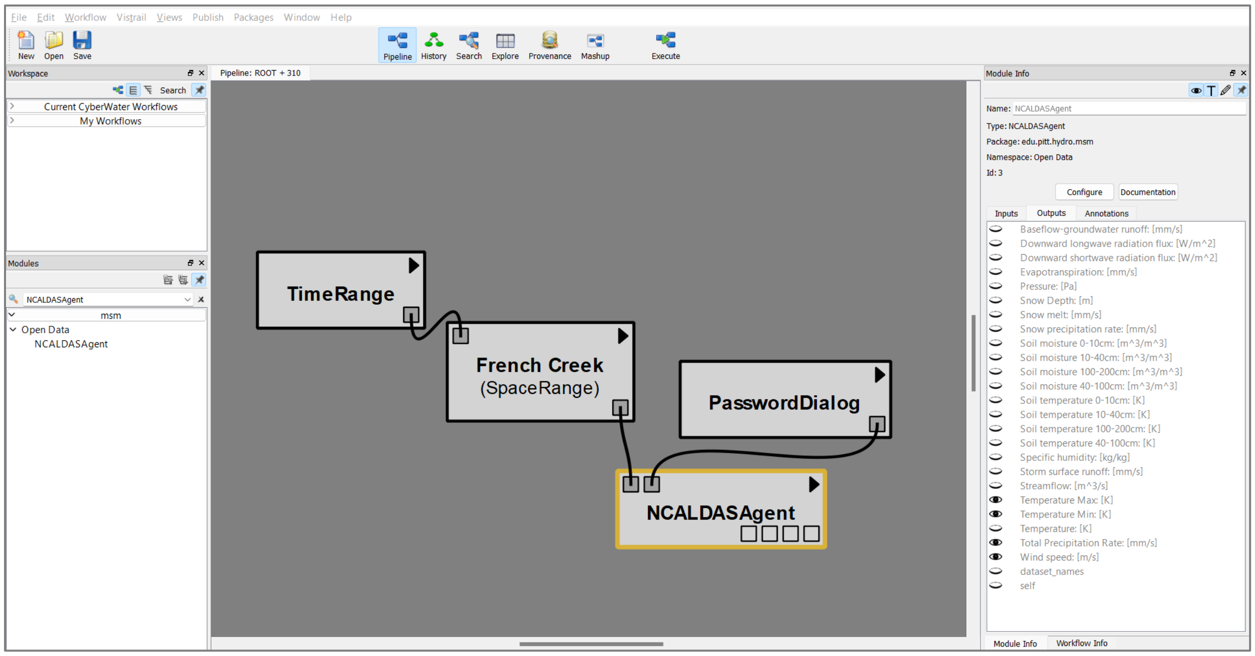
[1] Located at msm>Open Data>NCALDASAgent
[2] Located at Dialogs>PasswordDialog


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Current CyberWater Workflows
My Workflows
Modules
ncaldas
msm
Open Data
NCALDASAgent
Pipeline: ROOT + 306
TimeRange
French Creek (SpaceRange)
PasswordDialog
NCALDASAgent
Module Info
Configure
Documentation
Inputs Outputs Annotations
Module Info Workflow Info

Name: NCALDASAgent
Type: NCALDASAgent
Package: edu.pitt.hydro.msm
Namespace: Open Data
Id: 3

username
String example_username
variableName
String Wind speed: [m/s]
String Total Precipitation Rate: [mm/s]
String Temperature Min: [K]
String Temperature Max: [K]
String Baseflow-groundwater runoff: [mm/s]
String Storm surface runoff: [mm/s]
01_case_study
password
rename_variable

*Figure 26. NCALDASAgent module to download forcing data.*


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Current CyberWater Workflows
My Workflows
Modules
NCALDASAgent
msm
Open Data
NCALDASAgent
Pipeline: ROOT + 310
TimeRange
French Creek (SpaceRange)
PasswordDialog
NCALDASAgent
Module Info
Configure
Documentation
Inputs Outputs Annotations
Module Info Workflow Info

Name: NCALDASAgent
Type: NCALDASAgent
Package: edu.pitt.hydro.msm
Namespace: Open Data
Id: 3

Baseflow-groundwater runoff: [mm/s]
Downward longwave radiation flux: [W/m^2]
Downward shortwave radiation flux: [W/m^2]
Evapotranspiration: [mm/s]
Pressure: [Pa]
Snow Depth: [m]
Snow melt: [mm/s]
Snow precipitation rate: [mm/s]
Soil moisture 0-10cm: [m^3/m^3]
Soil moisture 10-40cm: [m^3/m^3]
Soil moisture 100-200cm: [m^3/m^3]
Soil moisture 40-100cm: [m^3/m^3]
Soil temperature 0-10cm: [K]
Soil temperature 10-40cm: [K]
Soil temperature 100-200cm: [K]
Soil temperature 40-100cm: [K]
Specific humidity: [kg/kg]
Storm surface runoff: [mm/s]
Streamflow: [m^3/s]
Temperature Max: [K]
Temperature Min: [K]
Temperature: [K]
Total Precipitation Rate: [mm/s]
Wind speed: [m/s]
dataset_names
self

*Figure 27. Activating output ports corresponding to the chosen forcing data in NCALDASAent*

- Add the *VICAgent* [1]module into the workflow. This example requires toggling the visibility of the *TMAX* and *TMIN* inputs just like what was done with the integration of the *PasswordDialog*

[1] Located at msm>Open Model>VicAgent

module. This can be achieved by clicking on the "eye" icons at the left of the input names, as shown in Figure 28.

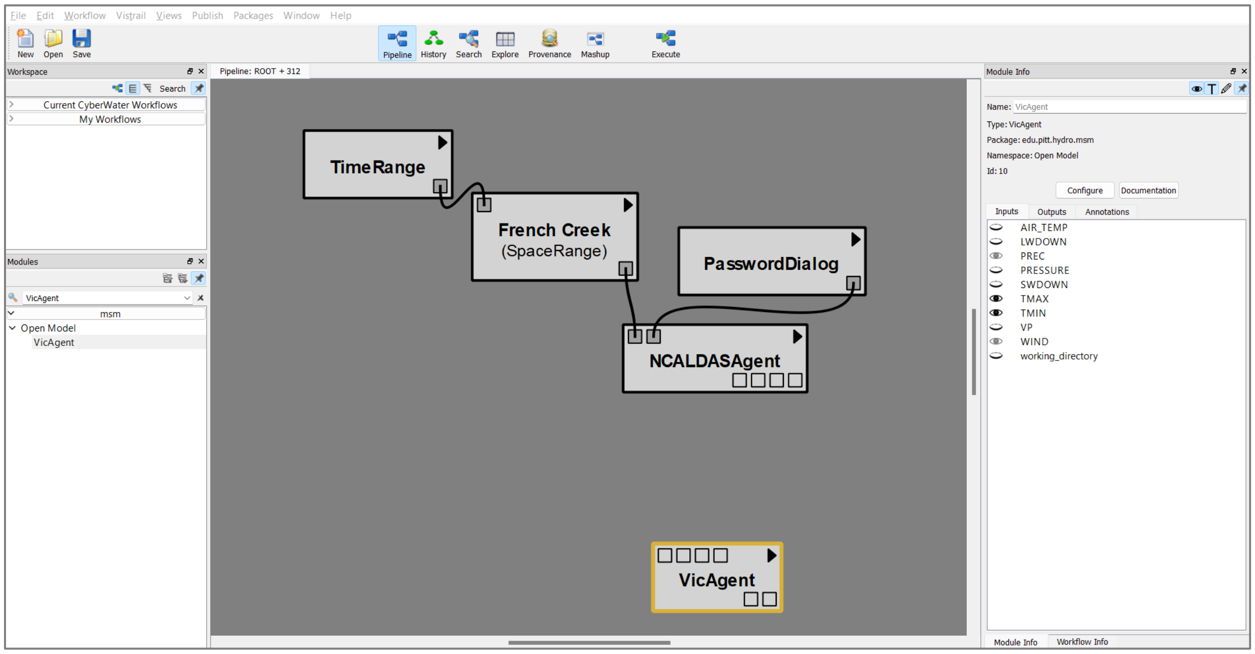
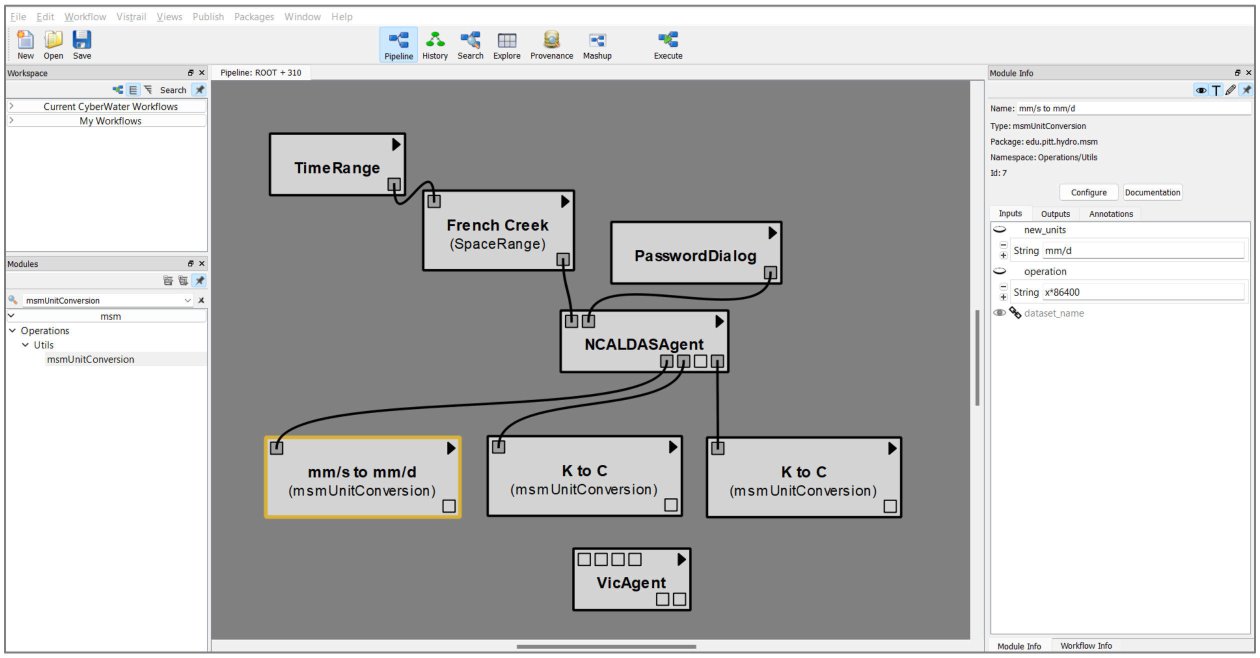


*Figure 28. Toggling visibility of the VIC Agent inputs, TMAX and TMIN.*

- *VICAgent* requires inputs to be included in specific units, depending on the individual variables. The details of each input are included in the *VICAgent* documentation available via the *Documentation* button in *VisTrails*. For this example, Temperature has to be in Celsius, and Precipitation needs to be in mm per day. To perform these transformations, the *msmUnitConversion*[1] module can be used. To transform the Kelvin units from both NCALDAS datasets into Celsius, set the input '*operation*' to be 'x-273.15'. This '*operation*' allows the user to execute simple mathematical operations using 'x' as the variable that represents the dataset given as input. Create another *msmUnitConversion* box to transform precipitation per second to daily precipitation, by setting 'operation' to 'x*86400'. This module also requires the user to indicate the new resulting units of the transformed dataset. For the first module set '*new_units*' to 'C', and for the second set '*new_units*' to 'mm/day'. The new workflow is shown in Figure 29.

[1] Located at msm>Operations>Utils>msmUnitConversion


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Current CyberWater Workflows
My Workflows
Modules
msmUnitConversion
msm
Operations
Utils
msmUnitConversion
Pipeline: ROOT + 310
TimeRange
French Creek (SpaceRange)
PasswordDialog
NCALDASAgent
mm/s to mm/d (msmUnitConversion)
K to C (msmUnitConversion)
K to C (msmUnitConversion)
VicAgent
Module Info
Configure
Documentation
Inputs
Outputs
Annotations
Workflow Info

Name: mm/s to mm/d
Type: msmUnitConversion
Package: edu.pitt.hydro.msm
Namespace: Operations/Utils
Id: 7

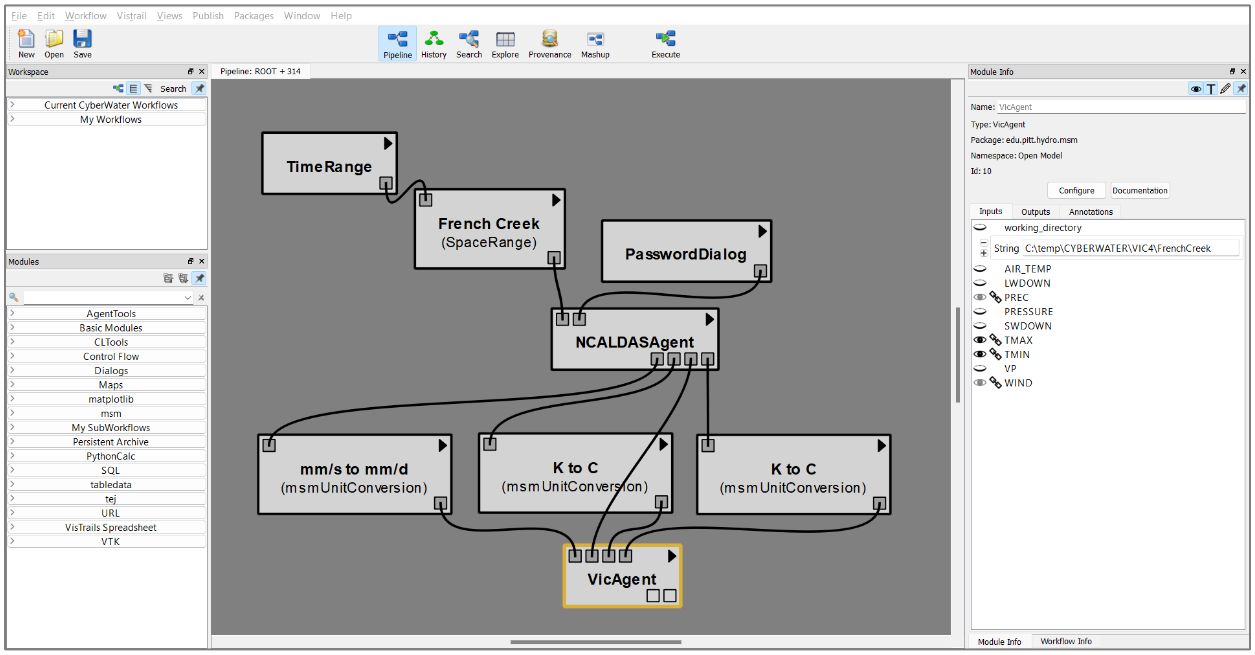
new_units
String mm/d
operation
String x*86400
dataset_name

*Figure 29. Adding msmUnitConversion modules to perform preprocessing of the data before it goes to VICAgent.*

- Proceed to connect each forcing Data Agent with the *VICAgent* box. The resulting workflow looks as depicted in Figure 30 from Figure 29. The user is free to add (or not) a folder where all the simulation files will be stored with the '*working_directory*' input. The default folder is located at *'CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_model_agents\VicAgent\tmp'*. It is recommended that users add their own directories especially if they want to change the default parameters, initial state, or perform further analysis to the results outside CyberWater environment. The VicAgent uses default parameters for soil, vegetation, snow, etc. These parameters can be changed by executing the model for the first time, then a folder named "params" will be created inside the working directory set in the *VICAgent* module. This folder includes the parameter files used by *VICAgent* for that first time execution. The user can modify these parameter files and then re-execute the model again keeping the same working directory in *VICAgent* so that it can read the modified parameter files. In addition, to ensure that the recent changes take effect in the next run, please delete the cache file from the following directory before starting the second execution:

  *C:\CyberWater\VisTrails\vistrails\packages\msm\msm_core\msm_dao\helper\cache_files*

  Furthermore, if a state file is present in the specified working directory location, it will be used for the execution of the *VICAgent*. If not, The VIC model runs with zeros as initial state of soil ice, snow, and canopy water contents as default and with standard default initializations on soil moisture and temperatures in the VIC code. Please note that Windows imposes a maximum path length of 255 characters, so the user must keep their chosen directory relatively short to prevent an error with an unreachable file.

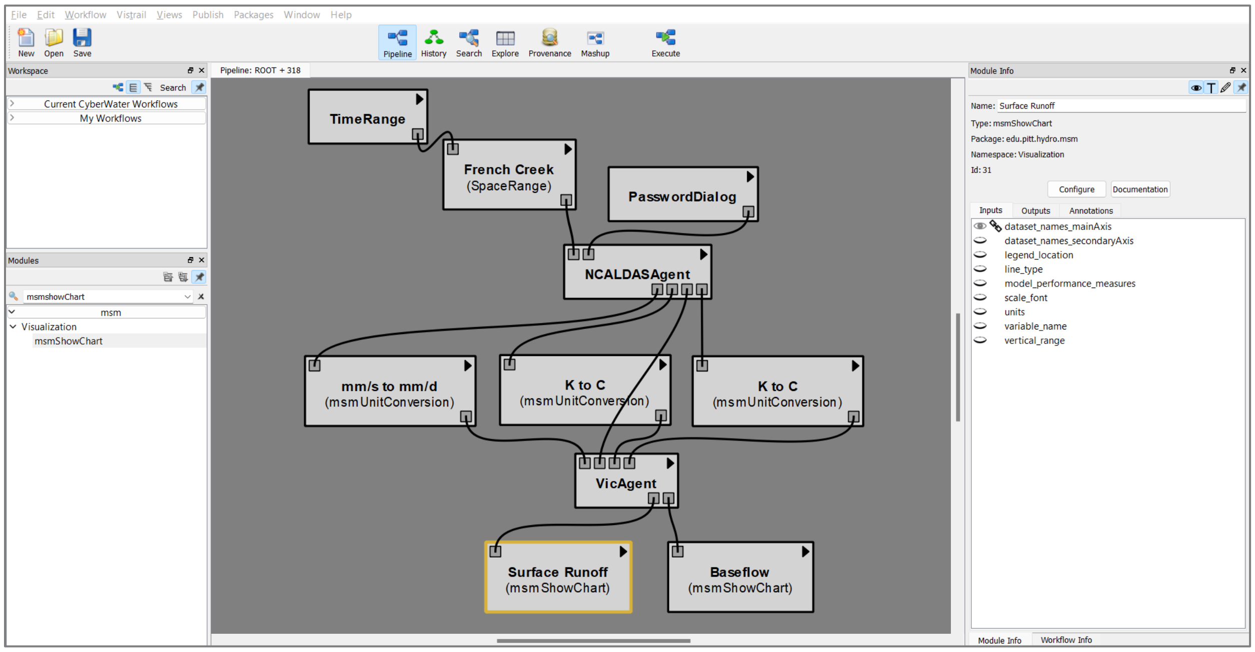


Name: VicAgent
Type: VicAgent
Package: edu.pitt.hydro.msm
Namespace: Open Model
Id: 10

working_directory
String C:\temp\CYBERWATER\VIC4\FrenchCreek
AIR_TEMP
LWDOWN
PREC
PRESSURE
SWDOWN
TMAX
TMIN
VP
WIND

*Figure 30. Full workflow where VICAgent is fed with Data Agent forcing data.*

- The last step before executing this workflow is to set up the visualization tools, which will allow us to see the results of the simulation. This can be achieved by adding two '*msmShowChart*' boxes into the workflow and naming them '*Baseflow*' and '*Surface Runoff*', as shown in Figure 31.


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Current CyberWater Workflows
My Workflows
Modules
msmshowChart
msm
Visualization
msmShowChart
Pipeline: ROOT + 318
TimeRange
French Creek (SpaceRange)
PasswordDialog
NCALDASAgent
mm/s to mm/d (msmUnitConversion)
K to C (msmUnitConversion)
K to C (msmUnitConversion)
VicAgent
Surface Runoff (msmShowChart)
Baseflow (msmShowChart)
Module Info
Configure
Documentation
Inputs
Outputs
Annotations
Module Info
Workflow Info

Name: Surface Runoff
Type: msmShowChart
Package: edu.pitt.hydro.msm
Namespace: Visualization
Id: 31

dataset_names_mainAxis
dataset_names_secondaryAxis
legend_location
line_type
model_performance_measures
scale_font
units
variable_name
vertical_range

*Figure 31. Workflow used to create a VIC simulation and plotting the results as time series using CyberWater.*

- Please note that this example uses the two outputs offered by default in the VIC Agent. However, if the user goes to the *Outputs* tab in the *Module Info* panel (to the right), more outputs available to be displayed are listed. Additionally, to find the global file for VIC and the other outputs of running VIC, the user is directed to the ‘*working_directory*’ they set previously, or the default directory at: ‘*CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_model_agents\VicAgent\tmp*’.

- After executing this workflow, the two charts that are shown in Figure 32 and Figure 33 will be prompted.

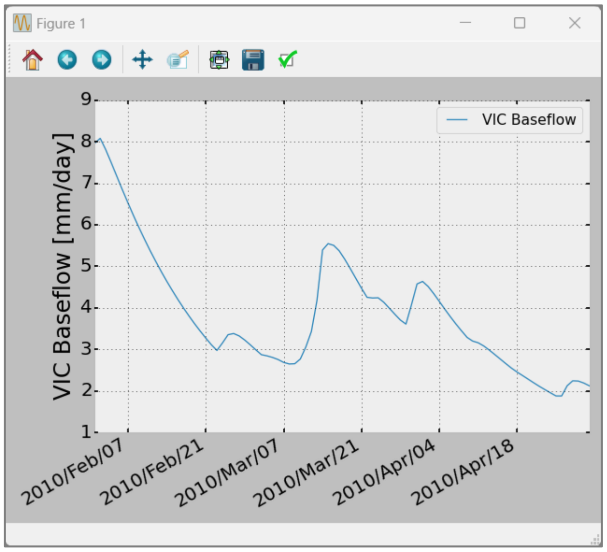


Figure 32. Resulting subsurface baseflow from the VIC simulation of the French Creek in PA.

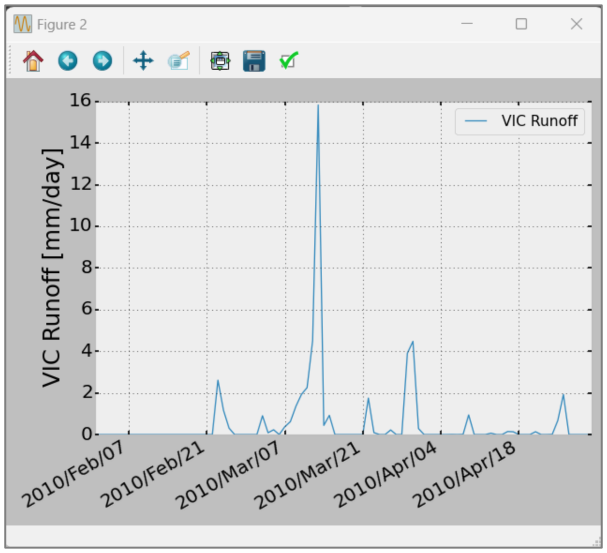
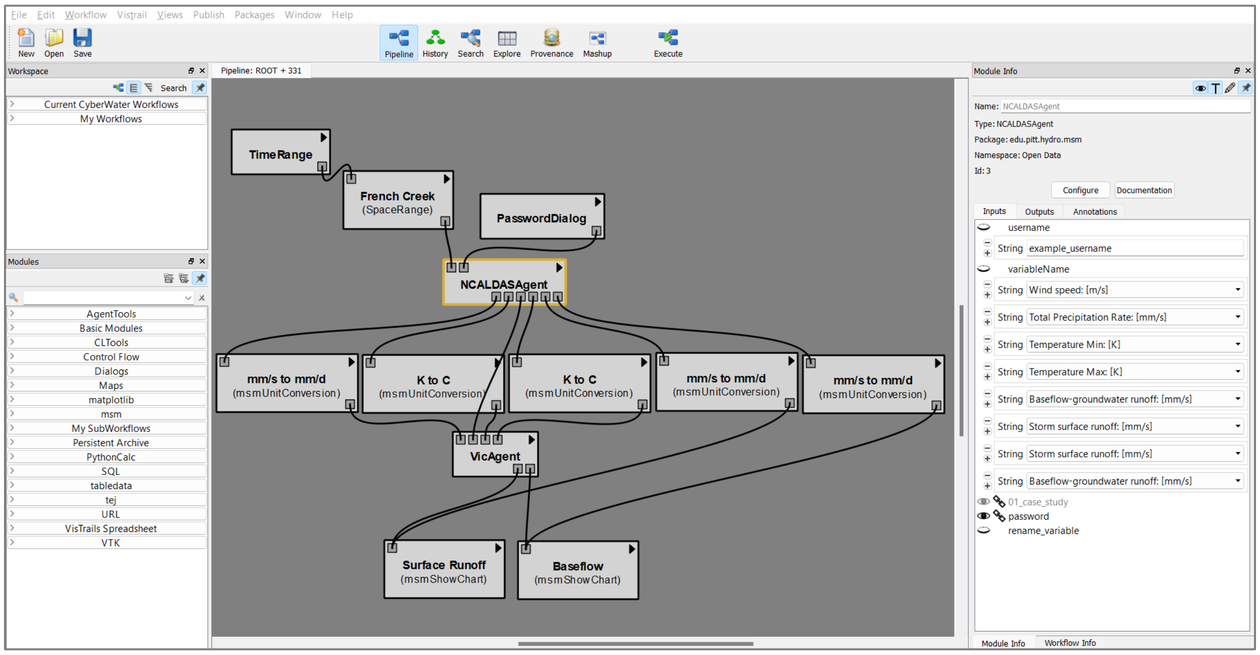


Figure 33. Resulting surface runoff from the VIC simulation of the French Creek in PA.

- Now, in order to compare the obtained results with a different model, the user can add two more ‘variableName’ in the existing *NCALDASAgent* and retrieve NASA’s values of *Runoff* and *Baseflow* as: ‘*Storm surface runoff: [mm/s]*’, and ‘*Baseflow-groundwater runoff: [mm/s]*’. The output ports for both data should be activated. Since these values are given in mm per second, the respective unit transformation needs to be performed, using the *msmUnitConversion* module with an ‘*operation’* of ‘*x*86400*’ in units of ‘*mm/day*’. The outputs of these operations can be connected directly to the plotting modules, as depicted in Figure 34. The workflow in Figure 34 can be downloaded from CyberWater website <2. VIC4 Model Simulation>.


TimeRange
French Creek (SpaceRange)
PasswordDialog
NCALDASAgent
mm/s to mm/d (msmUnitConversion)
K to C (msmUnitConversion)
K to C (msmUnitConversion)
mm/s to mm/d (msmUnitConversion)
mm/s to mm/d (msmUnitConversion)
VicAgent
Surface Runoff (msmShowChart)
Baseflow (msmShowChart)

Name: NCALDASAgent
Type: NCALDASAgent
Package: edu.pitt.hydro.msm
Namespace: Open Data
Id: 3

username
String example_username
variableName
String Wind speed: [m/s]
String Total Precipitation Rate: [mm/s]
String Temperature Min: [K]
String Temperature Max: [K]
String Baseflow-groundwater runoff: [mm/s]
String Storm surface runoff: [mm/s]
String Storm surface runoff: [mm/s]
String Baseflow-groundwater runoff: [mm/s]
01_case_study
password
rename_variable

Figure 34. Workflow used to simulate and compare values of baseflow and surface runoff of the French Creek basin, using VIC.

- Results from the final workflow are shown in Figures 35 and 36.

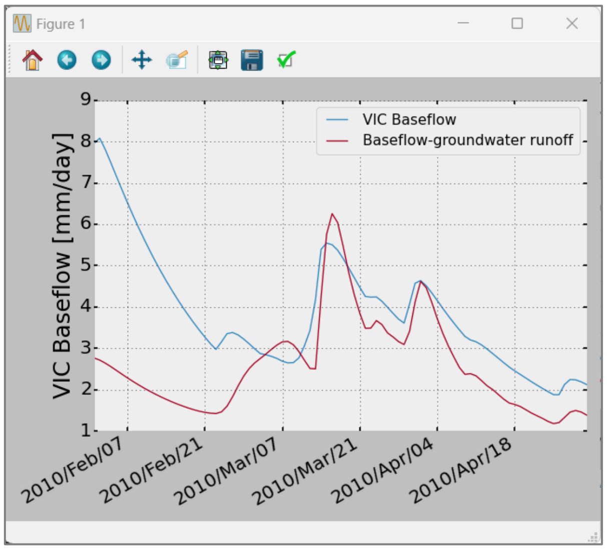


Figure 35. Resulting subsurface baseflow from the VIC simulation of the French Creek in PA.

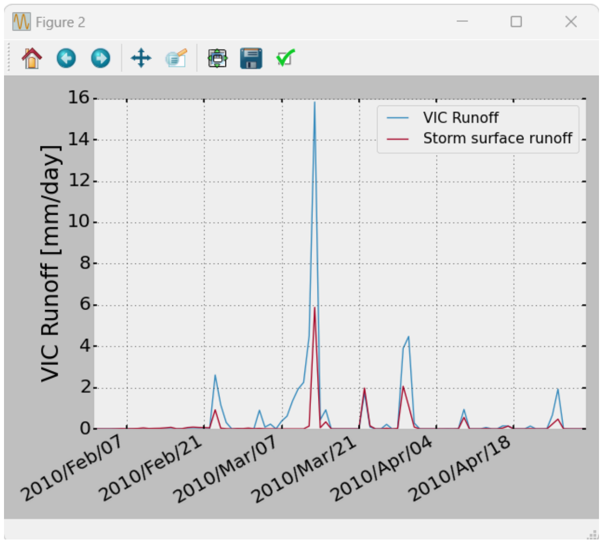


Figure 36. Resulting surface runoff from the VIC simulation of the French Creek in PA.

### 3.1.2 *Creating a simulation with the generated VIC model Agent*

CyberWater not only offers a model agent but also includes user-friendly generic model agent toolkits. These toolkits allow users to create their own model agents effortlessly, without any programming required. To further simplify the process of setting up workflows using these toolkits, CyberWater has developed a pre-defined workflow. This workflow can be seamlessly integrated into a new model agent.

Users can then conveniently select and drag-and-drop the resulting module—identified by a "_g" suffix—from the module panel. This can be incorporated into their workflow in the same manner as the VICAgent module, as illustrated in Section 3.1.1. This approach significantly streamlines the workflow setup process, making it more accessible and efficient for users. User can resume the workflow from what Figure 27 shows, and simply replace the VICAgent moduel with VIC4Agent_g module by following steps with the configuration, parameters and initial state files required by the VIC4 model for the Indiantown Run watershed. All the parameter values used in the following example are default values that can be modified in the future by the user to perform parameter calibrations. These data files are available at *Hydroshare* at:

<Examples Data\GT\VIC4>

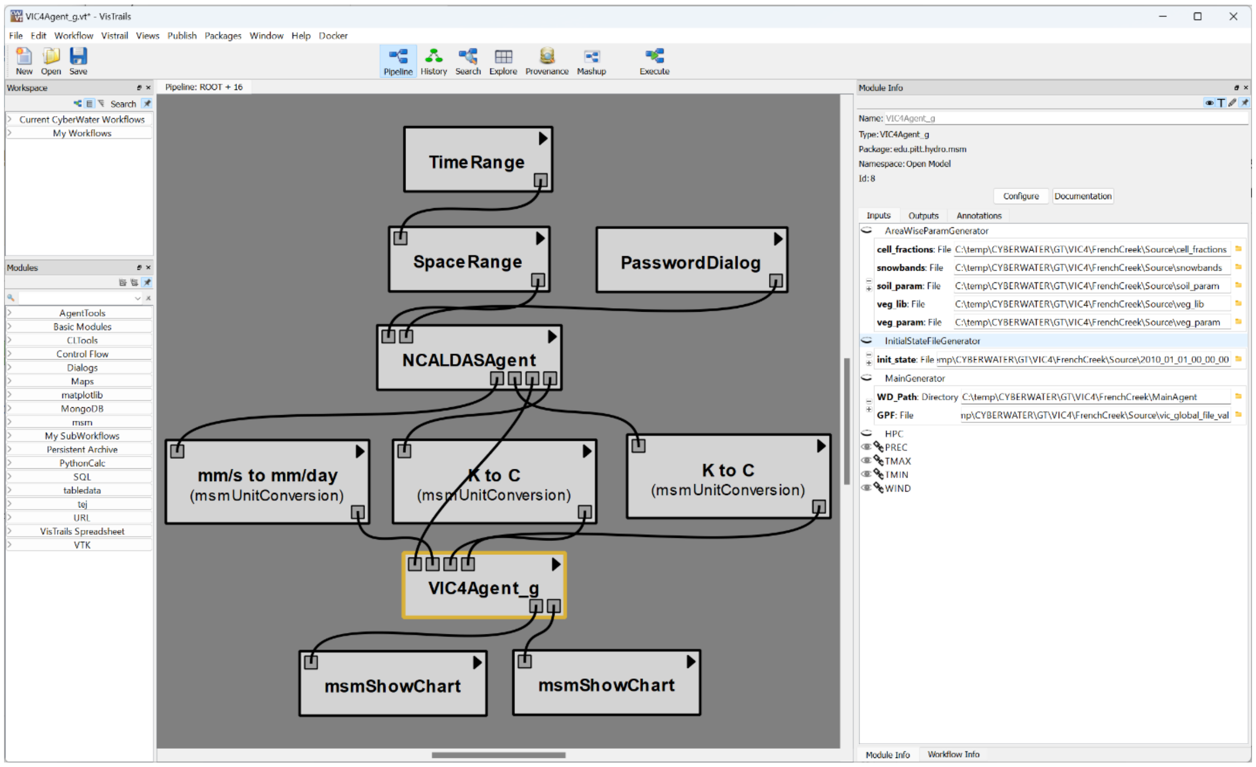
And this example assumes that you will place this folder at:

< C:\temp\CYBERWATER\GT>

- Add the *VIC4Agent_g*[1] module into the workflow, and configure the inputs as follows.
  - MainGenerator:
    - WD_Path: C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\MainAgent, Note: Create the new folder "MainAgent"
    - GPF: C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\vic_global_file_val
  - PREC: Connect the module that brings Precipitation in mm/day.
  - TMAX: Connect the module that brings Temperature Max in C.
  - TMIN: Connect the module that brings Temperature Min in C.
  - WIND: Connect the module that brings Wind Speed in m/s.
  - AreaWiseParamGenerator:
    - cell_fractions: C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\cell_fractions
    - snowbands: C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\snowbands
    - soil_param: C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\soil_param
    - veg_lib: C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\veg_lib
    - veg_param: C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\veg_param
  - InitialStateFileGenerator:
    - init_state: C:\temp\CYBERWATER\GT\ VIC4\FrenchCreek\Source\2010_01_01_00_00_00
  - HPC (Optional, if user doesn't have the HPC resource and want to execute the model on the local machine, please keep the text area blank):
    - platform: Bridges-shared (or another accessible platform in the drop-down list)
    - customized: Users' customized server address
    - username: Users' username to access the HPC platform.

The resulting workflow should look as shown in Figure 37. The workflow in Figure 37 can be downloaded from CyberWater website <12. VIC4 Model Simulation with Generated VIC4 Model Agent>.

[1] Located at msm>Open Model>VIC4Agent_g

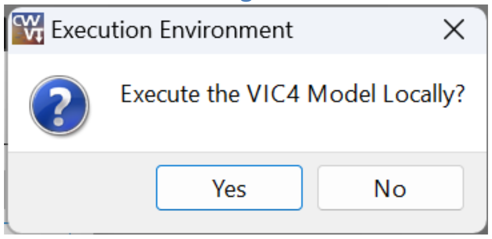


Name: VIC4Agent_g
Type: VIC4Agent_g
Package: edu.pitt.hydro.msm
Namespace: Open Model
Id: 8

AreaWiseParamGenerator
**cell_fractions**: File C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\cell_fractions
**snowbands**: File C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\snowbands
**soil_param**: File C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\soil_param
**veg_lib**: File C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\veg_lib
**veg_param**: File C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\Source\veg_param
InitialStateFileGenerator
**init_state**: File :mp\CYBERWATER\GT\VIC4\FrenchCreek\Source\2010_01_01_00_00_00
MainGenerator
**WD_Path**: Directory C:\temp\CYBERWATER\GT\VIC4\FrenchCreek\MainAgent
**GPF**: File np\CYBERWATER\GT\VIC4\FrenchCreek\Source\vic_global_file_val
HPC
PREC
TMAX
TMIN
WIND

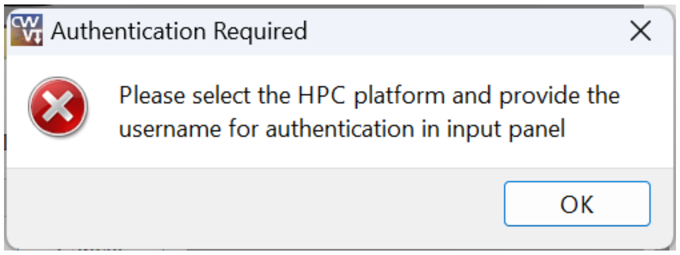
*Figure 37. Workflow used to simulate and compare values of baseflow and surface runoff of the French Creek basin, using generated VIC4Agent_g module.*

- To initiate the workflow, users simply click the "execute" button. During the processing of the VIC4Agent_g module, a prompt will appear, inquiring whether the user wishes to run the VIC4 model locally, as depicted in Figure 38. Choosing 'Yes' triggers the model to execute on the user's machine. On the other hand, selecting 'No' leads to a different scenario. If the user has not previously configured the High-Performance Computing (HPC) settings in the input panel, a warning message will emerge, as shown in Figure 39. This message reminds users to enter the necessary HPC information in the input panel. Once the HPC details are provided and the user opts to execute the model remotely upon re-executing the workflow, a password prompt will appear. This prompt requests the user's password for accessing the HPC platform, ensuring secure and authorized use of the resources in Figure 40.


Execution Environment

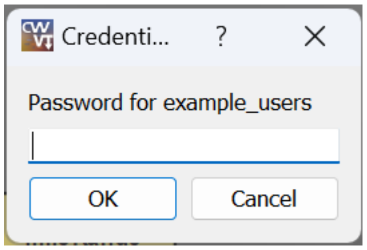
Execute the VIC4 Model Locally?

*Figure 38. The prompt window asks users whether to execute the VIC4 model locally.*


Authentication Required

Please select the HPC platform and provide the username for authentication in input panel

*Figure 39. The warning window to ask user to configure the HPC entrances in input panel.*


Credenti... ?

Password for example_users

*Figure 40. The credential prompt to ask user type in their password to access the HPC platform.*

### 3.1.3 *Adding Routing to a VIC-Agent workflow*

With the information of surface runoff and baseflow provided by *VICAgent*, the Routing Agent is able to perform a simple Muskingum routing scheme. This process computes the values of streamflow at the outlet of the basin that lies inside the spatial extent selected. Since this operation requires knowledge of the topography of the area, a digital elevation map (DEM) must be provided.

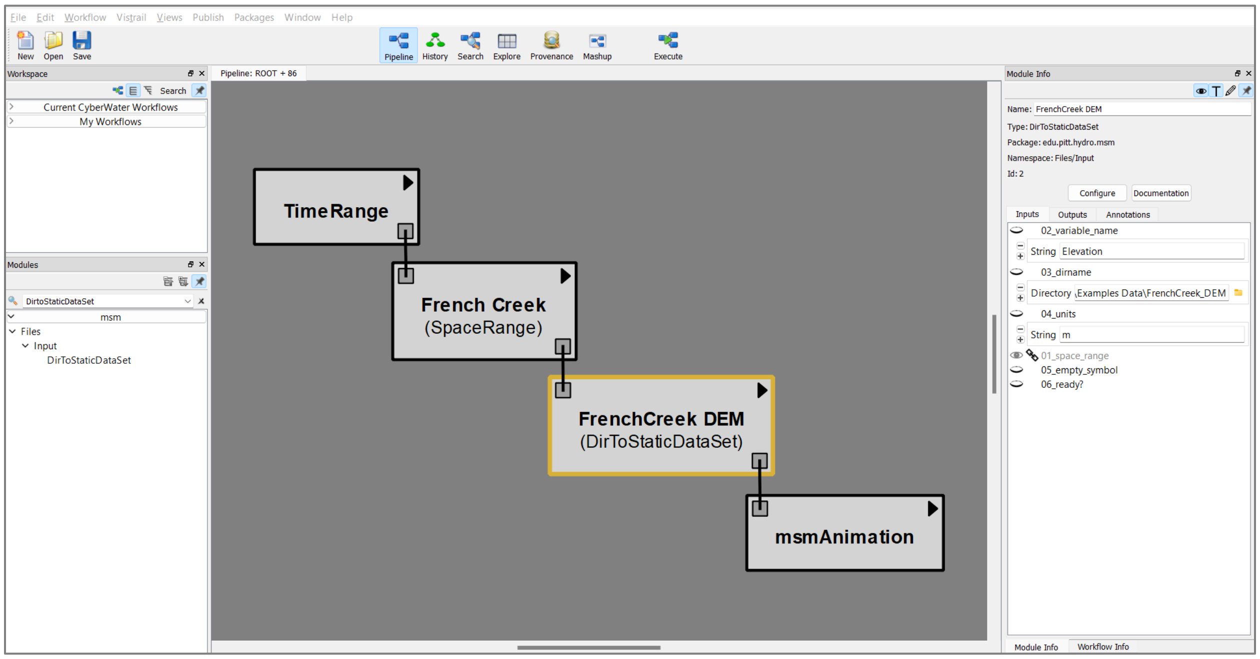
An already cropped DEM data of the French Creek Basin with a resolution of 30 meters is included on Hydroshare at <Examples Data\FrenchCreek_DEM>. The *DirToStaticDataSet*[1] module in *CyberWater* is capable of loading this map into *VisTrails*. In order to have a visual representation of this DEM, copy and paste the *SpaceRange-TimeRange* tuple created in the previous example (Figure 34) into a new workflow. Connect it to a new *DirToStaticDataSet* module. Its inputs should be:

- dirname: The full path to the **folder** where the DEM is located in your local machine
- variable_name: "Elevation"
- units: "m"

Next, the user can use the *msmAnimation*[2] plotting tool to create a visual representation of the DEM. Make sure to select the 'terrain_map' from the color_bar option. The resulting workflow should look as shown in Figure 41. The workflow in Figure 41 can be downloaded from CyberWater website <3. DEM Data Visualization>.

[1] Located at msm>File>Input>*DirToStaticDataSet*

[2] Located at msm>Visualization>*msmAnimation*



Name: FrenchCreek DEM
Type: DirToStaticDataSet
Package: edu.pitt.hydro.msm
Namespace: Files/Input
Id: 2

02_variable_name
String Elevation
03_dirname
Directory \Examples Data\FrenchCreek_DEM
04_units
String m
01_space_range
05_empty_symbol
06_ready?

*Figure 41. A workflow to create a display of a local DEM map*

And the plotted map is shown in Figure 42. This display allows the user to see a geo-located map, using a Latitude-Longitude projection.

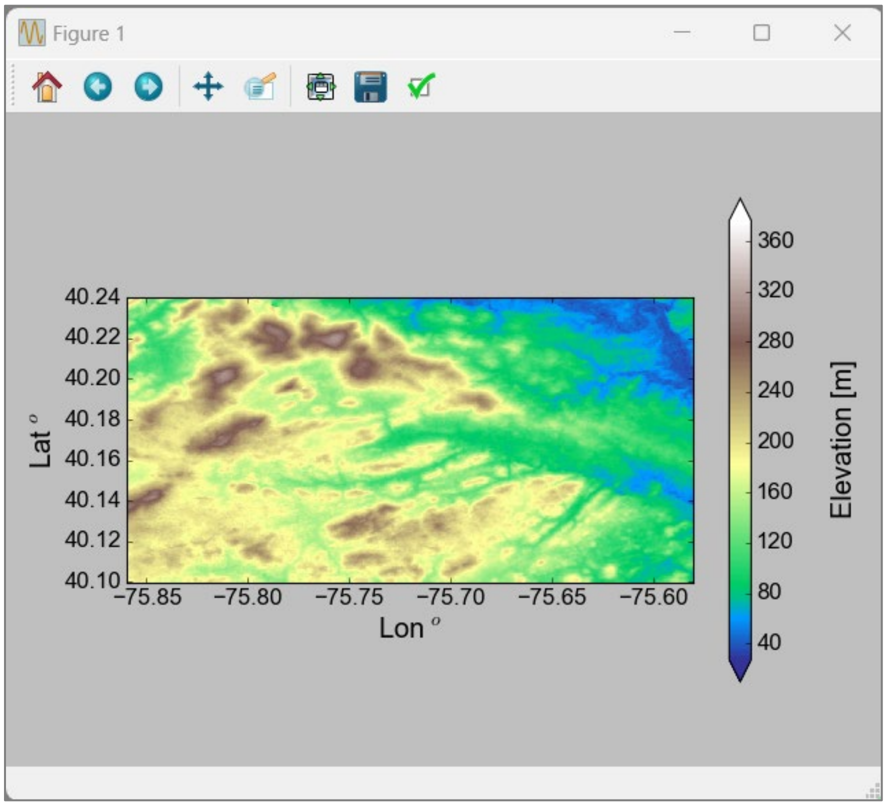


*Figure 42. DEM of the French Creek plotted using the CyberWater msmAnimation tool.*

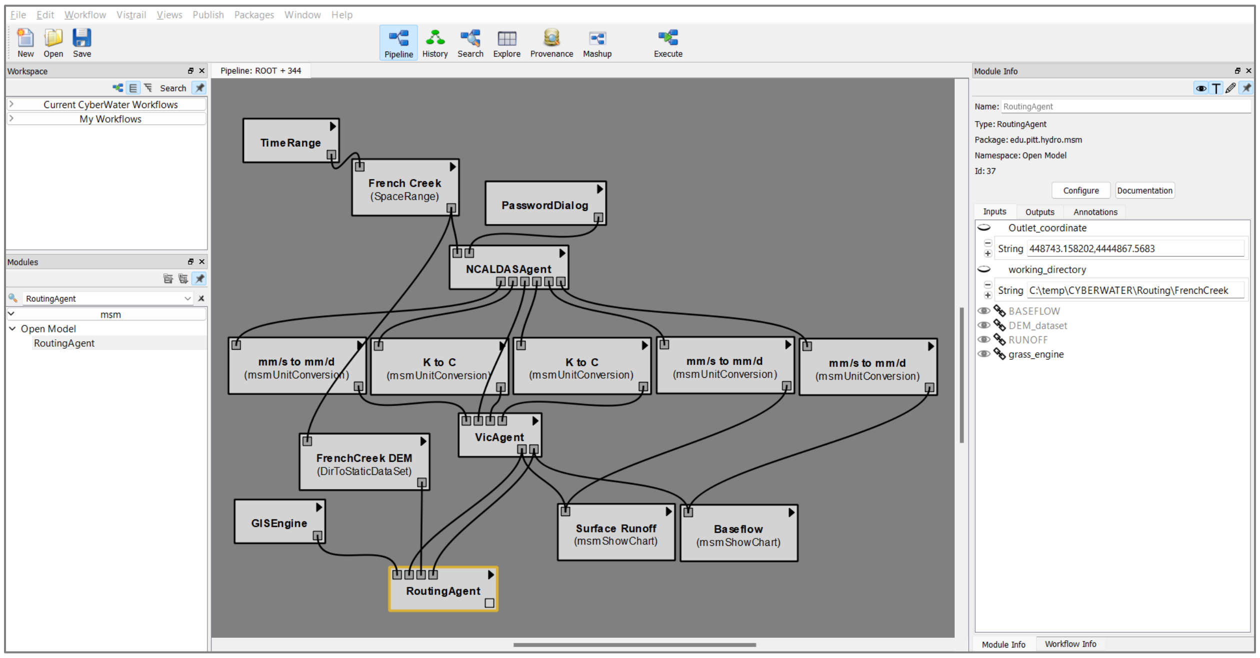
Using this DEM, the Routing Agent creates a drainage network given a set of output coordinates provided as input. Starting from the workflow displayed in Figure 34, the user can add the *DirToStaticDataSet* module used in Figure 41 to upload the DEM map. The input port of the *DirToStaticDataSet* module should be connected to the *SpaceRange* box as done before. The user will also need to add the *GISEngine*[1] module to create a GRASS engine object for the Routing Agent. CyberWater's *GISEngine* module was created to make a bridge between the main framework and GRASS GIS. Basically, the *GISEngine* allows users to execute all commands available in GRASS GIS directly inside the CyberWater's interface, so the user does not need to open and close third-party applications, nor spend any time dealing with computationally-taxing GIS interfaces. Next, bring a *RoutingAgent*[2] module, and provide the following inputs:

- BASEFLOW: The 'Baseflow' output of the *VICAgent*
- RUNOFF: The 'Runoff' output of the *VICAgent*
- DEM_dataset: The output port of the *DirToStaticDataSet* module.
- grass_engine: The 'GRASS_GISEngine' output of the *GISEngine*
- Outlet_coordinate: 448752.971,4444801.471 Note: The outlet coordinates must be provided in UTM coordinate system. Users can get approximate coordinates for the outlet from USGS database in latitude/longitude coordinate system and then convert them. Another method for conversion will be described in Section 3.1.3 using the *msmComputeExactOutput* module.
- working_directory: This is optional. If the user didn't specify one, the default folder is located at '*CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_model_agents\RoutingAgent\tmp*'. It is recommended that users add their own directories if the routing results will be used for any other purpose outside the CyberWater environment.

The resulting workflow should look as shown in Figure 43.

[1] Located at AgentTools>GenericGIS>GISEngine

[2] Located at msm>Open Model>RoutingAgent


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Search
Current CyberWater Workflows
My Workflows
Modules
RoutingAgent
msm
Open Model
RoutingAgent
Pipeline: ROOT + 344
TimeRange
French Creek (SpaceRange)
PasswordDialog
NCALDASAgent
mm/s to mm/d (msmUnitConversion)
K to C (msmUnitConversion)
K to C (msmUnitConversion)
mm/s to mm/d (msmUnitConversion)
mm/s to mm/d (msmUnitConversion)
VicAgent
FrenchCreek DEM (DirToStaticDataSet)
GISEngine
Surface Runoff (msmShowChart)
Baseflow (msmShowChart)
RoutingAgent
Module Info
Configure
Documentation
Inputs
Outputs
Annotations
Module Info
Workflow Info

Name: RoutingAgent
Type: RoutingAgent
Package: edu.pitt.hydro.msm
Namespace: Open Model
Id: 37

Outlet_coordinate
String 448743.158202,4444867.5683
working_directory
String C:\temp\CYBERWATER\Routing\FrenchCreek
BASEFLOW
DEM_dataset
RUNOFF
grass_engine

*Figure 43. A workflow where VICAgent is coupled with the Routing Agent.*

The *UsgsAgent* gets data from USGS hydrometric stations, making it a great tool to compare the results obtained by the *RoutingAgent* coupled with *VIC*. After including it into the current workflow, connect its input port to the SpaceRange box and set the '*unit_conversion_factor*' input to 0.028317. This unit conversion transforms the units of the streamflow from the default unit of ft$^3$/s into the unit of m$^3$/s. The final step is to add the *msmShowChart* module to compare between the the *RoutingAgent* results and the observed results from *UsgsAgent* on the same time-series chart. Connect both *RoutingAgent* and *UsgsAgent* to the input port of the *msmShowChart*. You can also modify the final plot by changing y-axis name, unit and legend location. This can be done by setting:

- variable_name: Streamflow
- units: m$^3$/s
- legend_location: upper center

The final workflow is shown in Figure 44. The workflow in Figure 44 can be downloaded from CyberWater website <4. VIC4 Coupled with Routing Model Simulation>.

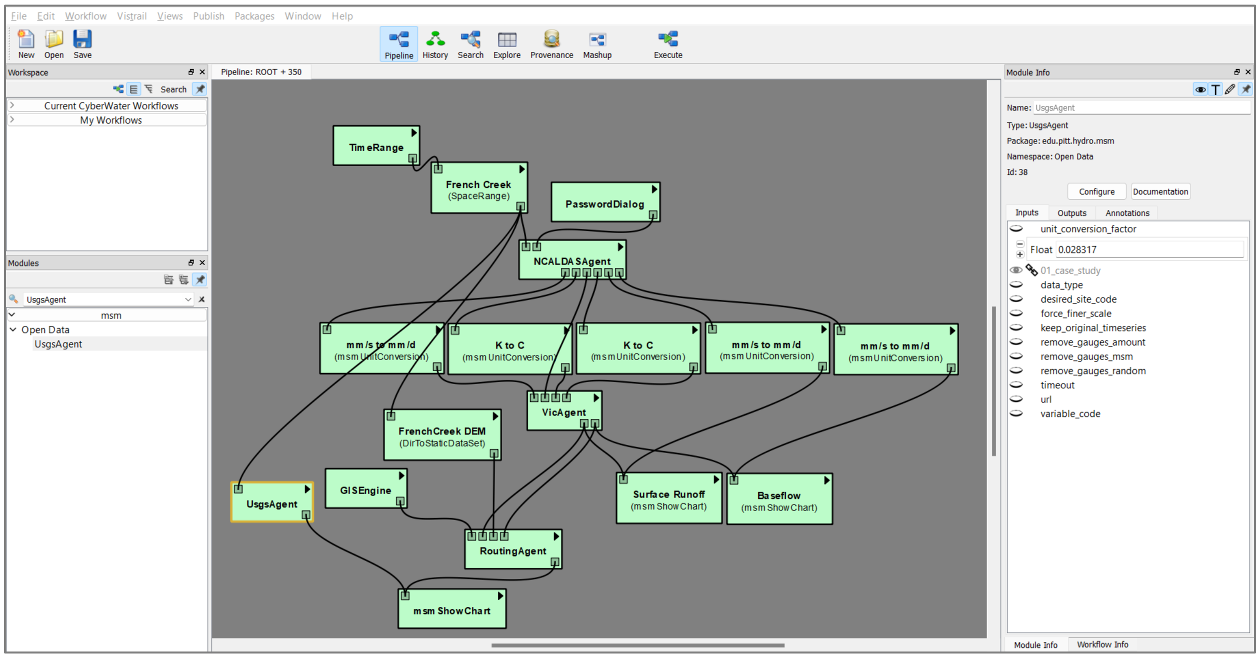


*Figure 44. A workflow where streamflow values from VICAgent coupled with Routing Agent in CyberWater can be compared with measurements from USGS.*

After executing this workflow, the user will see the chart displayed in Figure 45. Please note that none of the models previously tested have been calibrated. *RoutingAgent* assumes a default initial state where none of the channels contain water. For more information about the use of this module, please check its documentation.

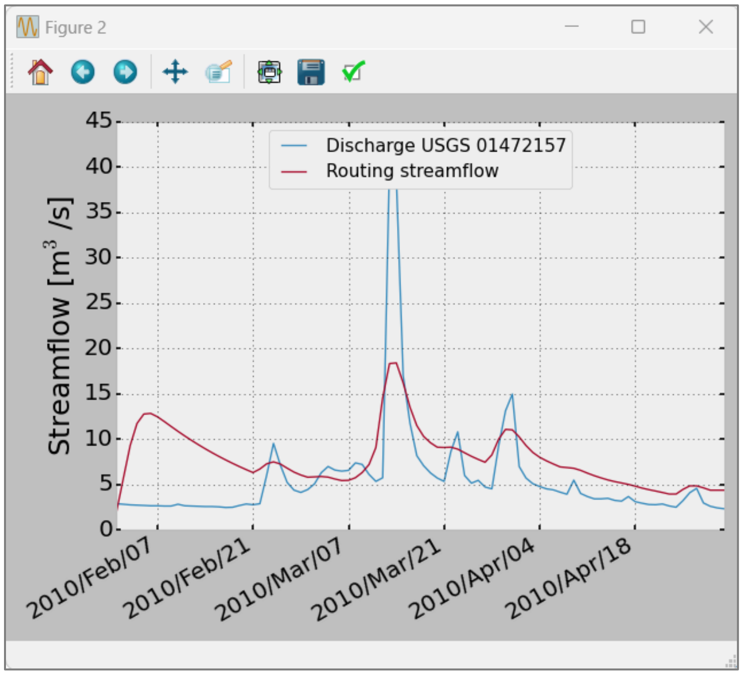


*Figure 45. Chart comparing streamflow values from VICAgent + Routing Agent in CyberWater with measurements from USGS.*

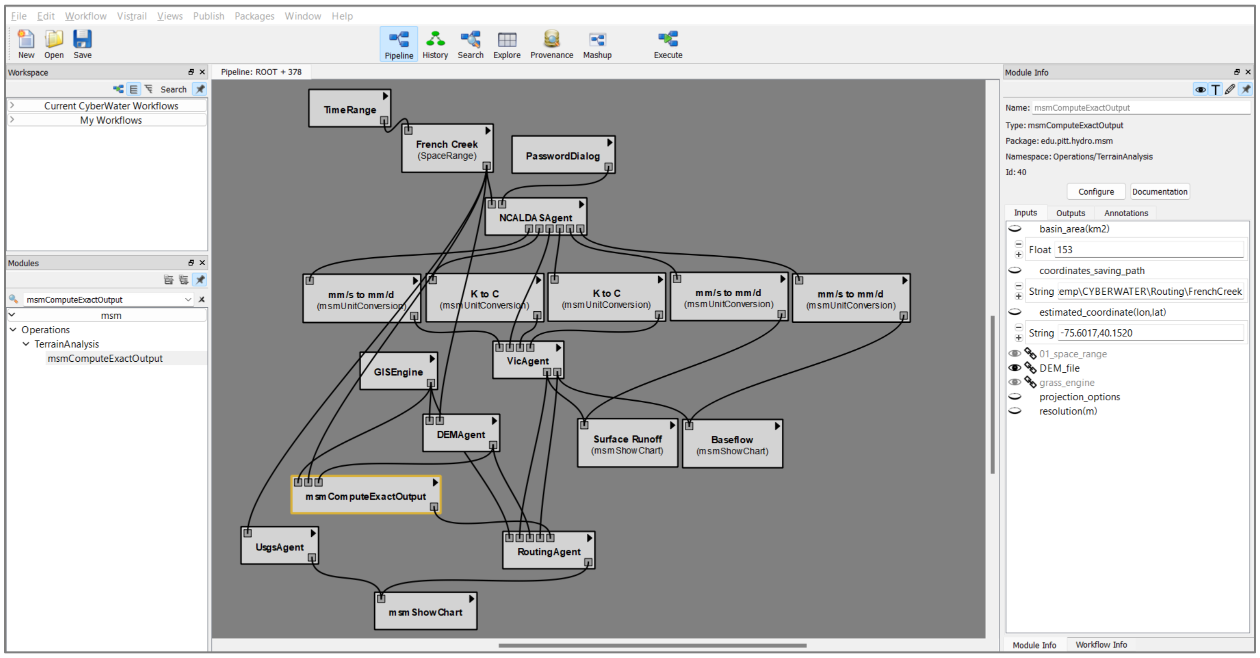
## 3.2 Using the DEMAgent to automate DEM downloading and processing

Using the *DEMAgent* module, DEM data can automatically be downloaded and loaded into CyberWater given any space range within the SRTM survey dataset. As an exercise, we will replace the *DirToStaticDataSet* module in the previous example with a *DEMAgent* module that automatically downloads the DEM data instead of using the DEM file provided.

Starting from the workflow displayed in Figure 44, remove the *DirToStaticDataSet* and then add the *DEMAgent* and *GISEngine* modules. Connect the *SpaceRange and GISEngine* boxes to their proper input ports of the *DEMAgent*. Also, provide the *DEMAgent* with your NASA Earth data credentials, the same ones used for the *NCALDASAgent*. You can type your password directly or activate the password icon and connect it to the *PasswrodDialog* box as we did before. The next step is to add the *msmComputExactOutput* module. This module is used to generate exact outlet coordinates from a guessed coordinates (lon,lat) of the watershed outlet and a guessed basin area (km$^2$). One way the user can get such information is from the USGS database (e.g., https://waterdata.usgs.gov/monitoring-location/01472157/#parameterCode=00065&period=P7D&showMedian=false). The options of this module for the French Creek watershed can be set as follows:

- Space_range: The output port of the *SpaceRange*
- DEM_file: The output port of the *DEMAgent* module
- grass_engine: The output of the *GISEngine*
- basin_area(km2): 153
- estimated_coordinate(lon,lat): -75.6017,40.1520
- coordinates_saving_path: Provide it with any folder path. The module will save a text file named "exact_outlet_coordinates" in this folder which includes the exact outlet coordinates of the watershed in (lat/lon) and UTM coordinate systems. This path must be provided by the user, otherwise the module will give an error.

The final step is to connect the *DEMAgent* output port to the proper input port of *RoutingAgent*. Then, delete the outlet coordinates that we provided before in the Outlet_coordinate option of the RoutingAgent and activate its input port instead to be connected to the *msmComputExactOutput.* The completed workflow is shown below in Figure 46.



Name: msmComputeExactOutput
Type: msmComputeExactOutput
Package: edu.pitt.hydro.msm
Namespace: Operations/TerrainAnalysis
Id: 40

basin_area(km2)
Float 153
coordinates_saving_path
String :emp\CYBERWATER\Routing\FrenchCreek
estimated_coordinate(lon,lat)
String -75.6017,40.1520
01_space_range
DEM_file
grass_engine
projection_options
resolution(m)

*Figure 46. The completed workflow with the DEMAgent.*

Run the workflow. You will see that the *DEMAgent* both downloads and crops the required DEM data so that the user need not change anything to run the VIC model with a coupled routing scheme compared with USGS data. The routing streamflow plot will be the same as that shown in Figure 45. This is just one more example of the ways in which *CyberWater* can make a workflow so much easier! Try changing the *SpaceRange* or *TimeRange*; the workflow will now be able to fully respond to any changes and download the correct data to execute the model and visualize the data. The workflow in Figure 46 be downloaded from CyberWater website <5. VIC4 and Routing Models Coupling with DEM Data>.

## 4. Generic Model Agent to execute a VIC (v5.0) – Routing simulation

There are two approaches to integrating a new model into the *CyberWater* system: write a model agent, or use the generic model agent tools provided in *CyberWater* to build the model agent without programming. We have used the specific models' agents that are coded beforehand, e.g., VicAgent and RoutingAgent, in the previous section in this manual. Other models like DHSVM are also already available natively in *CyberWater*. We show now, in this section, how to use the generic model agent tools to construct/configure these model agents. The important benefit here is that the user can integrate their own model into the *CyberWater* environment using these generic tools in the future with no or little coding effort. Therefore, users are expected to be highly knowledgeable about how to run their own models if they want to integrate them into *CyberWater*. First, a basic description of the generic model agent toolkit is provided. Then, an example with the use of the VIC (v5.0) model alone is provided first through a large size of a new study area, followed by one with the use of VIC (v5.0) model that is coupled with the Routing model.

## 4.1 Why integrate a model

*CyberWater* offers several features that make integrating a model appealing, especially useful for big models that require large amounts of heterogenous data. *CyberWater* allows this data to be downloaded and prepared at runtime, making models much more portable and able to be tested under different times and regions very easily. *CyberWater* also makes the coupling and combining of models easily, with the software handling the intermediate data preparation and transfer. It also offers a number of useful visualization tools, and makes comparing simulated and measured data easy by being able to download measured data at runtime according to the established case study.

## 4.2 Toolkit Overview

There are five import modules to keep in mind when integrating a model into *CyberWater*. The *MainGenerator* module is the module that receives the forcing data and establishes both the working directory and the global parameter file. The *AreaWiseParamGenerator* module is responsible for organizing the parameter files necessary for execution. The *ForcingDataFileGenerator* module takes the forcing information from the *MainGenerator* and puts it into the folder and file structure required by the model for its forcing data. The *InitialStateFileGenerator* imports any necessary state files into the working directory and organizes them as needed. Finally, the *RunModuleAgent* executes the model and interprets its results. All five modules can be found in *AgentTools>GenericModelAgent*. For more information about these modules and their design principles behind, please read Chen et al. (2022a).

## 4.3 Runtime preparation

It is assumed that any user integrating a model into *CyberWater* has ample knowledge of their own model, what it requires, and what it outputs. As such, it is assumed that the user knows exactly what the model expects for inputs. *CyberWater* allows for the structure of all the parameter files, forcing files, and initial state files to be defined, respectively, through the *AreaWiseParamGenerator, ForcingDataFileGenerator*, and *InitalStateFileGenerator*. Each of these modules include a “Ready” flag as an output to allow the user to properly time them before model execution. Folder and file names and content can be adjusted to match the expected input of the model. For obvious reasons, it is very important that these settings are adjusted to be accurate for the model being integrated.

## 4.4 Simulation output

The *RunModuleAgent* handles both the proper execution of the model and the processing of results. The user must know how the output of the model is structured to get a useful output in *CyberWater*. The *RunModuleAgent* is able to interpret both single file saved as a time series and outputs that are distributed over many files. The user is able to add outputs by clicking the “+” button and making the associated outputs visible. It is especially important to note that most counting of file rows and columns start at zero. For example, the 5th column has an index of 4.

## 4.5 Data Preparation for VIC5.0 (version 5.0)

It is important to mention that the concept and procedures of the data preparation described here are similar to those described in Section 3.1 for the VIC model. The main differences are in the study areas and in the input requirements associated with different models, i.e., VIC4.0 in Section 3.1 versus VIC5.0. Essential data sets for operating VIC5.0 can be categorized as follows:

- Forcing data in gridded format
- Static parameter files
- Results are either gridded (divided into files per cell) or in a single time-series output.

Users can pre-download the forcing data map files used by the Data Agents to their local machines to reduce the time it takes by directly retrieving such information from *CyberWater* in the run time. These files are available from HydroShare at:

<Examples Data\Forcing Data\NLDAS>

When you download the "NLDAS" folder to your local machine, open it, then copy all the folders that are found inside it and paste them inside the "downloads" folder on your local machine at:

<C:\CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_data_agents\NLDASAgent\downloads>

The user also has the choice of not downloading these files but obtaining them through the execution of the workflow instead. If taking the latter choice, the data downloading process for the following example might take around one hour due to the slow network. The VIC model simulation time takes a few minutes to finish in a typical personal computer (with 8GB memory) for this example with a large drainage area.

### 4.5.1 *Generic Model Agents used to execute a VIC simulation*

For this example, an hourly simulation of a large watershed inside the state of Pennsylvania will be constructed. The watershed is the West-Branch Susquehanna (WBS) basin[1] which covers more than 17,700 square kilometers (6,847 mi$^2$). This river basin is selected to illustrate that *CyberWater* can scale up to handle very large watersheds. In this case, VIC model will be executed in both water-balance mode and energy-balance mode using VIC version 5.0. This watershed includes 299 model simulation cells, making its VIC parameter preparation a very tedious and time-consuming task if done manually. Similar to the example case with the French Creek watershed, all the parameter values used in this simulation are default values that can be modified in the future by the user to perform parameter calibrations.

- Add a *TimeRange* box for a simulation between 2010/01/01 00:00:00 (timeini) and 2010/03/01 00:00:00 (timeend). Since this simulation is hourly, it requires more than 1,400 data files per forcing variable. It would take around one hour to download all the forcing variables, depending on the internet connection and speed. Remember, once the data is downloaded to the user's machine, it does not need to be re-downloaded again unless the files are deleted[2].
- Add a *SpaceRange* box with the limits: -76.213, -78.9155, 41.933, and 40.454 (x_max, x_min, y_max, y_min respectively). Connect the '*timerange*' output of the *TimeRange* box with the '*subrange*' input of the *SpaceRange* box, as shown in Figure 25.
- Add *NLDASAgent* boxe, and fill it with your NASA Earth data credentials. Again, the user can add a *PasswordDialog* module to collect credentials at runtime. Use the *NLDASAgent* to bring the following variables: Temperature [K], Pressure [Pa], Radiation Flux Long Wave [W/m^2], Radiation

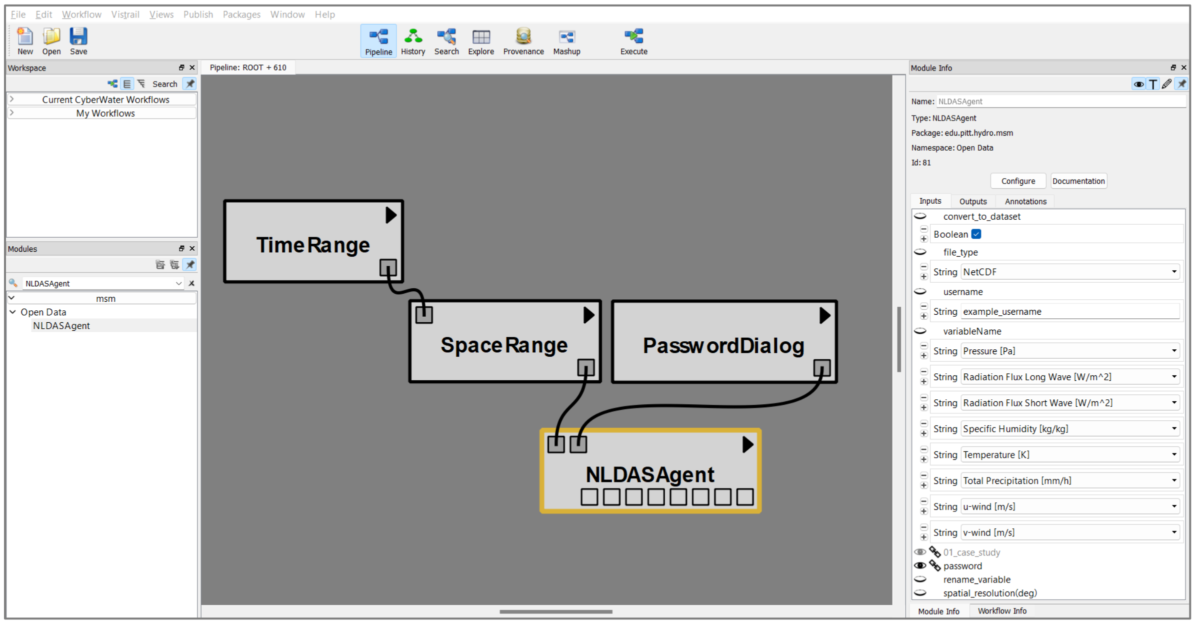
[1] USGS information used: https://waterdata.usgs.gov/pa/nwis/uv?site_no=01553500

[2] The user can access these files at < C:\CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_data_agents>. Then, select the Agent you want to check. The data will be in the "downloads" folder.

Flux Short Wave [W/m^2], Total Precipitation [mm/h], Specific Humidity [kg/kg], u-wind [m/s], and v-wind [m/s] by setting the "Variable_Name" to the forcing data you want and then clicking the "+" icon to add them one by one. Mark the "Boolean" choice in "convert_to_dataset" option. Set the "file_type" to "NETCDF". Finally, activate the output port corresponding to the 8 forcing variables from the "Outputs" panel. The resulting workflow is shown in Figure 47.

- Since this simulation is hourly and the time range is two months, it requires more than 1,400 data files per forcing variable to be downloaded. It would take around one hour to download all the forcing variables, depending on the internet connection and speed. Remember, once the data is downloaded to the user's machine, it does not need to be re-downloaded again unless the files are deleted. To save time, these data can be downloaded directly from *Hydroshare*. Users can pre-download the map files used by the Data Agents to their local machines to reduce the time it takes by directly retrieving such information from *CyberWater* in the run time. These files are compressed and available on HydroShare at the folder:
  <Examples Data/Forcing Data/NLDAS/-76.655_40.425_-76.585_40.475.zip>
  When you download the zip file to your local machine, open it, then extract all files in the zip file and put them inside the "downloads" folder found at:

  <C:\CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_data_agents\NLDASAgent\downloads>



Name: NLDASAgent
Type: NLDASAgent
Package: edu.pitt.hydro.msm
Namespace: Open Data
Id: 81

convert_to_dataset
Boolean ✓
file_type
String NetCDF
username
String example_username
variableName
String Pressure [Pa]
String Radiation Flux Long Wave [W/m^2]
String Radiation Flux Short Wave [W/m^2]
String Specific Humidity [kg/kg]
String Temperature [K]
String Total Precipitation [mm/h]
String u-wind [m/s]
String v-wind [m/s]
01_case_study
password
rename_variable
spatial_resolution(deg)

*Figure 47. Workflow with the CyberWater NLDASAgent bringing 8 different variables to execute the VIC model in energy-balance mode.*

- Add the *msmDatasetOperation* to compute the magnitude of the Wind Speed vector, summing its two components. For this, set the inputs as:
  - operation: (x**2+y**2)**0.5
  - units: m/s
  - variable name: Wind Speed

  - dataset_name1: Connect the *NLDAS* module with *U-Wind*.
  - dataset_name2: Connect the *NLDAS* module with *V-Wind*.
- Use the *msmUnitConversion* to transform the units of the Temperature and Pressure datasets. For the Pressure conversion you can use:
  - new units: kPa
  - operation: x/1000
- And for the Temperature conversion:
  - new units: C
  - operation: x-273.15
- The energy balance mode of VIC requires the user to provide with Water Vapor Pressure as a forcing variable. For this, the below equation can be used, which approximates the vapor pressure as a function of specific humidity and atmospheric pressure:

$$VP = \frac{SH * P}{0.622}$$

  Where:
    - VP: Water vapor pressure (kPa)
    - SH: Specific Humidity (dimensionless)
    - P: Pressure (kPa)
- This operation can be performed using the *msmDatasetOperation* module. The inputs should be set as follows:
  - operation: (x*y)/0.622
  - units: kPa
  - variable_name: Vapor Pressure
  - dataset_name1: Connect the *NLDAS* module with *Specific Humidity*.
  - dataset_name2: Connect the *msmUnitConversion* module of *Pressure in kPa*.

## 4.6 How to Construct Model Agents

This section introduces the use of the Generic Model Agent Tools that are designed to allow users to build the model agents to run simulations of their own models in *CyberWater* in the future. In this section, we use VIC5.0 as an example to illustrate the procedures of constructing the model agents with the Generic Model Agent Tools to achieve similar objectives as the functions of the *VicAgent* and *RoutingAgent* used in the examples of Section 3 in this manual.

### 4.6.1 *Construct model agents for VIC5.0 using generic model agent tools*

First of all, download the global, parameters and initial states files used for this example. Knowing that the pre-prepared parameter files used in this example are default and can be modified in the future by the user to perform parameter calibrations. These data files are available on Hydroshare at:
<Examples Data\GT>
This example assumes that you will place the folder "GT" inside the folder "CYBERWATER" found at: <C:\temp\CYBERWATER> so that the final directory becomes <C:\temp\CYBERWATER\GT>
Note: Check if the folders "temp" and "CYBERWATER" are already found, otherwise create them.

### 1. *MainGenerator*

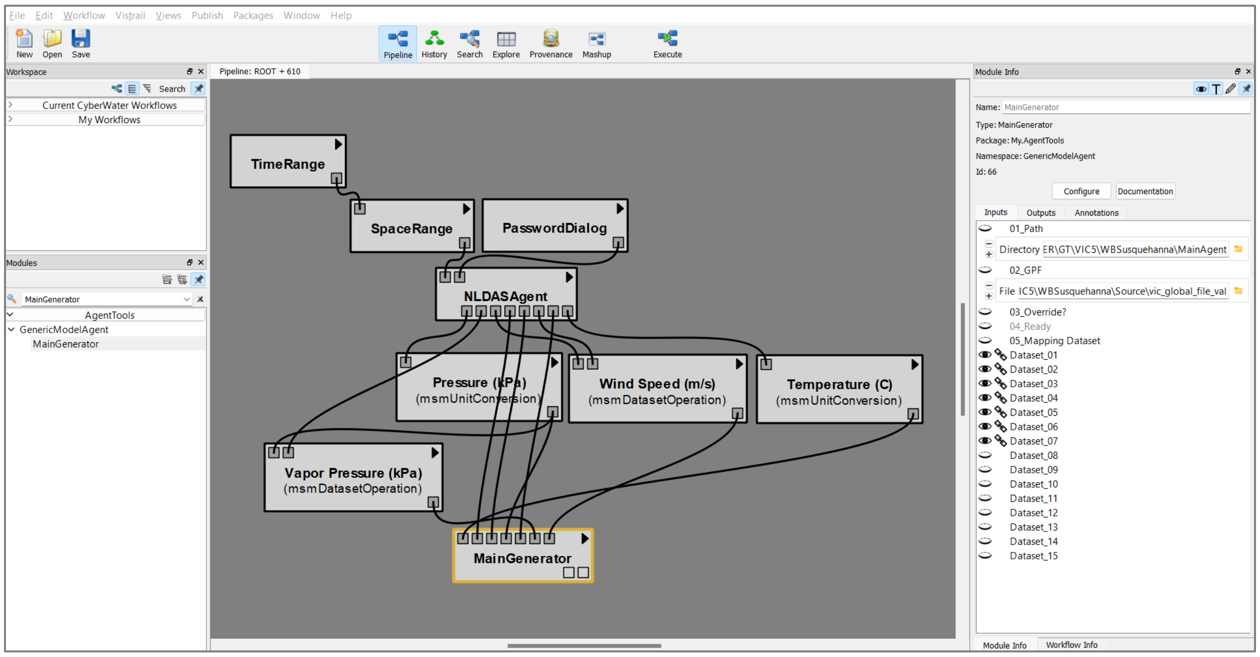
With the data preparation from the previous section, we are now ready to start constructing the VIC model agent from the Generic Model Agent tools. First, bring a *MainGenerator*[1] module, the main control component of the Generic Model Agent tools. This component is responsible for setting up the folder where the simulation will be performed. It receives all the forcing datasets as inputs. **The order at which they are added matters since the final forcing files will display the data in the same order: first Dataset_01, then Dataset_02, and so on.** The inputs should be set up as follows:

- 01_Path: C:\temp\CYBERWATER\GT\VIC5\WBSusquehanna\MainAgent
  Note: Create the new folder "MainAgent"
- 02_GPF: C:\temp\CYBERWATER\GT\VIC5\WBSusquehanna\Source\vic_global_file_val
- Dataset_01: Connect the module that brings Temperature in C.
- Dataset_02: Connect the module that brings Long Wave Radiation in W/m$^2$.
- Dataset_03: Connect the module that brings Precipitation in mm/h.
- Dataset_04: Connect the module that brings Pressure in kPa.
- Dataset_05: Connect the module that brings Short Wave Radiation in W/m$^2$.
- Dataset_06: Connect the module that brings Water Vapor Pressure in kPa.
- Dataset_07: Connect the module that brings Wind Speed in m/s.

The users should activate Dataset_01, Dataset_02, Dataset_03, Dataset_04, Dataset_05, Dataset_06, and Dataset_07 if they are not activated. The resulting workflow is shown in Figure 48.

*VisTrails* offers a nice feature that allows the user to group multiple modules in order to facilitate display. Select all the modules between *SpaceRange* and *MainGenerator* and go to the Workflow>Group menu on the top toolbar. This is shown in Figure 49.

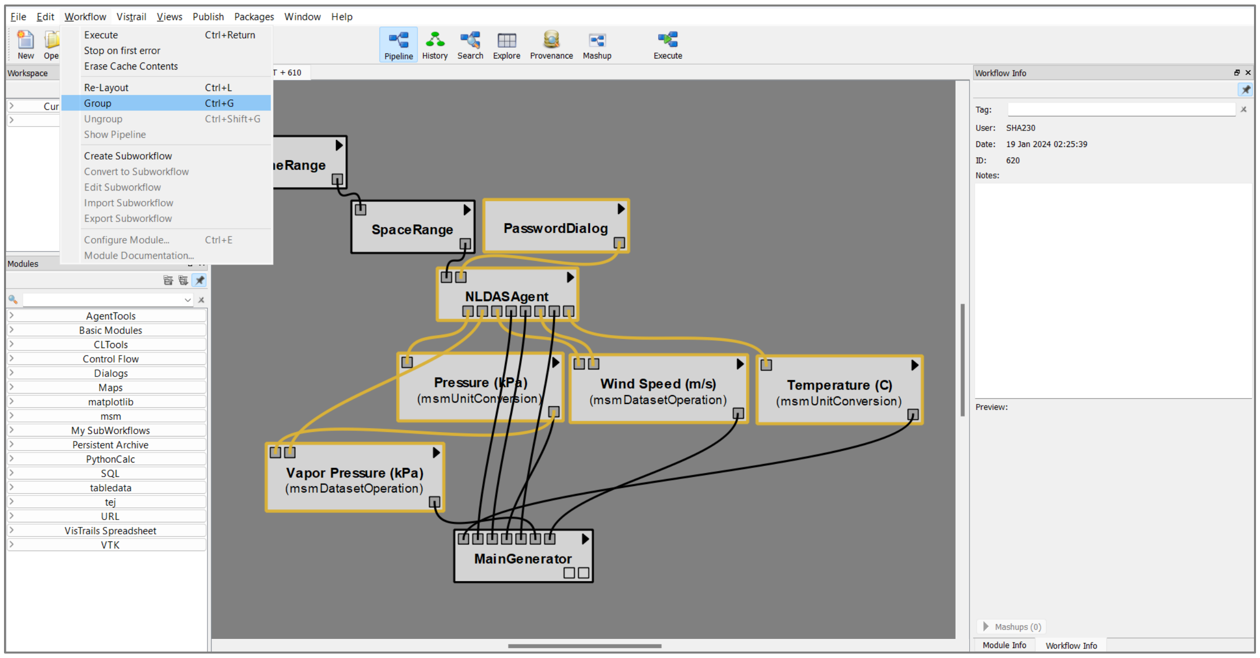
[1] Located at AgentTools>GenericModelAgent>MainGenerator


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Search
Current CyberWater Workflows
My Workflows
Modules
MainGenerator
AgentTools
GenericModelAgent
MainGenerator
Pipeline: ROOT + 610
TimeRange
SpaceRange
PasswordDialog
NLDASAgent
Pressure (kPa) (msmUnitConversion)
Wind Speed (m/s) (msmDatasetOperation)
Temperature (C) (msmUnitConversion)
Vapor Pressure (kPa) (msmDatasetOperation)
MainGenerator
Module Info
Configure
Documentation
Inputs Outputs Annotations
Module Info Workflow Info

Name: MainGenerator
Type: MainGenerator
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 66

01_Path
Directory ER\GT\VIC5\WBSusquehanna\MainAgent
02_GPF
File IC5\WBSusquehanna\Source\vic_global_file_val
03_Override?
04_Ready
05_Mapping Dataset
Dataset_01
Dataset_02
Dataset_03
Dataset_04
Dataset_05
Dataset_06
Dataset_07
Dataset_08
Dataset_09
Dataset_10
Dataset_11
Dataset_12
Dataset_13
Dataset_14
Dataset_15

*Figure 48. MainGenerator with the inputs required to run the VIC model on energy-balance mode.*



Tag:
User: SHA230
Date: 19 Jan 2024 02:25:39
ID: 620
Notes:

*Figure 49. VisTrails grouping feature used to group the forcing portion of the VIC simulation with the Generic Model Agent.*

The resulting workflow is a cleaner version of the previous one (see Figure 50). The grouped module named ‘Group’ can be renamed. The user can ungroup it by selecting it and going to the Workflow>Ungroup menu on the top toolbar.

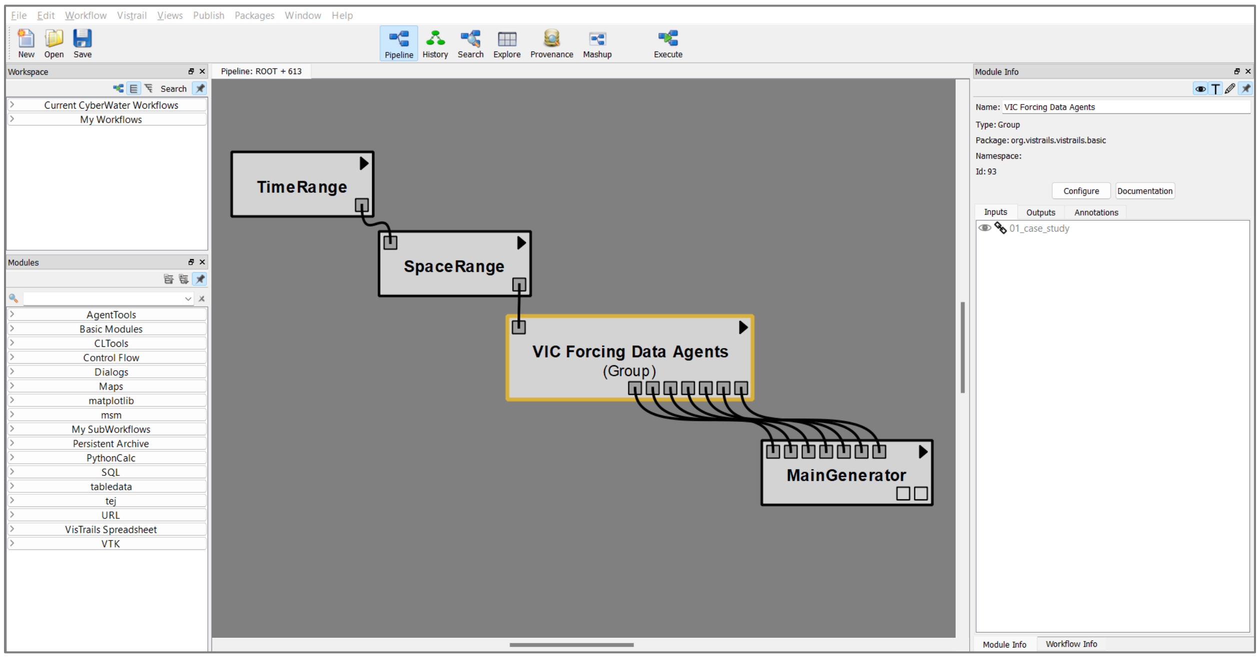


Figure 50. Workflow with the forcing portion grouped.

2. ***AreaWiseParamGenerator***
Add an *AreaWiseParamGenerator*[1], which is in charge of setting up the parameter files (e.g., soil parameter files and vegetation parameter files) of the model. These files need to be previously created by the user. The user is required to add the paths of each of the required parameter files as inputs through the module information panel on the right of the VisTrails window in the *AreaWiseParamGenerator* module, as described below:
   - WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
   - Parameter_Folder_Name: params.
   - File_In_00: C:\temp\CYBERWATER\GT\VIC5\WBSusquehanna\Source\cell_fractions
   - File_In_01: C:\temp\CYBERWATER\GT\VIC5\WBSusquehanna\Source\snowbands
   - File_In_02: C:\temp\CYBERWATER\GT\VIC5\WBSusquehanna\Source\soil_param
   - File_In_03: C:\temp\CYBERWATER\GT\VIC5\WBSusquehanna\Source\veg_lib
   - File_In_04: C:\temp\CYBERWATER\GT\VIC5\WBSusquehanna\Source\veg_param
3. ***ForcingDataFileGenerator***
Add a *ForcingDataFileGenerator*[2], responsible for the creation of the forcing data of the model. This component takes the information plugged in the *MainGenerator* and saves it into the folder that the user specifies. The inputs should be set up as follows:
   - WD_Path: Connect the 'WD_Path' output port from the *MainGenerator* module.
   - DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.
   - Forcing_Folder_Name: forcing

4. ***InitialStateFileGenerator***

[1] Located at AgentTools>GenericModelAgent>AreaWiseParamGenerator
[2] Located at AgentTools>GenericModelAgent>ForcingDataFileGenerator

Add an *InitialStateFileGenerator.* This module is responsible for placing the initial state files in the right folder structure on the working directory. Set the inputs as follows:

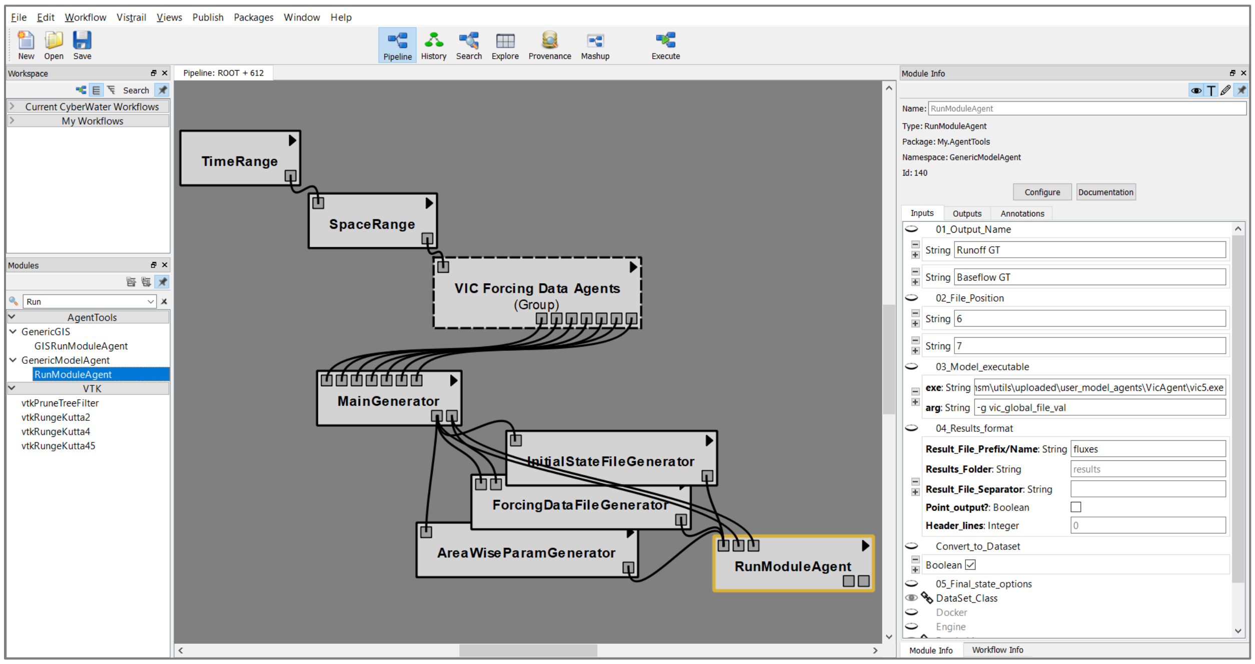
- WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
- File_In_0: C:\temp\CYBERWATER\GT\VIC5\WBSusquehanna\Source\state_20100101_00000

**5.** ***RunModuleAgent***

Add the final component of the Generic Model Agents, the *RunModuleAgent*. It is responsible for setting up the path where the executable file of the model (compiled, python, and java executables having been tested) should be located by *CyberWater*. Also, the arguments of the execution, as well as the outputs expected, and their format is configured here. Please note that the 'exe' path should be adjusted according to the location of the *VisTrails* in the local machine. Also, the path of the executable file should not include any spaces. For this example, please use the following inputs:

- WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
- DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.
- Ready_List: Connect the output of the three previous components into this port: *ForcingDataFileGenerator*, *AreaWiseParamGenerator*, and *InitialStateFileGenerator*.
- 01_Output_Name: Click the "+" button twice and set the top string as "Runoff GT" and the bottom string as "Baseflow GT"
- 02_File_Position: Click the "+" button twice and set the top string as "6" and the bottom string as "7"
- 03_Model_executable: Set the exe string as "C:\CyberWater\Vistrails\vistrails\packages\msm\utils\uploaded\user_model_agents\VicAgent\vic5.exe" and the arg string as "-g vic_global_file_val"
  Note: The "vic5.exe" is the executable of VIC5 model on Windows environment.
- 04_Results_format: Set the Result_File_Prefix/Name string as "fluxes"
- Activate "Boolean" option in "Convert_to_Dataset"
- Activate output ports corresponding to the number of outputs needed (e.g., if Runoff and Baseflow are the only variables wanted to be outputted as shown in this example, users only need to activate two output ports)

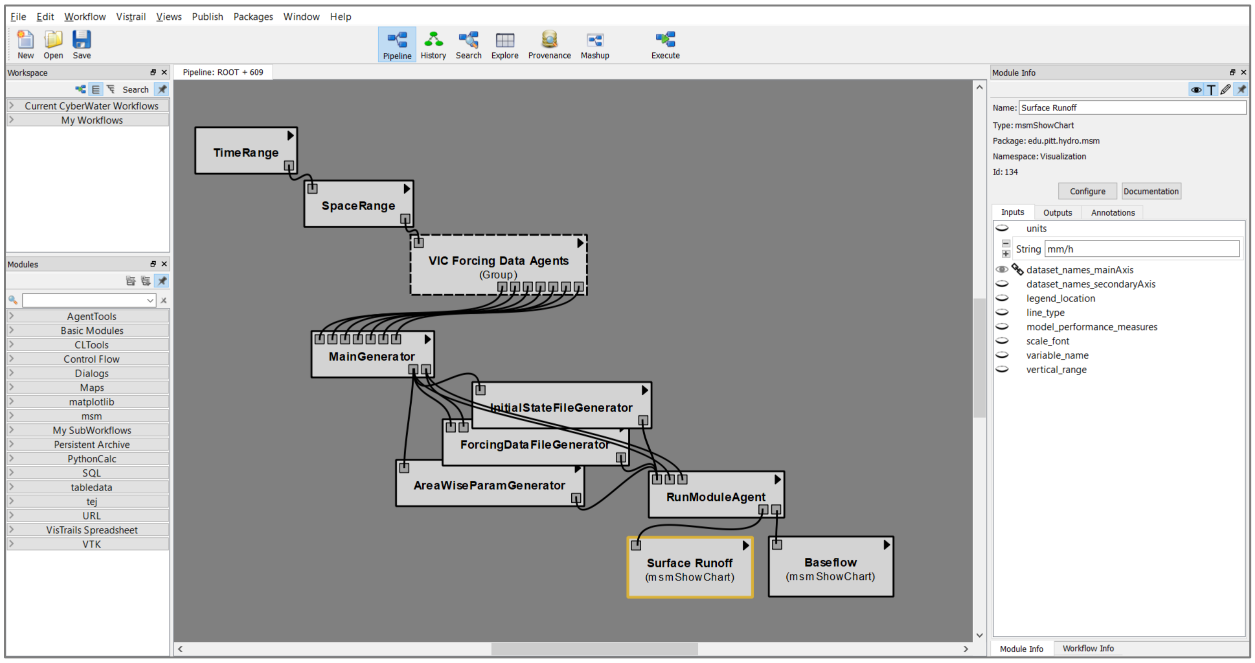
The resulting workflow is shown in Figure 51. Like before, now the user can add the *msmShowChart* modules to plot the surface runoff and the baseflow for viewing. After adding two *msmShowChart* modules to plot the Output01 (Surface Runoff) and Output02 (Baseflow) ports of the *RunModuleAgent*, the user will be able to execute the workflow shown in Figure 52. The charts obtained are depicted in Figure 53 and Figure 54. The workflow in Figure 52 can be downloaded from CyberWater website <6. VIC5 Model Simulation>.



Name: RunModuleAgent
Type: RunModuleAgent
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 140

01_Output_Name
String Runoff GT
String Baseflow GT
02_File_Position
String 6
String 7
03_Model_executable
**exe**: String ısm\utils\uploaded\user_model_agents\VicAgent\vic5.exe
**arg**: String -g vic_global_file_val
04_Results_format
**Result_File_Prefix/Name**: String fluxes
**Results_Folder**: String results
**Result_File_Separator**: String
**Point_output?**: Boolean ☐
**Header_lines**: Integer 0
Convert_to_Dataset
Boolean ☑
05_Final_state_options
DataSet_Class
Docker
Engine

*Figure 51. Workflow with the full set of Generic Model Agent tools to perform a VIC execution.*



Name: Surface Runoff
Type: msmShowChart
Package: edu.pitt.hydro.msm
Namespace: Visualization
Id: 134

units
String mm/h
dataset_names_mainAxis
dataset_names_secondaryAxis
legend_location
line_type
model_performance_measures
scale_font
variable_name
vertical_range

*Figure 52. Workflow where the fluxes of the WBS basin are simulated using the Generic Model Agents executing the VIC model.*

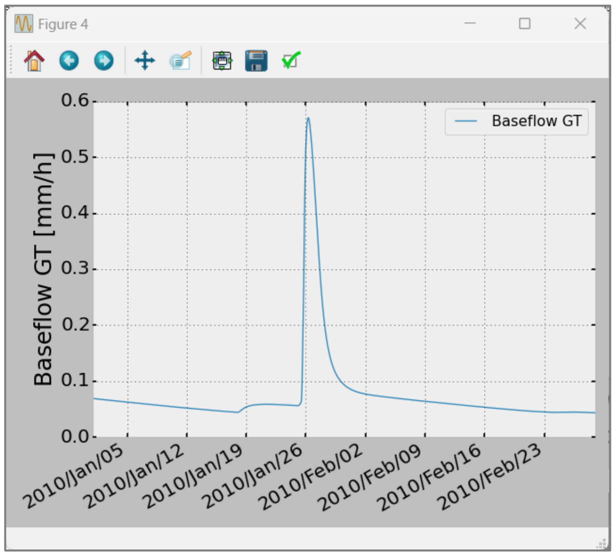


Figure 53. Baseflow of the WBS basin obtained using the Generic Model Agents executing the VIC model.

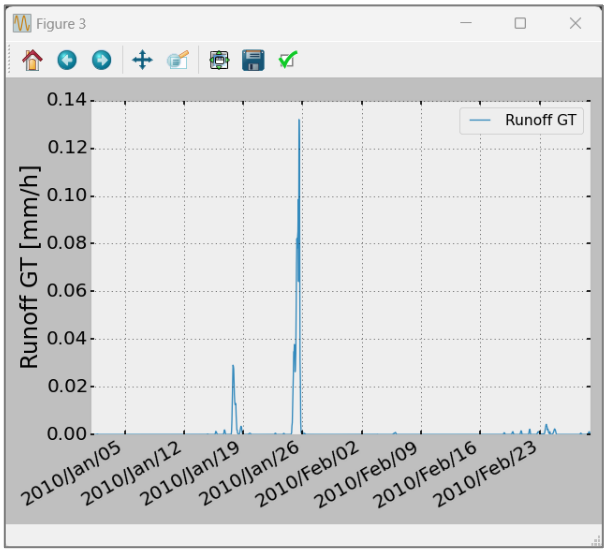


Figure 54. Surface Runoff of the WBS basin obtained using the Generic Model Agents executing the VIC model.

### 4.6.2 *Construct model agents for routing and its coupling with VIC5.0 using generic model agent tools*

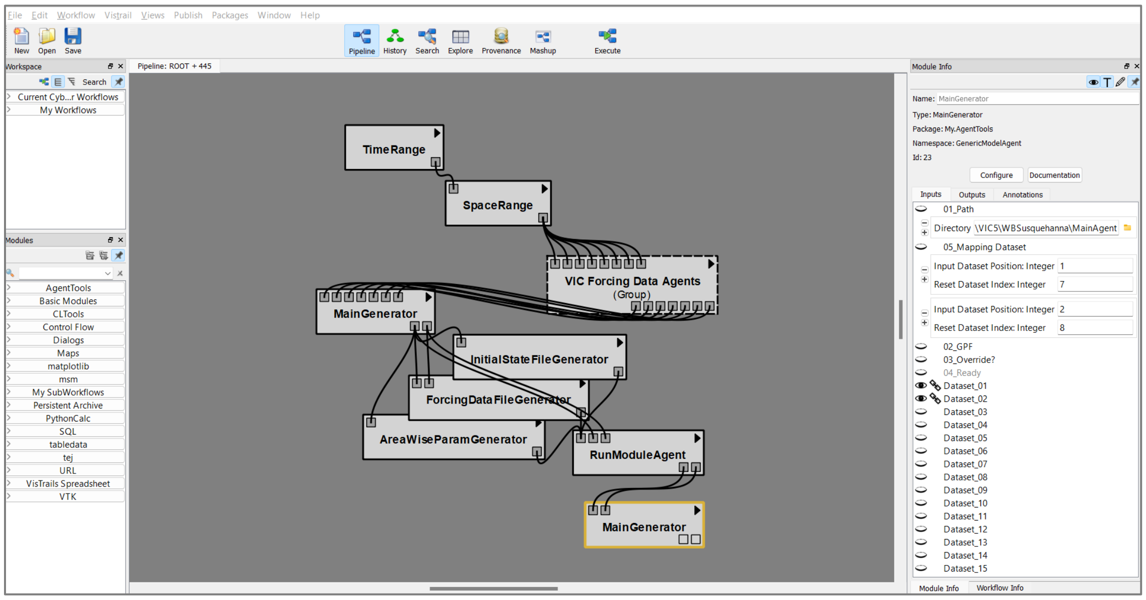
The routing model agent can be constructed following similar concepts and procedures described for VIC5.0 above. That is, to use the five modules (i.e., *MainGenerator, AreaWiseParamGenerator, ForcingDataFileGenerator, InitialStateFileGenerator, and RunModuleAgent*) provided in the generic model agent tools to build the routing agent. To follow this section you have to finish Section 4.6.1 and if you did so, then you will find the files required for this example on your local machine at: “C:\temp\CYBERWATER\GT\Routing\WBSusquehanna\Source” on your local machine

1. ***MainGenerator***

   Again, let’s start with *MainGenerator*. Since it needs to bring forcing data in the same order as the model expects, the user needs to be aware that the routing scheme of the Routing Agent requires only time-series of Surface Runoff and Baseflow. However, the model expects these two variables to be the seventh and eighth columns of all forcing files. This can be done by using the Mapping Dataset option with the MainGenerator module as follow:

   - 01_Path: C:\temp\CYBERWATER\GT\Routing\WBSusquehanna\MainAgent
   - Dataset_01: Connect Surface Runoff output of the *RunModuleAgent*.
   - Dataset_02: Connect Baseflow output of the *RunModuleAgent*.
   - Mapping Dataset: Set “Input Dataset Position: Integer” to 1 and the “Reset Dataset Index: Integer” to 7. This way the Surface runoff will be mapped from the first to the seventh column.
   - Mapping Dataset: Set “Input Dataset Position: Integer” to 2 and the “Reset Dataset Index: Integer” to 8. This way the Baseflow will be mapped from the second to the eighth column.

   The workflow should look as depicted in Figure 55.

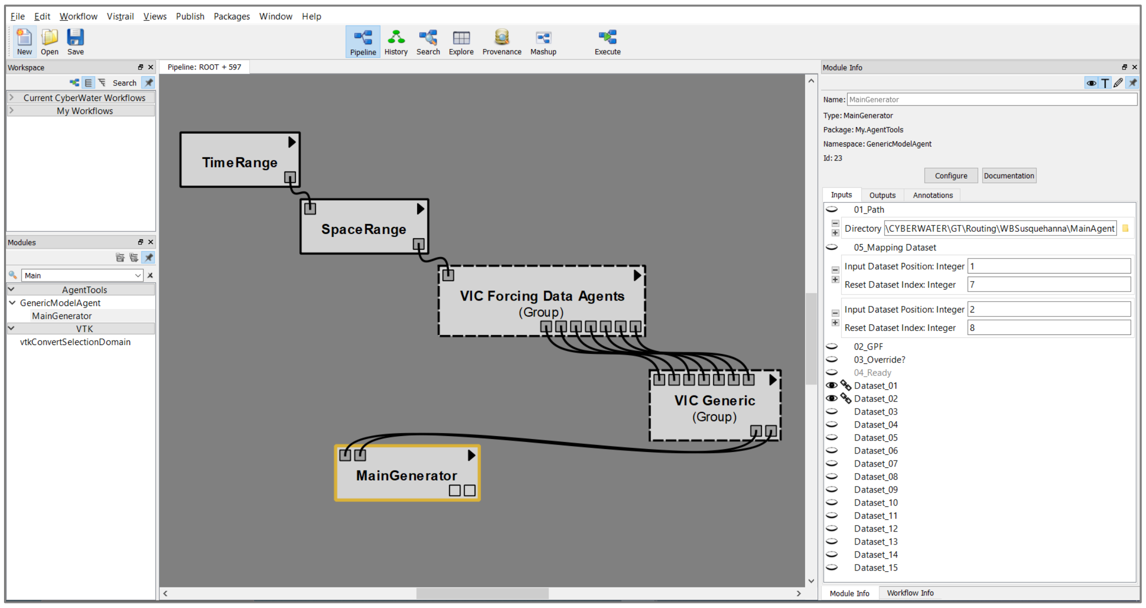


Name: MainGenerator
Type: MainGenerator
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 23

01_Path
Directory \VIC5\WBSusquehanna\MainAgent
05_Mapping Dataset
Input Dataset Position: Integer 1
Reset Dataset Index: Integer 7
Input Dataset Position: Integer 2
Reset Dataset Index: Integer 8
02_GPF
03_Override?
04_Ready
Dataset_01
Dataset_02
Dataset_03
Dataset_04
Dataset_05
Dataset_06
Dataset_07
Dataset_08
Dataset_09
Dataset_10
Dataset_11
Dataset_12
Dataset_13
Dataset_14
Dataset_15

*Figure 55. Coupling of the Generic Model Agent tools to execute both VIC and Routing models.*

Now the set of Generic Model Agent tools responsible for executing the VIC model can be grouped. The resulting workflow is shown in Figure 56.



Name: MainGenerator
Type: MainGenerator
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 23

01_Path
Directory \CYBERWATER\GT\Routing\WBSusquehanna\MainAgent
05_Mapping Dataset
Input Dataset Position: Integer 1
Reset Dataset Index: Integer 7
Input Dataset Position: Integer 2
Reset Dataset Index: Integer 8
02_GPF
03_Override?
04_Ready
Dataset_01
Dataset_02
Dataset_03
Dataset_04
Dataset_05
Dataset_06
Dataset_07
Dataset_08
Dataset_09
Dataset_10
Dataset_11
Dataset_12
Dataset_13
Dataset_14
Dataset_15

*Figure 56. Grouped Generic Model Agent tools that execute the VIC model coupled with a MainGenerator for Routing.*

## 2. *AreaWiseParamGenerator*

Add an *AreaWiseParamGenerator*, and set the inputs as follows:

- WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
- Parameter_Folder_Name: params.
- File_In_00: C:\temp\CYBERWATER\GT\Routing\WBSusquehanna\Source\routing.txt

3. ***ForcingDataFileGenerator***

Add a *ForcingDataFileGenerator*, and set the inputs as follows:

- WD_Path: Connect the 'WD_Path' output port from the *MainGenerator* module.
- DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.
- Forcing_Folder_Name: forcing.
- Forcing_File_Prefix: fluxes.

4. ***InitialStateFileGenerator***

Add an *InitialStateFileGenerator*, and set the inputs as follows:

- WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
- File_In_0: C:\temp\CYBERWATER\GT\Routing\WBSusquehanna\Source\state_20100101

5. ***RunModulelAgent***

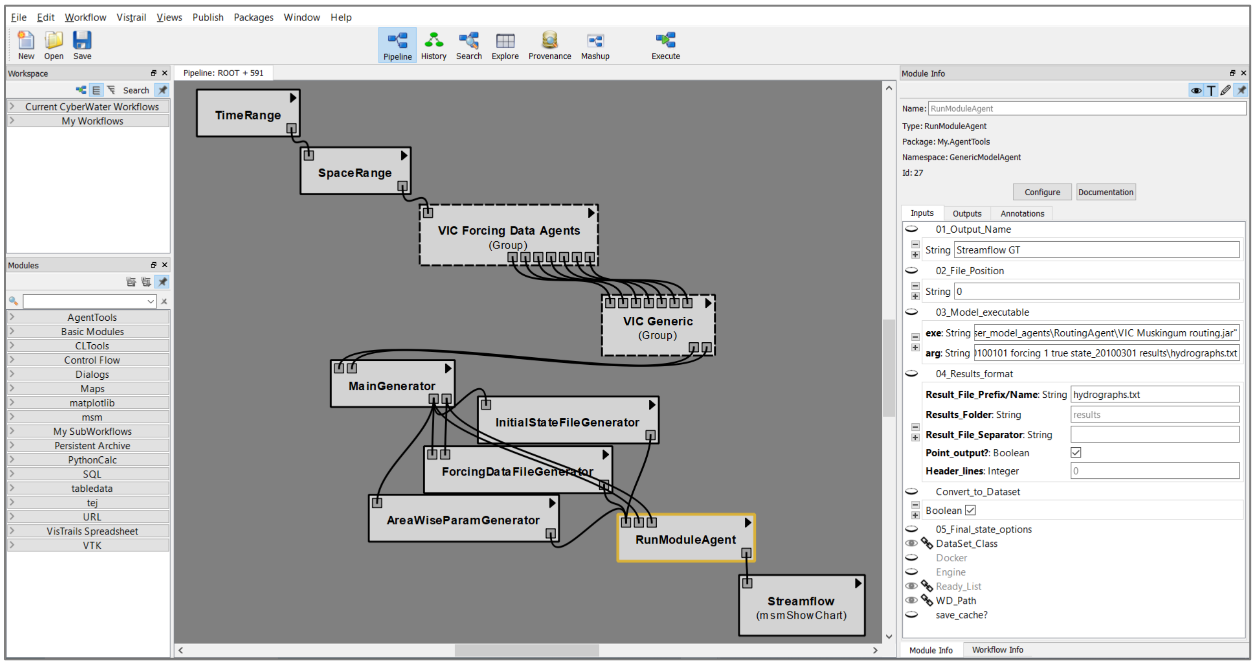
Finally add the *RunModulelAgent*, with the following inputs. Please note that the 'exe' path should be adjusted according to the location of the *VisTrails* in the local machine:

- WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
- DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.
- Ready_List: Connect the output of the three previous components into this port: *ForcingDataFileGenerator*, *AreaWiseParamGenerator*, and *InitialStateFileGenerator*.
- 01_Output_Name: Click the "+" button once and set the string as "Streamflow GT".
- 02_File_Position: Click the "+" button once and set the string as "0".
- 03_Model_executable: Set the exe string as "java -jar"C:\CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_model_agents \RoutingAgent\VIC Muskingum routing.jar"" and the arg string as "params\routing.txt state_20100101 forcing 1 true state_20100301 results\hydrographs.txt"
- 04_Results_format: Set the Result_File_Prefix/Name string as "hydrographs.txt" and the Point_output? as True (check).
- Activate "Boolean" option in Convert_to_Dataset.

Similarly, the user can add the *msmShowChart* module to plot the resulting streamflow:

- units: m$^3$/s

The resulting workflow is shown in Figure 57.



Name: RunModuleAgent
Type: RunModuleAgent
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 27

01_Output_Name
String Streamflow GT
02_File_Position
String 0
03_Model_executable
**exe**: String ser_model_agents\RoutingAgent\VIC Muskingum routing.jar"
**arg**: String )100101 forcing 1 true state_20100301 results\hydrographs.txt
04_Results_format
**Result_File_Prefix/Name**: String hydrographs.txt
**Results_Folder**: String results
**Result_File_Separator**: String
**Point_output?**: Boolean ✓
**Header_lines**: Integer 0
Convert_to_Dataset
Boolean ✓
05_Final_state_options
DataSet_Class
Docker
Engine
Ready_List
WD_Path
save_cache?

*Figure 57. Complete workflow with two sets of Generic Model Agents executing both the VIC model and Routing model.*

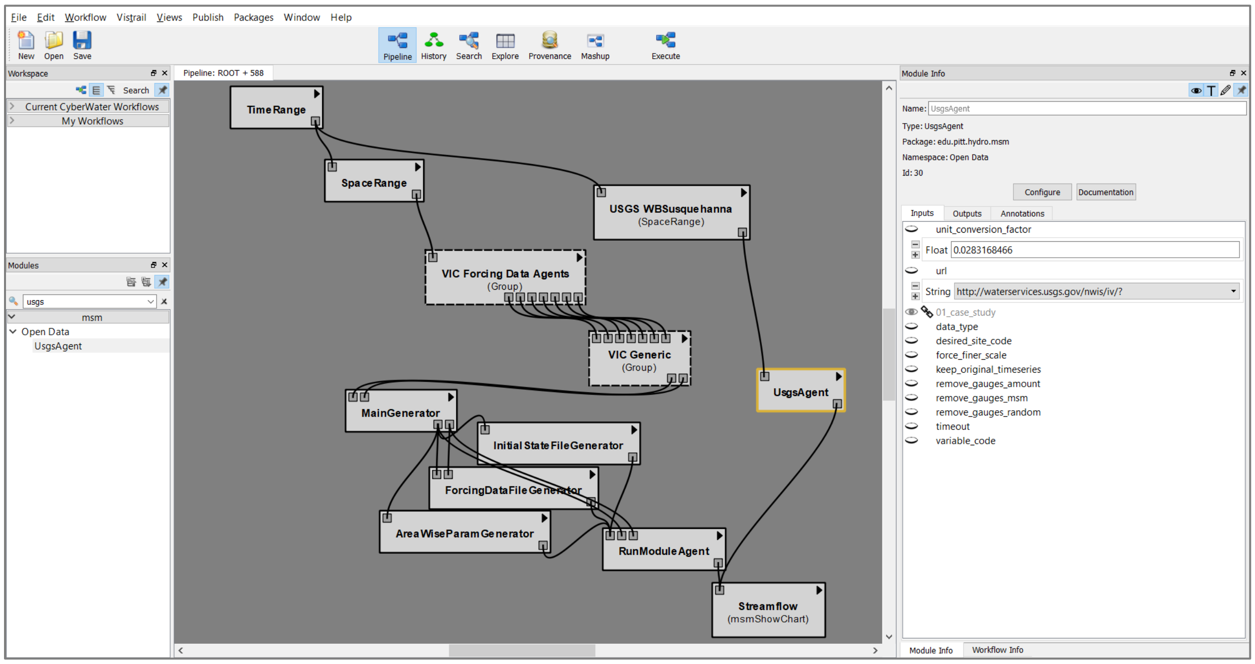
It is worth mentioning that at this point, the set of modules of the Generic Model Agents that execute routing can be grouped as well. Now, you can compare the streamflow values with the USGS measurements. To do so, the user needs to bring a new *SpaceRange* box. This has to be re-done because in the original *SpaceRange* defined region it includes multiple USGS hydrometric stations. It is important to note that the *USGSAgent* brings the information of all stations inside the defined spatial boundaries. Previously in Section 3.1.3 of this manual, this additional *SpaceRange* was not necessary, because the area that is covered by the *SpaceRange* for the French Creek watershed is small enough which only includes one USGS hydrometric station. Therefore, a new *SpaceRange* has to be defined here, where only the outlet station is included. For this, set the following inputs:

- x_max: -76.777
- x_min: -76.977
- y_max: 41.067
- y_min: 40.867

This box needs to be connected to the original *TimeRange*. Now, a *USGSAgent* module can be connected to this new *SpaceRange*. Make sure the inputs are:

- unit_conversion_factor: 0.0283168466.
- url: pick the <http://waterservices.usgs.gov/nwis/iv/?> option.

Connect *UsgsAgent* to the *msmShowChart* module as shown in Figure 58. The workflow in Figure 58 can be downloaded from CyberWater website <7. VIC5 Coupled with Routing Model Simulation>.


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Search
Current CyberWater Workflows
My Workflows
Modules
usgs
msm
Open Data
UsgsAgent
Pipeline: ROOT + 588
Time Range
Space Range
USGS WBSusquehanna (SpaceRange)
VIC Forcing Data Agents (Group)
VIC Generic (Group)
UsgsAgent
MainGenerator
Initial State File Generator
ForcingDataFileGenerator
AreaWiseParamGenerator
RunModuleAgent
Streamflow (msmShowChart)
Module Info
Configure
Documentation
Inputs Outputs Annotations
Module Info Workflow Info

Name: UsgsAgent
Type: UsgsAgent
Package: edu.pitt.hydro.msm
Namespace: Open Data
Id: 30

unit_conversion_factor
Float 0.0283168466
url
String http://waterservices.usgs.gov/nwis/iv/?
01_case_study
data_type
desired_site_code
force_finer_scale
keep_original_timeseries
remove_gauges_amount
remove_gauges_msm
remove_gauges_random
timeout
variable_code

*Figure 58. Workflow with Generic Tools executing VIC and Routing, compared with USGS measurements.*

The resulting chart is depicted in Figure 59.

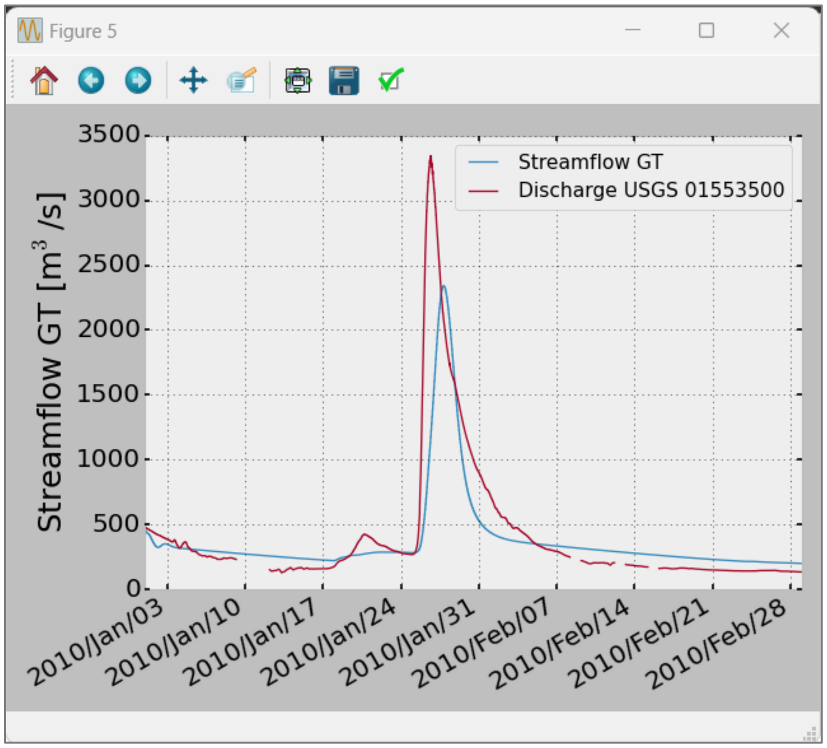


*Figure 59. Resulting streamflow of the WBS basin obtained with Generic Model Agents.*

- To enhance user convenience in replicating the VIC5 and Routing Model Workflow, which is constructed using the provided model agent toolkits, users can effortlessly substitute the original sub-workflow with VIC5Agent_g[1] and RoutingAgent_g[2]. This straightforward replacement process allows for the easy assembly of the corresponding model within the overall workflow. As illustrated in Figure 60, this method simplifies the construction of the desired workflow configuration. The final layout of this user-customized workflow, achieved through such replacements, is readily accessible and can be easily followed. The workflow in Figure 61 can be downloaded from CyberWater website <13. VIC5 Coupled with Routing Model Simulation with Generated Model Agents>.

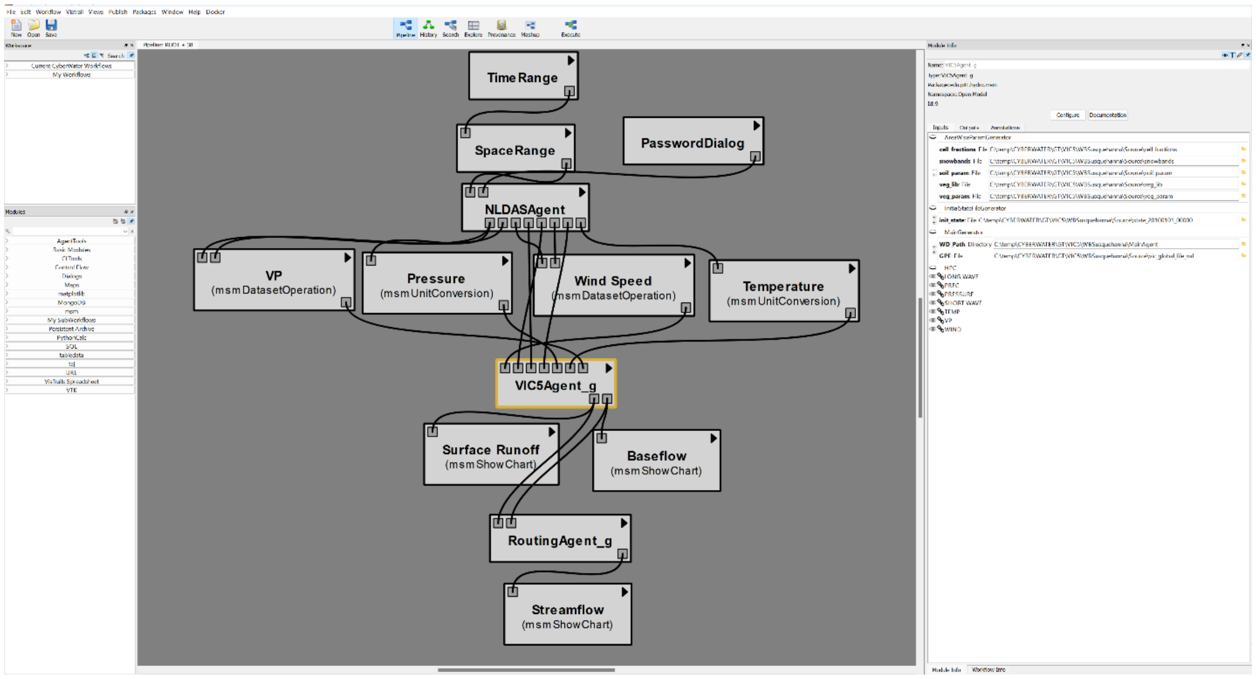

Time Range
Space Range
PasswordDialog
NLDASAgent
VP (msm DatasetOperation)
Pressure (msm UnitConversion)
Wind Speed (msm DatasetOperation)
Temperature (msm UnitConversion)
VIC5Agent_g
Surface Runoff (msm ShowChart)
Baseflow (msm ShowChart)
RoutingAgent_g
Streamflow (msm ShowChart)

*Figure 60. Workflow with VIC5Agent_g module and RoutingAgent_g module.*

### 4.6.3 *Construct model agents for DHSVM using generic model agent toolkits*

DHSVM, the Distributed Hydrology Soil Vegetation Model, is a useful and effective tool for simulating water movement within watersheds. Its high-resolution capabilities allow it to account for variations in topography, soil, and vegetation, leading to more accurate predictions of snowmelt, streamflow, and other crucial hydrological processes. More details about this model can be found be found at: https://www.pnnl.gov/projects/distributed-hydrology-soil-vegetation-model.

In this section, we will show you an example on how to integrate another model, DHSVM, into the CyberWater system using the generic model agent tools. For this example, an hourly simulation will be performed on a relatively small watershed inside the state of Pennsylvania called *Indiantown Run basin*[3] with a drainage area of about 153 km$^2$. The DHSVM requires 7 forcing data: Pressure, Precipitation, Relative Humidity, Temperature, Wind Speed, Longwave Radiation, and Shortwave Radiation. We will use

[1] Located at msm>Open Model>VIC5Agent_g

[2] Located at msm>Open Model>RoutingAgent_g

[3] https://waterdata.usgs.gov/monitoring-location/01572950/#parameterCode=00065&period=P7D&showMedian=false

the NLDASAgent to download these forcing data, but you can download them from *Hydroshare* to save some time. These pre-downloaded data files are compressed and available at:

<Examples Data/Forcing Data/NLDAS/-76.655_40.425_-76.585_40.475>

Unzip this file and put the folder "-76.655_40.425_-76.585_40.475" inside the "downloads" folder at:

<C:\CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_data_agents\NLDASAgent\downloads>

Note that when you unzip the file, use the "**Extract here"** option so that no extra folder that has the SAME name is created by the unzipped command. Also note that if you followed the steps in Section 4.5, you already copied all the folders found in the downloaded "NLDAS" folder and pasted them in the "downloads" folder so you will find "-76.655_40.425_-76.585_40.475" already exists.

Again, the user has a choice of not downloading this folder beforehand but obtaining such forcing data through the execution of the workflow instead. In this latter case the downloaded data will be automatically placed into the correct location and folder by the NLDASAgent as indicated by the path above. Similar to the previous examples, we prepared the configuration, parameters and initial state files required by the DHSVM model for the Indiantown Run watershed. All the parameter values used in the following example are default values that can be modified in the future by the user to perform parameter calibrations. These data files are available at *Hydroshare* at:

<Examples Data\GT\DHSVM>

And this example assumes that you will place this folder at:

< C:\temp\CYBERWATER\GT>

Note: If you tested the previous examples (Section 4.6.1) then you have already created the folders "temp", "CYBERWATER" and "GT", so just download the folder "DHSVM" and place it inside the "GT" folder.

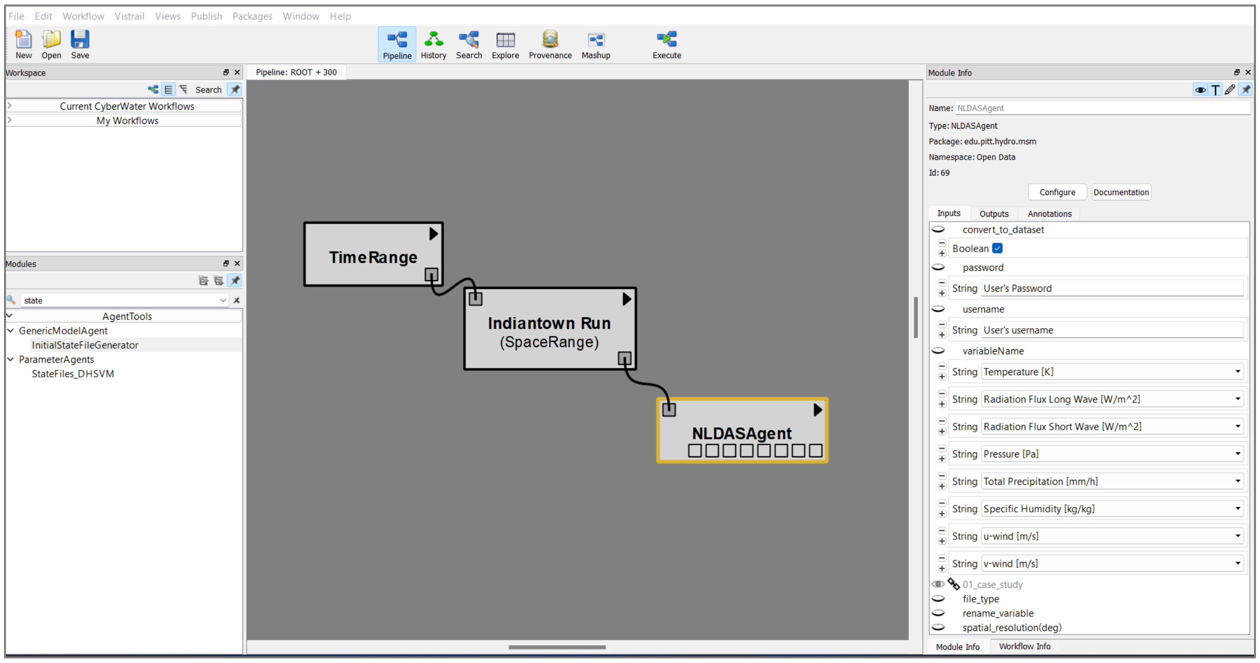
- Add a *TimeRange* box for a simulation between *2007/01/01 00:00:00* (timeini) and *2007/06/01 00:00:00* (timeend).

- Add a *SpaceRange* box with the limits: -76.585, -76.655, 40.475, and 40.425 (x_max, x_min, y_max, y_min, respectively). Connect the '*timerange'* output of the *TimeRange* box with the '*subrange'* input of the *SpaceRange* box.

- Add *NLDASAgent* box and set up the inputs as follows:
    - convert_to_dataset: True (activate)
    - file_type: NetCDF
    - password: Type your NASA password.
    - username: Type your NASA username.
    - variableName: Add the following 8 variables:
        - Temperature [K],
        - Pressure [Pa],
        - Radiation Flux Long Wave [W/m^2],

        - Radiation Flux Short Wave [W/m^2],
        - Total Precipitation [mm/h],
        - Specific Humidity [kg/kg],
        - u-wind [m/s],
        - v-wind [m/s]
    - From the Outputs panel, activate the 8 output ports that correspond to the previous variables (Temperature [K], Pressure [Pa], Radiation Flux Long Wave [W/m^2], Radiation Flux Short Wave [W/m^2], Total Precipitation [mm/h], Specific Humidity [kg/kg], u-wind [m/s], and v-wind [m/s]).

The resulting workflow is shown in Figure 61.



Name: NLDASAgent
Type: NLDASAgent
Package: edu.pitt.hydro.msm
Namespace: Open Data
Id: 69

convert_to_dataset
Boolean ✓
password
String User's Password
username
String User's username
variableName
String Temperature [K]
String Radiation Flux Long Wave [W/m^2]
String Radiation Flux Short Wave [W/m^2]
String Pressure [Pa]
String Total Precipitation [mm/h]
String Specific Humidity [kg/kg]
String u-wind [m/s]
String v-wind [m/s]
01_case_study
file_type
rename_variable
spatial_resolution(deg)

*Figure 61. Workflow with the CyberWater NLDASAgent bringing 8 different variables to execute the DHSVM model*

- The DHSVM requires precipitation to be in *m/h* and temperature to be in *Celsius*. Add two *msmUnitConversion* boxes to transform the units of the *Total Precipitation* and *Temperature* datasets from *NLDASAgent*:
    - For the total precipitation conversion, you can use:
        - operation: x/1000
        - new units: m/h
        - dataset_name: Connect the *NLDAS* output port for *Total precipitation in mm/h*.
    - For the temperature conversion, you can use:
        - operation: x-273.15
        - new units: C
        - dataset_name: Connect the *NLDAS* output port for *Temperature in K*.

- Also, the DHSVM requires Wind Speed and Relative Humidity as forcing data inputs. You can calculate both using the *msmDatasetOperation* module as follows:

- For Wind Speed, add one *msmDatasetOperation* with the following inputs:
    - operation: pow(x*x+y*y,0.5)
    - units: m/s
    - variable_name: Wind Speed
    - dataset_name1: Connect the *NLDAS* output port for *U*-Wind *in m/s.*
    - dataset_name2: Connect the *NLDAS* output port for *V-Wind in m/s.*
- Relative Humidity can be calculated using the *Pressure*, *Specific Humidity*, and *Temperature* datasets downloaded from *NLDASAgent*. A modified version of the Clausius-Clapeyron equation can be used as an approximation[1]:

$$RH(\%) = 0.263 * P * SH * e^{-\frac{17.3T}{T+237.3}}$$

Where:

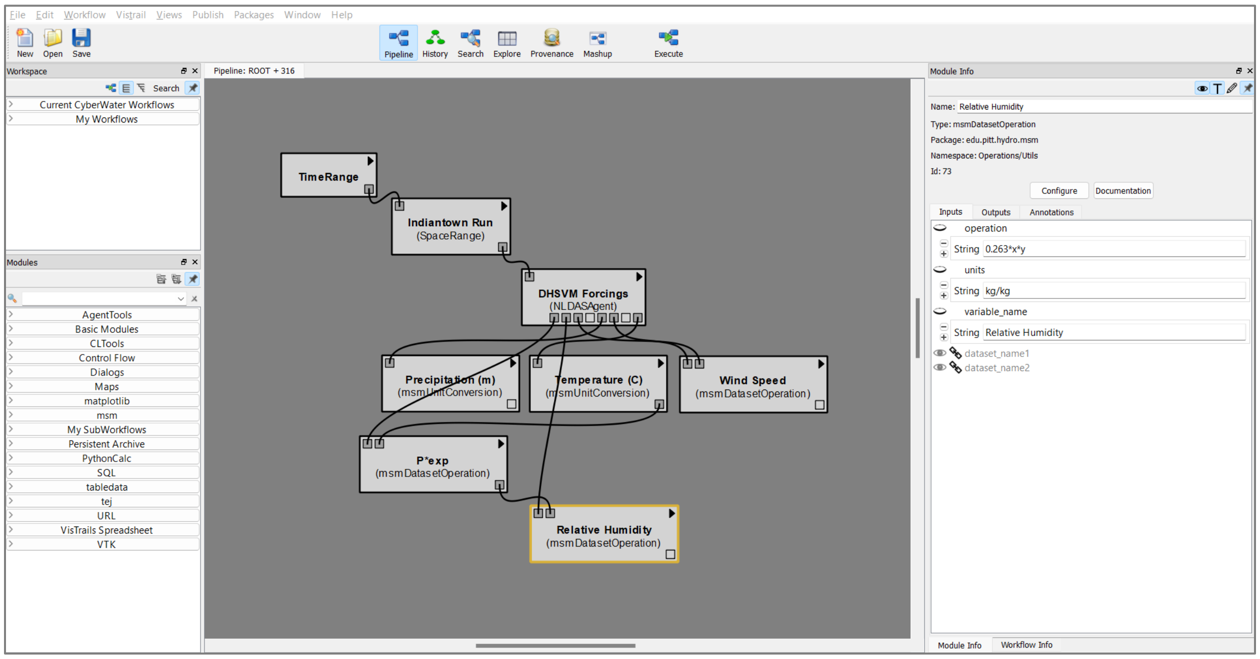
- RH (%) is the relative humidity expressed in percentage. That is, in DHSVM, the input value for RH of 70%, for example, would be 70 instead of 0.7.
- P is the atmospheric pressure (Pa)
- SH is the Specific Humidity (dimensionless)
- T is the air temperature (C)

This operation can be included in two-steps using *msmDatasetOperation* module, with their inputs set as:

- First *msmDatasetOperation*:
    - operation: x/(2.7183**((17.3*y)/(y+237.3)))
    - units: kg/kg
    - variable name: P*exp
    - dataset_name1: Connect the NLDAS output port for *Pressure in Pa*
    - dataset_name2: Connect the *msmUnitConversion* of Temperature in *Celsius*
- Second *msmDatasetOperation*:
    - operation: 0.263*x*y
    - units: kg/kg
    - variable name: Relative Humidity
    - dataset_name1: Connect the NLDAS output port for *Specific Humidity in* kg/kg
    - dataset_name2: Connect the *P*exp* from previous *msmDatasetOperation*

The resulting workflow is shown in Figure 62.

[1] Dingman, S.L., Physical Hydrology, pp. 646, 2002, 2nd Edition, Prentice Hall.


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Search
Current CyberWater Workflows
My Workflows
Modules
AgentTools
Basic Modules
CLTools
Control Flow
Dialogs
Maps
matplotlib
msm
My SubWorkflows
Persistent Archive
PythonCalc
SQL
tabledata
tej
URL
VisTrails Spreadsheet
VTK
Pipeline: ROOT + 316
TimeRange
Indiantown Run (SpaceRange)
DHSVM Forcings (NLDASAgent)
Precipitation (m) (msmUnitConversion)
Temperature (C) (msmUnitConversion)
Wind Speed (msmDatasetOperation)
P*exp (msmDatasetOperation)
Relative Humidity (msmDatasetOperation)
Module Info
Configure
Documentation
Inputs Outputs Annotations
Module Info Workflow Info

Name: Relative Humidity
Type: msmDatasetOperation
Package: edu.pitt.hydro.msm
Namespace: Operations/Utils
Id: 73

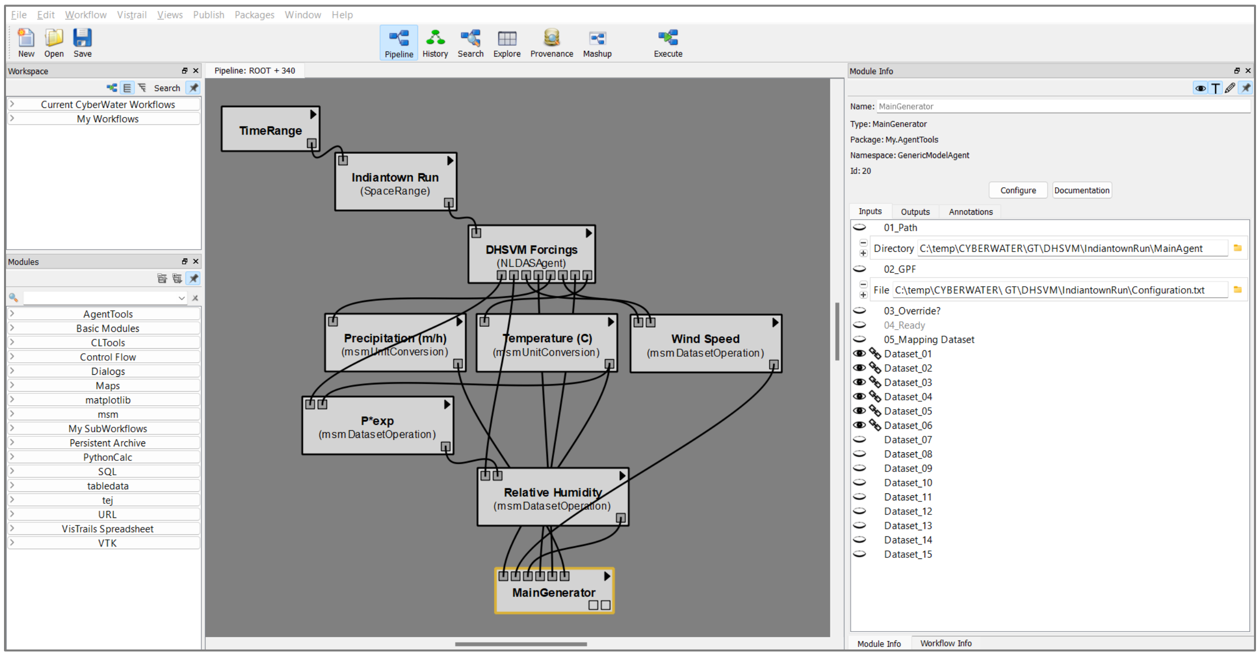
operation
String 0.263*x*y
units
String kg/kg
variable_name
String Relative Humidity
dataset_name1
dataset_name2

*Figure 62. Adding msmUnitConversion and msmDatasetOperation modules to perform preprocessing of the forcing data from NLDASAgent before it goes to the DHSVM.*

Now, we are ready to start constructing a model agent for the DHSVM model using the generic model agent toolkit:

- Add a *MainGenerator* box, the main control component of the Generic Model Agent toolkit. The inputs should be set up as follows:
  - 01_Path: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\MainAgent
    - Please note that you may need to create the MainAgent folder.
  - 02_GPF: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Configuration.txt
  - Activate 7 Datasets:
    - Dataset_01: Connect the *msmUnitConversion* for *Temperature in C*.
    - Dataset_02: Connect the *msmDatasetOperation* for *Wind Speed in m/s*.
    - Dataset_03: Connect the *msmDatasetOperation* for *Relative Humidity*.
    - Dataset_04: Connect the *NLDAS* output port for *Short Wave Radiation in W/m$^2$*.
    - Dataset_05: Connect the *NLDAS* output port for *Long Wave Radiation in W/m$^2$*.
    - Dataset_06: Connect the *msmUnitConversion* for *Precipitation in m/h*

The users should activate these: Dataset_01, Dataset_02, Dataset_03, Dataset_04, Dataset_05, and Dataset_06, if they are not activated yet. The resulting workflow is shown in Figure 63.



Name: MainGenerator
Type: MainGenerator
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 20

01_Path
Directory C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\MainAgent
02_GPF
File C:\temp\CYBERWATER\ GT\DHSVM\IndiantownRun\Configuration.txt
03_Override?
04_Ready
05_Mapping Dataset
Dataset_01
Dataset_02
Dataset_03
Dataset_04
Dataset_05
Dataset_06
Dataset_07
Dataset_08
Dataset_09
Dataset_10
Dataset_11
Dataset_12
Dataset_13
Dataset_14
Dataset_15

*Figure 63. MainGenerator with the inputs required to run the DHSVM.*

- Add an *AreaWiseParamGenerator box*, which sets up the model's parameter files. These files need to be created previously by the user. For more details about how to prepare these files for DHSVM visit https://www.pnnl.gov/model-input-files. In this example, you can use the data we prepared for the Indiantown Run watershed that you already downloaded from Hydroshare. Just add the paths of each of the required parameter files as inputs through the module information panel on the right of the VisTrails window in the *AreaWiseParamGenerator* module, as described below:
  - WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
  - Parameter_Folder_Name: Parameters
  - File_In_00: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\DEM.bin
  - File_In_01: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Flow_dir.bin
  - File_In_02: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Mask.bin
  - File_In_03: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Soil.bin
  - File_In_04: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Soil_depth.bin
  - File_In_05: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\stream_class.txt
  - File_In_06: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\stream_map.txt
  - File_In_07: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\stream_network.txt
  - File_In_08: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\surface_routing.txt
  - File_In_09: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Vegetation.bin

- Add a *ForcingDataFileGenerator* box, which is responsible for organizing the forcing data of the model. This component takes the information plugged in the MainGenerator and saves it into the users' specified folder. The inputs should be set up as follows:
  - WD_Path: Connect the 'WD_Path' output port from the *MainGenerator* module.
  - DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.

  - Forcing_Folder_Name: Forcing
  - Data_Label_Format: %m-%d-%Y-%H.%M.%S
    Note: The DHSVM requires that forcing data has the time in the first column in a format like this "01-03-2007-03.00.00". That's why we used the *Data_label_Format* option here.

- Add an *InitialStateFileGenerator* box for setting the model initial state condition and placing the initial state files in the right folder structure on the working directory. These files need to be created previously by the user. For more details about how to prepare these files for the DHSVM visit https://www.pnnl.gov/model-input-files. In this example, you can use the data we prepared for the Indiantown Run watershed that you already downloaded from Hydroshare.
  - WD_Path: Connect the 'WD_Path' output port from the *MainGenerator* module.
  - Init_State_Folder_Name: state
  - File_In_0: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\state\Channel.State.01.01.2007.00.00.00
  - File_In_1: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\state\Interception.State.01.01.2007.00.00.00.bin
  - File_In_2: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\state\Snow.State.01.01.2007.00.00.00.bin
  - File_In_3: C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\state\Soil.State.01.01.2007.00.00.00.bin

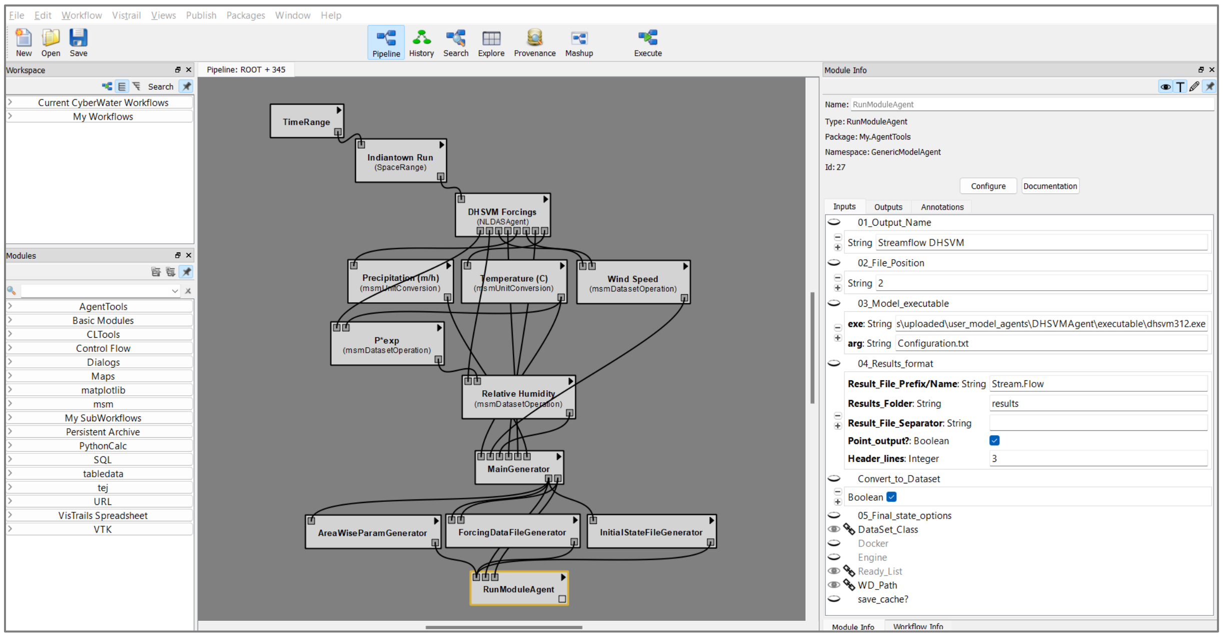
- Add the final component of the Generic Model Agents, the *RunModuleAgent*. Please note that the 'exe' path should be adjusted according to the location of the *VisTrails* in the local machine. Also, the path of the executable file should not include any spaces. For this example, please use the following inputs:
  - WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
  - DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.
  - Ready_List: Connect the output of the three previous modules into this port: *ForcingDataFileGenerator*, *AreaWiseParamGenerator*, and *InitialStateFileGenerator*.
  - 01_Output_Name: Streamflow
  - 02_File_Position: 2
  - 03_Model_executable: Set the exe string as
    - exe String: C:\CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_model_agents\DHSVMAgent\executable\dhsvm312.exe
    - arg string: Configuration.txt

    Note: The 'exe' path should be adjusted according to the location of the installed *CyberWater* in the local machine and this path should not include any spaces.
  - 04_Results_format:
    - Result_File_Prefix/Name string: Stream.Flow
    - Results_Folder string: results
    - Point_output? Boolean: Activate
    - Header_lines Integer: 3
      Note: The streamflow that we are trying to output here is not grid or cell dependent (average for the whole watershed). That's why we must activate the *Point_output* option. Also, the initial output hours usually have odd values for

streamflow due to model initiation. That's why we used the *Header_lines* option to neglect the first three hours in the model simulation results, for example.

- Activate "Boolean" option in "Convert_to_Dataset"

The resulting workflow is shown in Figure 64.



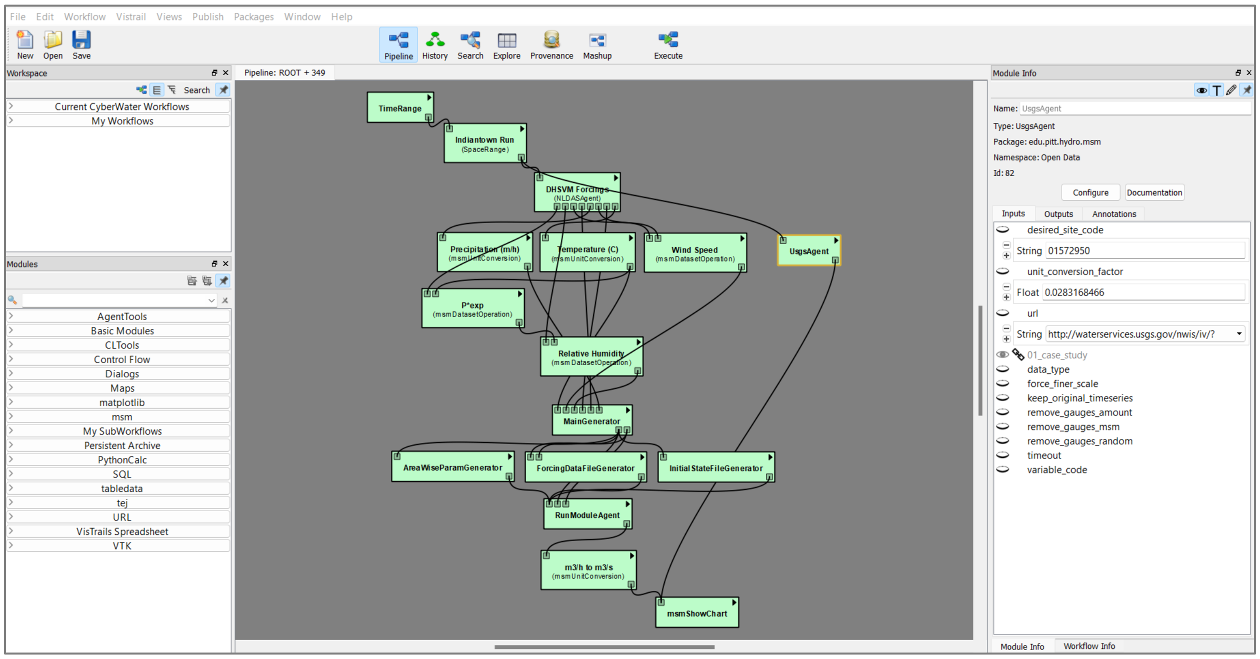
Name: RunModuleAgent
Type: RunModuleAgent
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 27

01_Output_Name
String Streamflow DHSVM
02_File_Position
String 2
03_Model_executable
**exe**: String s\uploaded\user_model_agents\DHSVMAgent\executable\dhsvm312.exe
**arg**: String Configuration.txt
04_Results_format
**Result_File_Prefix/Name**: String Stream.Flow
**Results_Folder**: String results
**Result_File_Separator**: String
**Point_output?**: Boolean ✓
**Header_lines**: Integer 3
Convert_to_Dataset
Boolean ✓
05_Final_state_options
DataSet_Class
Docker
Engine
Ready_List
WD_Path
save_cache?

*Figure 64. Workflow with the full set of Generic Model Agent tools to perform a DHSVM execution.*

- The simulation is set up on an hourly basis, the output units of the DHSVM will be in m$^3$/h. We can use the *msmUnitConversion* module to convert it to m$^3$/s:
  - new_units: m$^3$/s
  - operation: x/3600
- Similarly, to the example of coupling VIC5 with Routing using the generic model toolkit, the USGSAgent can be used to compare the results of the DHSVM with observations. Add UsgsAgent box and set it as follows:
  - desired_site_code: 01572950
    (This is the site number of Indiantown Run basin from USGS)
  - unit_conversion_factor: 0.0283168466
    (The above unit conversion factor is to convert USGS streamflow from ft$^3$/sec to m$^3$/sec. That is, the output discharge will be in m$^3$/s.)
  - url: http://waterservices.usgs.gov/nwis/iv/?
    (This option forces the UsgsAgent to bring measurements with the highest time-resolution available, whereas the default option always retrieves daily data.)

Finally, click the "Execute" button to run the workflow. The final workflow is shown in Figure 65 and the results diagrams are shown in Figure 66. The workflow in Figure 65 can be downloaded from CyberWater website <8. DHSVM Model Simulation>.


File Edit Workflow Vistrail Views Publish Packages Window Help
New Open Save
Pipeline History Search Explore Provenance Mashup Execute
Workspace
Search
Current CyberWater Workflows
My Workflows
Modules
AgentTools
Basic Modules
CLTools
Control Flow
Dialogs
Maps
matplotlib
msm
My SubWorkflows
Persistent Archive
PythonCalc
SQL
tabledata
tej
URL
VisTrails Spreadsheet
VTK
Pipeline: ROOT + 349
TimeRange
Indiantown Run (SpaceRange)
DHSVM Forcings (NLDASAgent)
Precipitation (m/h) (msmUnitConversion)
Temperature (C) (msmUnitConversion)
Wind Speed (msmDatasetOperation)
UsgsAgent
P*exp (msmDatasetOperation)
Relative Humidity (msmDatasetOperation)
MainGenerator
AreaWiseParamGenerator
ForcingDataFileGenerator
InitialStateFileGenerator
RunModuleAgent
m3/h to m3/s (msmUnitConversion)
msmShowChart
Module Info
Module Info
Workflow Info

Name: UsgsAgent
Type: UsgsAgent
Package: edu.pitt.hydro.msm
Namespace: Open Data
Id: 82

Configure Documentation

Inputs Outputs Annotations

desired_site_code
String 01572950
unit_conversion_factor
Float 0.0283168466
url
String http://waterservices.usgs.gov/nwis/iv/?
01_case_study
data_type
force_finer_scale
keep_original_timeseries
remove_gauges_amount
remove_gauges_msm
remove_gauges_random
timeout
variable_code

*Figure 65. Final workflow to execute the DHSVM and compare its results with USGS streamflow measurements.*

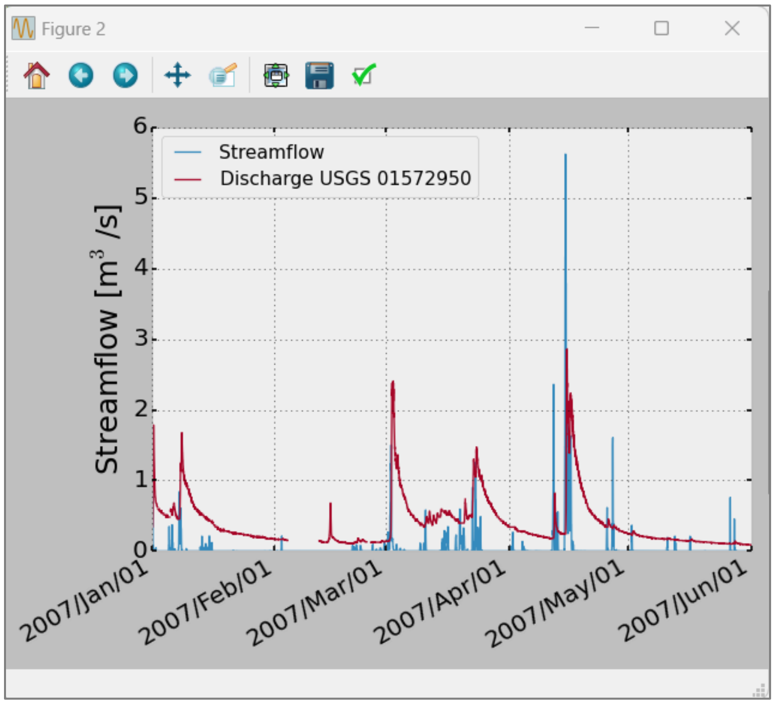
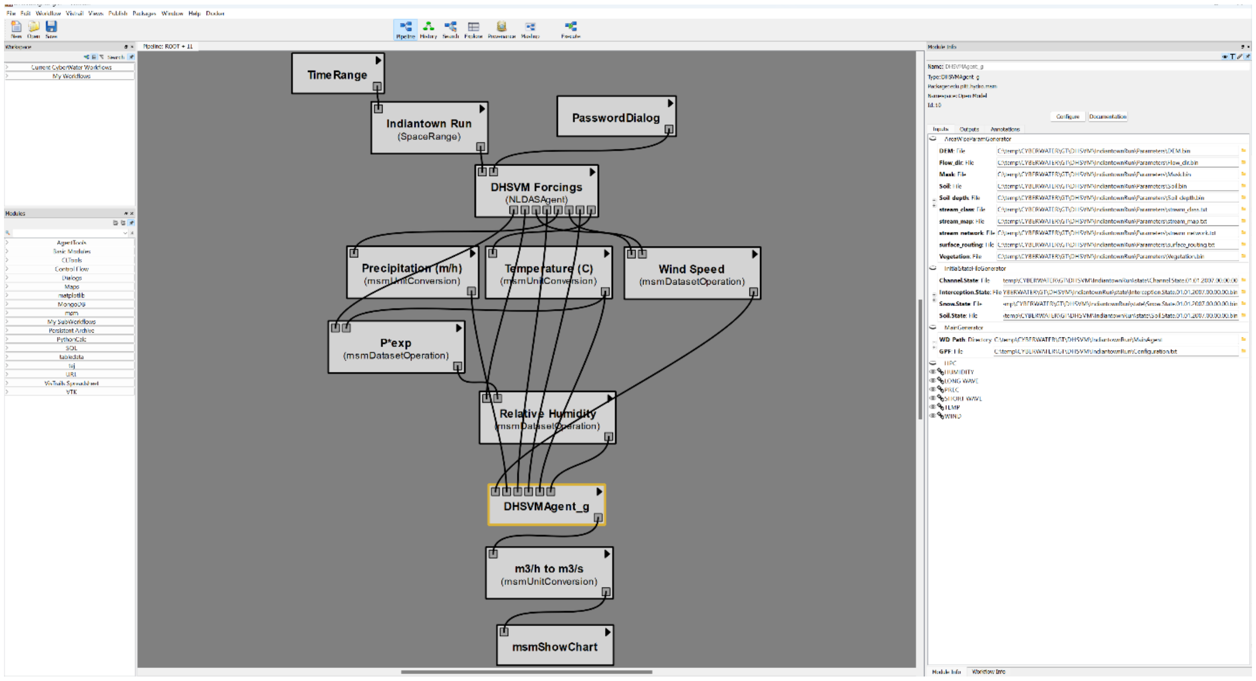


*Figure 66. Chart comparing streamflow values from DHSVM in CyberWater with measurements from USGS.*

- Similarly, the DHSVM model can be integrated using the generated model agent toolkit. Users have the option to utilize the DHSVMAgent_g[1] module as a replacement for the original sub-

[1] Located at msm>Open Model>DHSVMAgent_g

workflow. This allows for the seamless construction of the corresponding DHSVM model within the overall workflow. Figure 67 provides a clear demonstration of this process. By following these steps, users can effortlessly configure and access the resulting workflow tailored to include the DHSVM model. The workflow in Figure 67 can be downloaded from CyberWater website <14. DHSVM Model Simulation with Generated DHSVM Model Agent>.


Time Range
Indiantown Run (SpaceRange)
PasswordDialog
DHSVM Forcings (NLDASAgent)
Precipitation (m/h) (msmUnitConversion)
Temperature (C) (msmUnitConversion)
Wind Speed (msmDatasetOperation)
P*exp (msmDatasetOperation)
Relative Humidity (msmDatasetOperation)
DHSVMAgent_g
m3/h to m3/s (msmUnitConversion)
msmShowChart
Module Info
Configure
Documentation
Inputs
Outputs
Annotations

Name: DHSVMAgent_g
Type: DHSVMAgent_g
Package: edu.pitt.hydro.msm
Namespace: Open Model
Id: 10

AreaWiseParamGenerator
DEM: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\DEM.bin
Flow_dir: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Flow_dir.bin
Mask: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Mask.bin
Soil: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Soil.bin
Soil depth: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Soil depth.bin
stream_class: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\stream_class.txt
stream_map: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\stream_map.txt
stream network: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\stream network.txt
surface_routing: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\surface_routing.txt
Vegetation: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Parameters\Vegetation.bin

InitialStateFileGenerator
Channel.State: File temp\CYBERWATER\GT\DHSVM\IndiantownRun\state\Channel.State.01.01.2007.00.00.00
Interception.State: File YBERWATER\GT\DHSVM\IndiantownRun\state\Interception.State.01.01.2007.00.00.00.bin
Snow.State: File emp\CYBERWATER\GT\DHSVM\IndiantownRun\state\Snow.State.01.01.2007.00.00.00.bin
Soil.State: File temp\CYBERWATER\GT\DHSVM\IndiantownRun\state\Soil.State.01.01.2007.00.00.00.bin

MainGenerator
WD Path: Directory C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\MainAgent
GPF: File C:\temp\CYBERWATER\GT\DHSVM\IndiantownRun\Configuration.txt

HPC
HUMIDITY
LONG WAVE
PREC
SHORT WAVE
TEMP
WIND

*Figure 67. Workflow to execute the DHSVM model with DHSVMAgent_g module.*

## 5. Integrating an One-Layer Water Balance model using generic model agent toolkit

The goal of this task is to demonstrate how to integrate a model, which is an easily understandable model for users, into CyberWater by using the generic model agent toolkit developed in CyberWater without writing any code. In other words, the user will construct their own model agent using the generic model agent toolkit to port a model to CyberWater. To help the user get familiar with the use of the generic model agent toolkit, we provide a simple water balance model for them to integrate. In the following exercise, we first describe in Section 5.1 what this one-layer water balance model is, its required input information, and how to set it up to run in a traditional way. Then, we describe in Section 5.2 how the user can use the generic model agent toolkit to integrate it into CybeWater. However, we recommend each user to try to integrate it themselves first before looking at our provided "solution" (i.e., Section 5.2). All the files required for this exercise can be found on Hydroshare <Examples Data\OneLayerWaterBalance>. It is recommended that the folder "OneLayerWaterBalance" is downloaded into <C:\temp\CYBERWATER>. An understanding of Section 5.1 is essential to the first exercise of Section 5.2, so a close reading will be required.

## 5.1 How does the One-Layer Water Balance model work

The One-Layer Water Balance model is a simple hydrological model which is used here to illustrate how to use the Generic Model Agent Tools to integrate a model into CyberWater. This one-layer model only considers water budget. The input data to the one-layer soil column includes only a time series of precipitation and potential evapotranspiration. Given specific soil parameters, this one-layer model will compute the soil moisture, surface runoff, and evapotranspiration, and output them as time series. Detailed information is given as follows:

Evapotranspiration will be modeled using a beta formulation. The watershed infiltration capacity will be modeled using a linear model. Information on these is given below:

$$E = \beta \cdot E_p ,$$

where

$$\beta = \begin{cases} \dfrac{\theta}{0.8 \cdot \phi}, & \theta \le 0.8\phi \\ \mathbf{1,} & \theta > 0.8\phi \end{cases}$$

Available soil depth D is 180 mm, and soil porosity $\phi$ is 0.47; initial moisture content (in degree of saturation, *s,* which is defined as $s = \theta / \phi$ ) is 0.60, and the soil moisture content in volumetric is represented by $\theta$ ; infiltration capacity (*f'*) depends on *s*, and is given by *f' = $f_o$(1-s)*; and infiltration parameter $f_o$ is 150 mm/day.

Assume that the evapotranspiration and the infiltration capacity depend on the moisture state at the beginning of the day. Using the time series of daily precipitation (Precp) and potential evapotranspiration (Ep) from 2020/06/01 00:00:00 to 2020/06/12 00:00:00 provided below, compute the following variables for each day for a study area defined by longitude between -79 and-80 and latitude between 41 and 40:

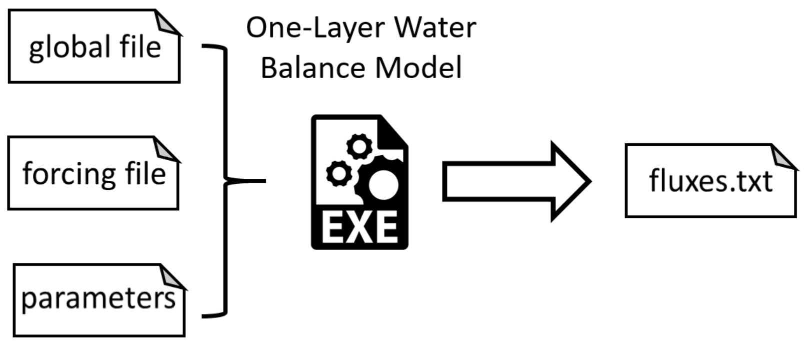
a. Evapotranspiration (mm/day)
b. Soil moisture content expressed in *s* and in $\theta * D$ with a unit of mm/day
c. Surface runoff in unit of mm/day
d. Plot the evapotranspiration, soil moisture, and precipitation and runoff (4 plots) versus time.

| Day | Prec (mm/day) | Ep (mm/day) |
|---|---|---|
| 1 | 10 | 7.5 |
| 2 | 15 | 7.5 |
| 3 | 0 | 8.5 |
| 4 | 0 | 8.5 |
| 5 | 0 | 7.0 |
| 6 | 25 | 6.0 |
| 7 | 50 | 6.0 |
| 8 | 10 | 7.5 |

| | | |
|---|---|---|
| 9 | 0 | 7.5 |
| 10 | 0 | 9.0 |
| 11 | 0 | 9.0 |
| 12 | 0 | 9.0 |

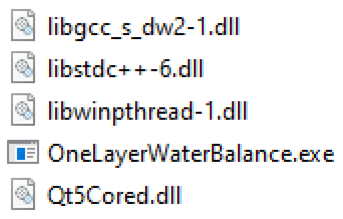
### 5.1.1 *Overview*

The model was developed to allow the user to process their own *precipitation* and (Potential) *evapotranspiration* series and generate the time series of degree of saturation, soil moisture expressed in $\theta * D$, soil moisture content expressed in $\theta$ (dimensionless), actual evapotranspiration, infiltration capacity, and the different types of runoff (i.e., saturation excess runoff and infiltration excess runoff). Below is a schematic of the workflow in running this simple one layer water balance model:



Inside the global file, the user can set up the simulation parameters and the paths where the forcing and the parameter files can be located. For every simulation, the user needs to keep in mind three basic concepts:

1. <u>The working directory:</u> This is where the global file has to be located. The model results will be created here, and the paths for the forcing and the parameter files can be relative to this location.
2. <u>The executable location:</u> The executable of your model can be located anywhere on your PC. As long as we know the path of it, the executable can be invoked. Make sure you keep the five files below under the same folder:

   ```
   libgcc_s_dw2-1.dll
   libstdc++-6.dll
   libwinpthread-1.dll
   OneLayerWaterBalance.exe
   Qt5Cored.dll
   ```

   Please note that except for the executable, the other four files above are the ones needed when run the executable (see more description below).
3. <u>The forcing and parameter files' location:</u> The user is free to place them anywhere, keeping in mind that their respective paths have to be set on the global file. A relative path description is recommended. For example, if the global file and the forcing file are both under the same directory, the relative (to the global file) path of the forcing file is: "./name_of_the_forcing_file."

If it's under another folder named "Forcing" then the relative path is: "./Forcing/ name_of_the_forcing_file.", etc.

### 5.1.2 *Required Inputs*

The One-Layer Water Balance model requires three files to work. Additionally, the model comes with the following four dynamic libraries that have to remain under the same folder as the executable:

- libgcc_s_dw2-1.dll
- libstdc++-6.dll
- libwinpthread-1.dll
- Qt5Cored.dll

#### *5.1.2.1 Global file*

The *global file* is the one that controls the simulation. It has to be inside the working folder, where the execution is performed. That is where it creates a result file under "./results/fluxes.txt." If the "results" folder does not exist, the model automatically creates one.

The One-Layer Water Balance model only requires three simulation parameters given in the *global file*:

- **EVAPOTRASPIRATION_ACTIVE:** TRUE if evapotranspiration is subtracted from precipitation when there is precipitation. FALSE if evapotranspiration is always subtracted from soil moisture regardless if there is precipitation.
- **SOIL_LIB:** The relative path of the soil parameter file
- **FORCING:** File where forcing data is stored

#### *5.1.2.2 Parameter file*

Since the One-Layer Water Balance model only requires soil parameters, only one parameter file is needed, and it is called the *soil parameter file* (or soil library). It holds information about parameters that do not change in time. These parameters are listed below:

- **D:** Depth of the soil layer
- **Phi:** Soil porosity
- **f0:** Initial infiltration capacity
- **s0:** Initial saturation
- **critic_c:** Critical coefficient

#### *5.1.2.3 Forcing file*

The forcing time series for the one-layer water balance model should be written in a tab-separated file, following the format below:

| | |
|---|---|
| $\text{Precipitation}_{t1}$ | $\text{Pot.Evapotranspiration}_{t1}$ |
| $\text{Precipitation}_{t2}$ | $\text{Pot.Evapotranspiration}_{t2}$ |
| $\text{Precipitation}_{t3}$ | $\text{Pot.Evapotranspiration}_{t4}$ |
| … | … |
| $\text{Precipitation}_{tn}$ | $\text{Pot.Evapotranspiration}_{tn}$ |

### 5.1.3 *Executing the model in a windows console and reading the results*

Here the user will learn how to use the windows command console to execute one example of the One-Layer Water Balance model and produce the desired results. This example will be given assuming that the working folder is located at "C:\temp\CYBERWATER \OneLayerWaterBalance\MainAgent." The user may need to create and populate this folder themselves from the files found in "C:\temp\CYBERWATER \OneLayerWaterBalance\Source". Simply copy the files in the source folder into the main agent folder. Also, the executable is located in the folder "C:\temp\CYBERWATER\OneLayerWaterBalance\Executable."

#### *5.1.3.1 Values of the global file*

The global file should be located in the working folder with the name "global_file." It will be populated with the following values:

```
EVAPOTRASPIRATION_ACTIVE = True # TRUE if infiltration comes from (precipitation - 
evapotranspiration). FALSE if it comes from precipitation only.
SOIL_LIB = ./params/soil_params # relative path of the soil parameter file
FORCING = ./forcing/data_40.5000_-79.5000 # File where forcing data is stored
```

#### *5.1.3.2 Values of the soil file*

The soil parameter file must have the path "C:\temp\CYBERWATER\OneLayerWaterBalance\MainAgent\params\soil_params". (As described on the relative path in the global file). It will contain the following values:

```
D = 180.0
phi = 0.47
f0 = 150.0
s0 = 0.8
critic_c = 0.8
```

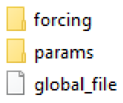
#### *5.1.3.3 Values of the forcing file*

The forcing file must have the path "C:\temp\CYBERWATER\OneLayerWaterBalance\MainAgent\forcing\data_40.5000_-79.5000". (As described on the relative path in the global file). It will contain the following values:

```
10.00        7.5
15.00        7.5
00.00        8.5
00.00        8.5
00.00        7.0
25.00        6.0
50.00        6.0
10.00        7.5
00.00        7.5
00.00        9.0
00.00        9.0
00.00        9.0
```

#### *5.1.3.4 Console execution*

Before starting, your working directory should look like this:

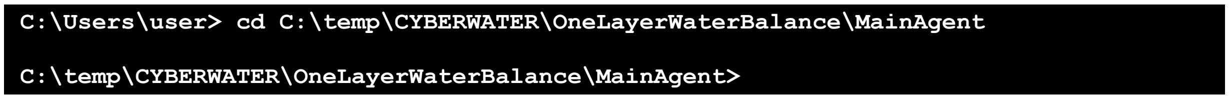


Open a windows console command. You can use one of either steps:

1. Type (Windows+R) and then, on the "Run" window that pops up, type "`cmd`" and press enter.
2. Search for the Command prompt application in your stating menu. It should look like this:

Type the following statements:

1. Change directory to the Working directory:
   `cd C:\temp\CYBERWATER\OneLayerWaterBalance\MainAgent`



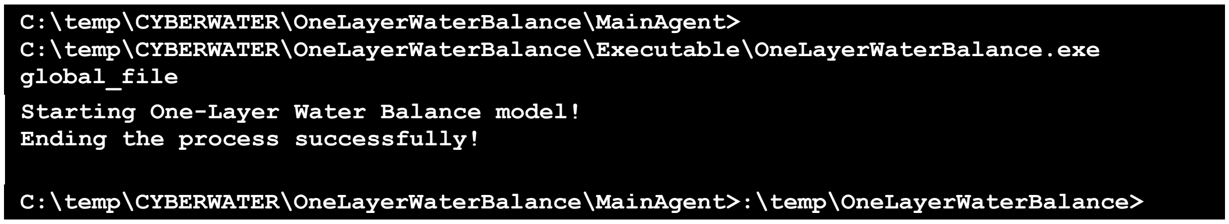
```
C:\Users\user> cd C:\temp\CYBERWATER\OneLayerWaterBalance\MainAgent

C:\temp\CYBERWATER\OneLayerWaterBalance\MainAgent>
```

2. Run the executable using the global file as the only argument:
3. `C:\temp\CYBERWATER\OneLayerWaterBalance\Executable\OneLayerWaterBalance.exe global_file`
   `The syntax of above comment can be represented as:`

   `PATH_OF_THE_EXECUTABLE global_file`

   `where this first phrase (i.e., PATH_OF_THE_EXECUTABLE) represents "C:\temp\CYBERWATER\OneLayerWaterBalance\Executable\OneLayerWaterBalance.exe", and the second phrase (i.e., global_file) represents "global_file"`



```
C:\temp\CYBERWATER\OneLayerWaterBalance\MainAgent>
C:\temp\CYBERWATER\OneLayerWaterBalance\Executable\OneLayerWaterBalance.exe
global_file
Starting One-Layer Water Balance model!
Ending the process successfully!

C:\temp\CYBERWATER\OneLayerWaterBalance\MainAgent>:\temp\OneLayerWaterBalance>
```

If you obtain the "Ending the process successfully!" message, then the model was executed without errors. The results should be located in a new "results" folder inside your working directory. Inside, a "fluxes.txt" file will contain the following information:

```
Precp   Evpt    s       theta    SoilM   beta    actual_E   f     actual_f    Horton    Dunne    Runoff
10    7.5    0.829551    0.376    70.18    1    7.5    30    2.5    0    0    0
15    7.5    0.918203    0.389889    77.68    1    7.5    25.5674    7.5    0    0    0
0    8.5    0.817731    0.431556    69.18    1    8.5    12.2695    0    0    0    0
0    8.5    0.717258    0.384333    60.68    1    8.5    27.3404    0    0    0    0
```

```
0   7   0.643073   0.337111   54.404   0.896572   6.276   42.4113   0   0   0   0
25   6   0.881571   0.302244   74.5809   0.803842   4.82305   53.539   20.177   0   0   0
50   6   1   0.414339   84.6   1   6   17.7643   17.7643   26.2357   7.74523   33.9809
10   7.5   1   0.47   84.6   1   7.5   0   0   2.5   0   2.5
0   7.5   0.911348   0.47   77.1   1   7.5   0   0   0   0   0
0   9   0.804965   0.428333   68.1   1   9   13.2979   0   0   0   0
0   9   0.698582   0.378333   59.1   1   9   29.2553   0   0   0   0
0   9   0.605685   0.328333   51.241   0.873227   7.85904   45.2128   0   0   0   0
```

Before proceeding, the user is **highly recommended** to now try and integrate the One-Layer Water Balance model into *CyberWater* using the generic model agent toolkit to construct your own model agent and then run this one layer water balance model in the CyberWater environment based on all the given input information (i.e., the global file, forcing data, parameter files, and executable) that has been described up to this point **without looking at the following section**. Once this integration exercise is completed, the user is invited to read the following section to see the recommended way to integrate the one-layer water balance model into *CyberWater*. If you don't remember how the generic model agent toolkit is used to integrate a model into *CyberWater*, please refer to the example in Section 4. A brief summary of several key modules is provided below.

There are five modules that form the essential foundation of the generic model agent toolkit. The *MainGenerator* module is the module that receives the forcing data and establishes both the working directory and the global parameter file. The *AreaWiseParamGenerator* module is responsible for organizing the parameter files necessary for execution. The *ForcingDataFileGenerator* module forms the folder and file structure required by the model for its forcing data. The *InitialStateFileGenerator* module is responsible for placing the initial state files in the right folder structure on the working directory. This module is used if the model expects an initial state file, such as initial states of soil moisture and temperature. If such information is not needed or it is provided through other input files by the user's model, this *InitialStateFileGenerator* module is not then needed. Finally, the *RunModuleAgent* executes the model and interprets its results. All these modules can be found in *AgentTools>GenericModelAgent*. Please note that the *InitialStateFileGenerator* module is not necessary for this One Layer Water Balance example. This is because in this example, the initial state (e.g., the initial soil moisture state s0 = 0.8) is provided in the soil parameter file called "soil_params". A brief summary of the relationships of the five modules of the generic model agent toolkit and their functions is shown in Figure 68. It is also important to highlight the *FileToDataSet* module, which can be found in *msm>Files>Input*. It will provide the easiest way for users to introduce their forcing data into *CyberWater*. The user may need to introduce multiple *FileToDataSet* modules to introduce all the required types of forcing data. Documentation of helpful information for each module can be found by clicking the "Documentation" button in *CyberWater*.

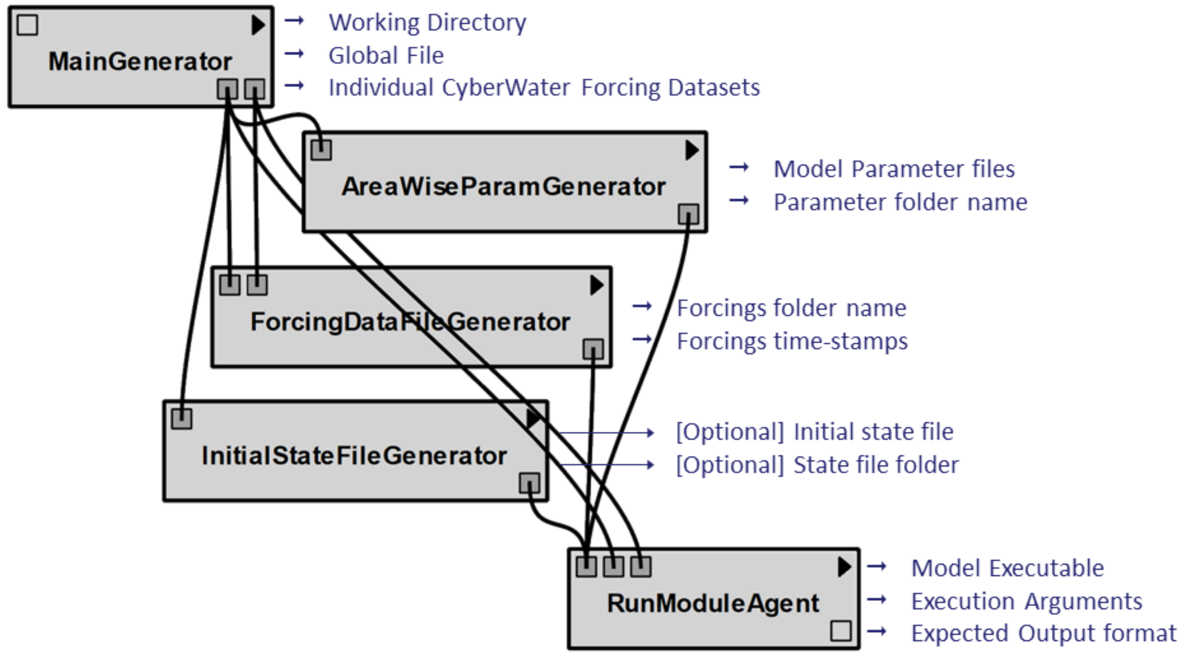


*Figure 68. Relationships of the five modules of the generic model agent toolkit and their functions.*

## 5.2 How to Integrate the One-Layer Water Balance model in *CyberWater*

This section will walk the user through integrating the One-Layer Water Balance model in *CyberWater* using the generic model agent toolkit. It is recommended that the user pauses reading this section before trying to integrate the model themselves as an exercise. It is recommended that you review and refer to Section 4 to refresh your memory on the concept and application of using the generic model agent toolkit to integrate a model into *CyberWater*.

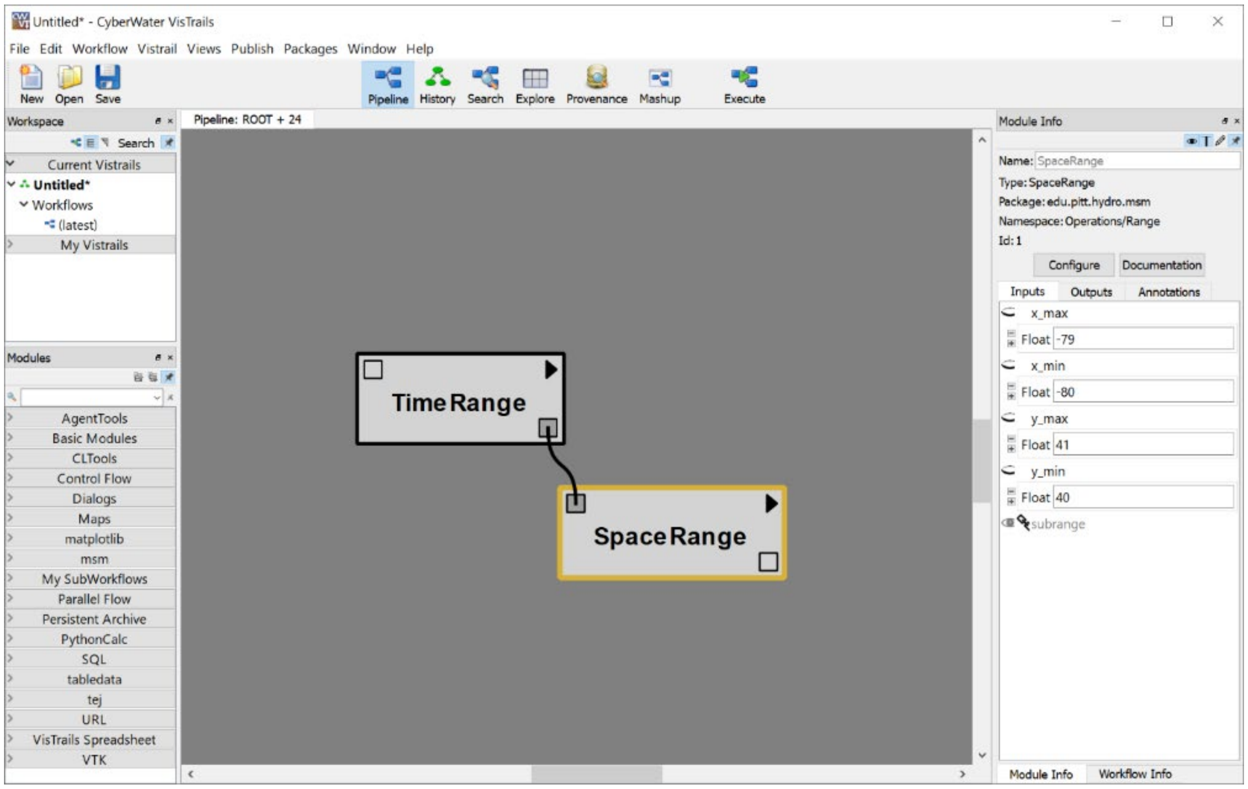
To integrate the One-Layer Water Balance model into *CyberWater*, the user needs to comply with the model structure that the framework requires. First, the user needs to provide both a temporal and a spatial setup for the forcing data.

### 5.2.1 *Building the workflow*

- Add a *TimeRange*[1] box, for a simulation between 2020/06/01 00:00:00 (timeini) and 2020/06/12 00:00:00 (timeend).
- Add a *SpaceRange*[2] box with the same limits as: -79,-80,41, 40 (x_max, x_min, y_max, y_min respectively). Connect the '*timerange'* output of the *TimeRange* box with the '*subrange'* input of the *SpaceRange* box, as shown in Figure 69.

[1] Located at msm>Operations>Range>TimeRange

[2] Located at msm>Operations>Range>SpaceRange



Name: SpaceRange
Type: SpaceRange
Package: edu.pitt.hydro.msm
Namespace: Operations/Range
Id: 1

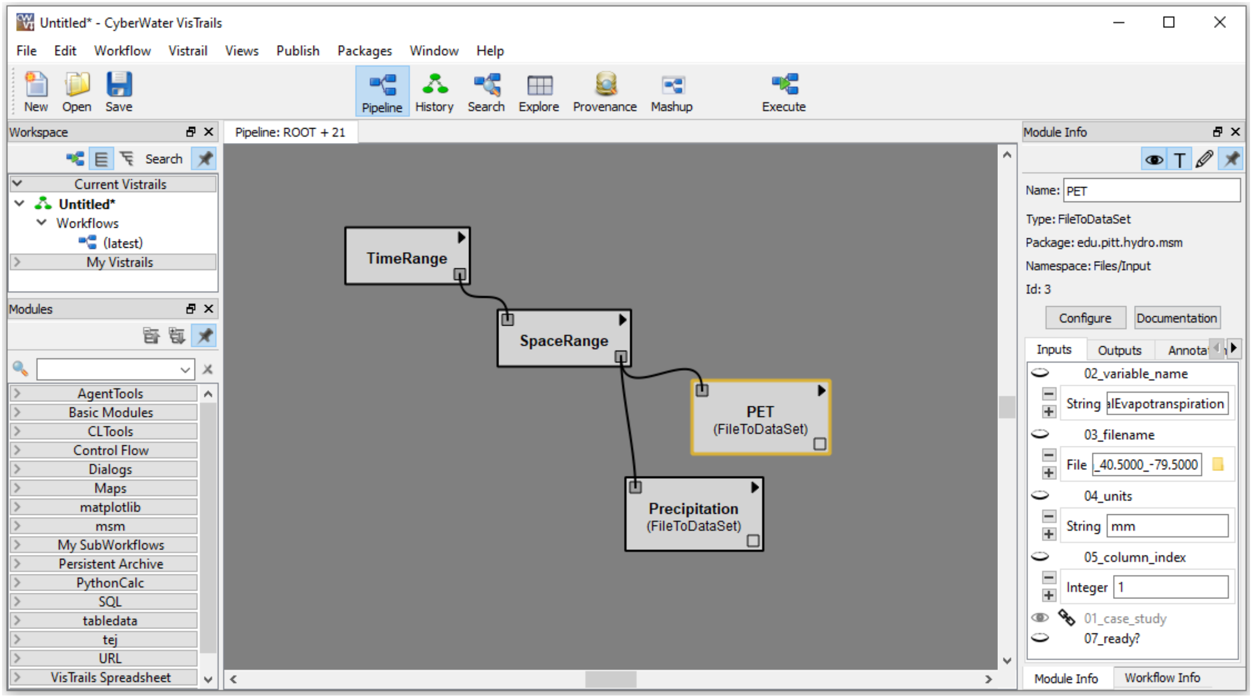
x_max Float -79
x_min Float -80
y_max Float 41
y_min Float 40
subrange

*Figure 69. Time and Space Range setup for the execution of the One-Layer Water Balance model*

- To upload the forcing information into CyberWater, use the *FileToDataSet*[1] module twice (one for precipitation and another for potential evapotranspiration). Their inputs should be:
  - 02_variable_name: *Precipitation* for one, *PotentialEvapotranspiration* for the other one.
  - 03_filename: The full path to the **file** where the forcings are located. C:\temp\CYBERWATER\OneLayerWaterBalance\Source\forcing\data_40.5000_-79.5000
  - 04_units: "mm" for both. This input will only affect the labels when plotting the datasets using a visualization tool.
  - 05_column_index: **0** for precipitation, and **1** for potential evapotranspiration.

  It is important to note that while only one file is needed for all the forcing data, a new *FileToDataSet* module will be needed for each type of data. The final workflow is displayed in Figure 70.

[1] Located at msm>File>Input>*FileToDataSet*



Name: PET
Type: FileToDataSet
Package: edu.pitt.hydro.msm
Namespace: Files/Input
Id: 3

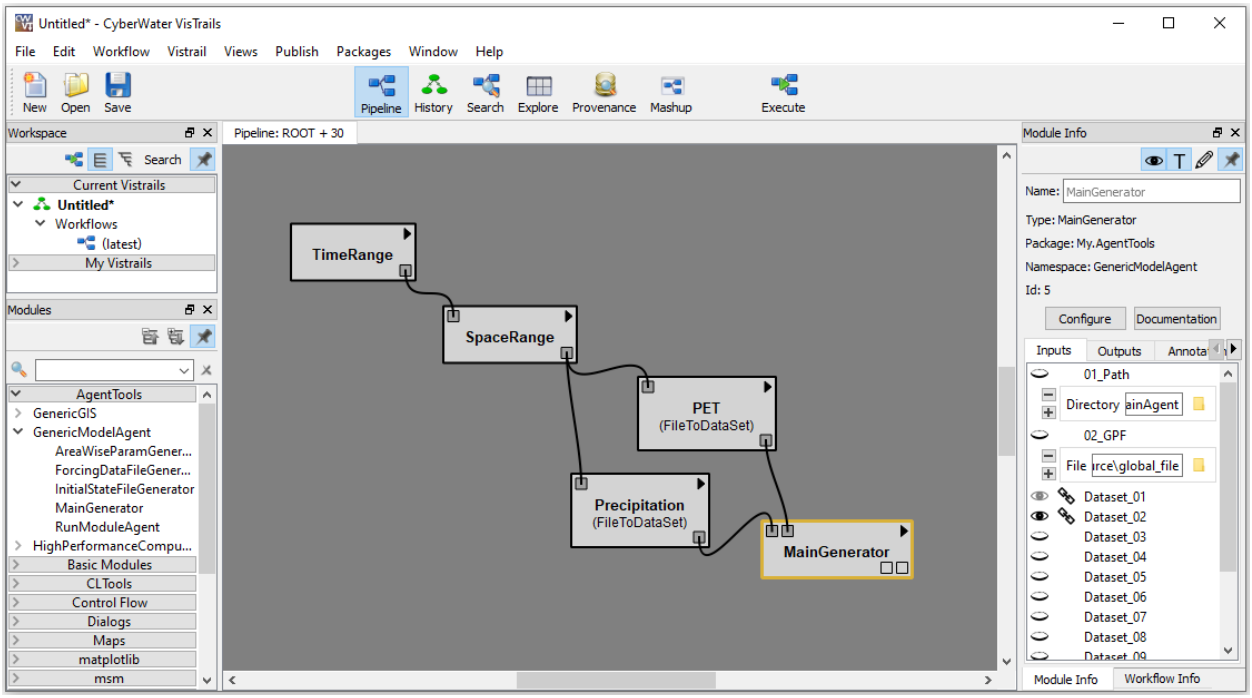
02_variable_name
String alEvapotranspiration
03_filename
File _40.5000_-79.5000
04_units
String mm
05_column_index
Integer 1
01_case_study
07_ready?

*Figure 70. Forcing files for the One-Layer Water Balance model uploaded into CyberWater.*

- Bring a *MainGenerator*[1] box, the main control component of the Generic Model Agent toolkit. This component is responsible for setting up the folder where the simulation will be performed. It receives all the forcing datasets as inputs. The order at which they are added matters since the final forcing files will display the data in the same order: first Dataset_01, then Dataset_02, and so on. Since the OneLayerWaterBalance model expects the first column to be precipitation and the second to be potential evapotranspiration, that is the order in which we will connect the forcings. The users can adjust the paths accordingly for their own folder structure. Please note that Windows imposes a maximum path length of 255 characters, so the user must keep their chosen directory relatively short to prevent an error with an unreachable file. The inputs should be set up as follow:
    - 01_Path (a directory):
      C:\temp\CYBERWATER\OneLayerWaterBalance\MainAgent
    - 02_GPF (a file):
      C:\temp\CYBERWATER\OneLayerWaterBalance\Source\global_file
    - Dataset_01: Connect the module that brings precipitation in mm.
    - Dataset_02: Connect the module that brings potential evapotranspiration in mm.

  The final workflow should look as shown in Figure 71.

[1] Located at AgentTools>GenericModelAgent>MainGenerator



Name: MainGenerator
Type: MainGenerator
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 5

*Figure 71. MainGenerator of the Generic Model Agent Tools prepared to execute the One-Layer Water Balance model.*

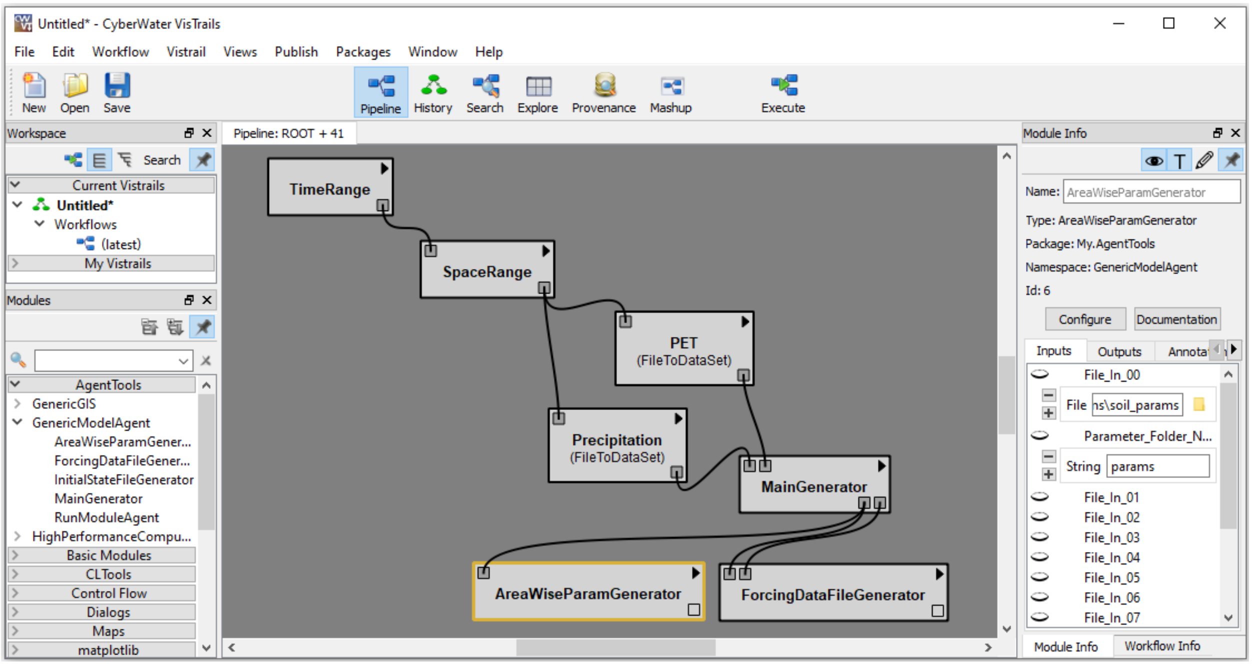
- Add an *AreaWiseParamGenerator*[1], which sets up the model's soil parameter file that the user has previously created. The user must add the required parameter files' paths as inputs through the module information panel on the VisTrails window's right in the *AreaWiseParamGenerator* module. The user needs to adjust the paths accordingly for their folder structure. The inputs are:
    - WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
    - Parameter_Folder_Name: params.
    - File_In_00: C:\temp\CYBERWATER\OneLayerWaterBalance\Source\params\soil_params

- Add a *ForcingDataFileGenerator*[2], responsible for the creation of the forcing data of the model. This component takes the information plugged in the *MainGenerator* and saves it into the users' specified folder. The inputs should be set up as follows:
    - WD_Path: Connect the 'WD_Path' output port from the *MainGenerator* module.
    - DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.
    - Forcing_Folder_Name: forcing.

  After these changes, the workflow should look like the image shown in Figure 72.

[1] Located at AgentTools>GenericModelAgent>AreaWiseParamGenerator

[2] Located at AgentTools>GenericModelAgent>ForcingDataFileGenerator



Name: AreaWiseParamGenerator
Type: AreaWiseParamGenerator
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 6
File ns\soil_params
String params

*Figure 72. One-Layer Water Balance model's parameter and forcing files integrated using the Generic Model Agent Tools.*

- Add the final component of the Generic Model Agents, the *RunModuleAgent*[1]. It is responsible for setting up the path where *CyberWater* should locate the model's executable file. Also, the arguments of the execution, the outputs expected, and their format is configured here. For this example, please use the following inputs:
  - WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
  - DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.
  - Ready_List: Connect the output of the three previous components into this port: *ForcingDataFileGenerator*, and *AreaWiseParamGenerator*.
  - 01_Output_Name: Click the "+" button twice and set the top string as "Soil Moisture" and the bottom string as "Surface Runoff".
  - 02_File_Position: Click the "+" button twice and set the top string as "4" and the bottom string as "11".
    - Note: This information, i.e. "4" and "11", is determined by the individual model (e.g. OneLayerWaterBalance.exe in this example). Also, the counting of "4" and "11" means that it is the 5th and 12th columns respectively in the output file of the model, as counting starts at 0.
  - 03_Model_executable: Set the exe string as "C:\temp\CYBERWATER\OneLayerWaterBalance\Executable\OneLayerWaterBalance.exe" and the arg string as "global_file"
  - 04_Results_format: Set the Result_File_Prefix/Name string as "fluxes.txt", the Point_output? as True (check), and the Header_lines as "1".

[1] Located at AgentTools>GenericModelAgent>RunModuleAgent

- Note: some models are written more flexible and thus the user can enter their own choice of result file name here (e.g., the VIC model). But some models don't allow such a flexibility, in which case the user can only enter the name consistent with the model, like in this example, where the user has to enter "fluxes.txt".

- Activate "Boolean" option in Convert_to_Dataset.

The user will need to go to the "Outputs" tab of the *RunModuleAgent* and make "Output02" visible to be able to see the Surface Runoff. Now, the user can add a *msmShowChart*[1] module to plot the data, with "fluxes" as *variable_name*. Connect the two first outputs of the *RunModuleAgent*, together with the precipitation and potential evapotranspiration forcings. The resulting workflow is shown in Figure 67. The workflow in Figure 73 can be downloaded from CyberWater website <9. One-Layer Water Balance Model Simulation>.

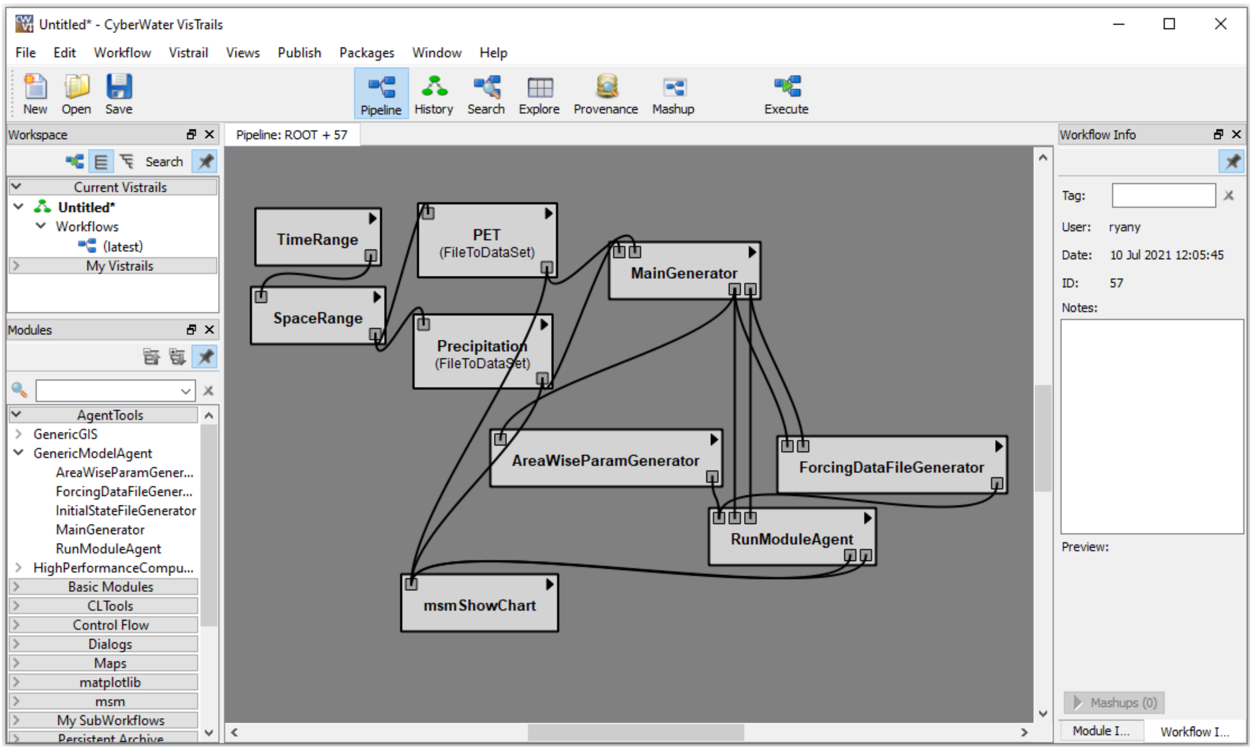


*Figure 73. Complete workflow to execute the One-Layer Water Balance model in CyberWater using the Generic Model Agent Tools.*

[1] Located at msm>Visualization>msmShowChart

- Figure 74 shows the results of the execution.

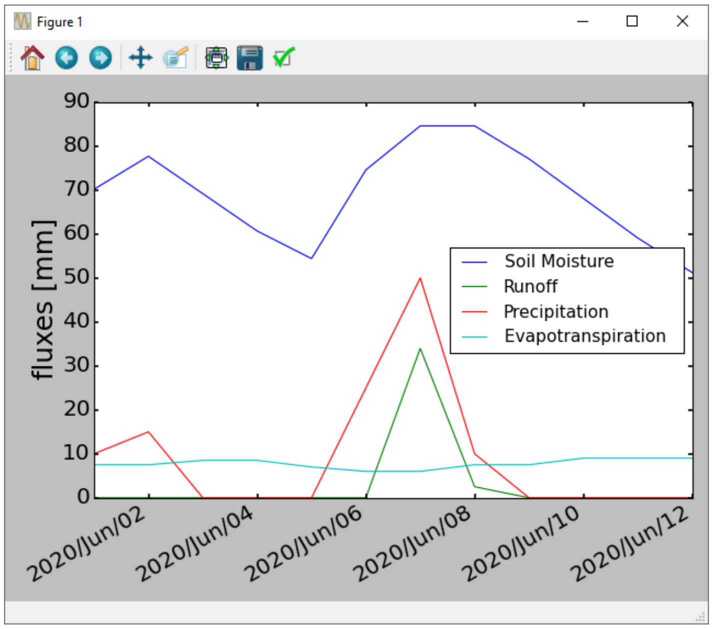


*Figure 74. Complete workflow to execute the One-Layer Water Balance model in CyberWater using the Generic Model Agent Tools when the EVAPOTRANSPIRATION ACTIVE option is set to TRUE.*

- Figure 75 shows the results of the execution when, in the global file, the EVAPOTRANSPIRATION ACTIVE option is set to FALSE.

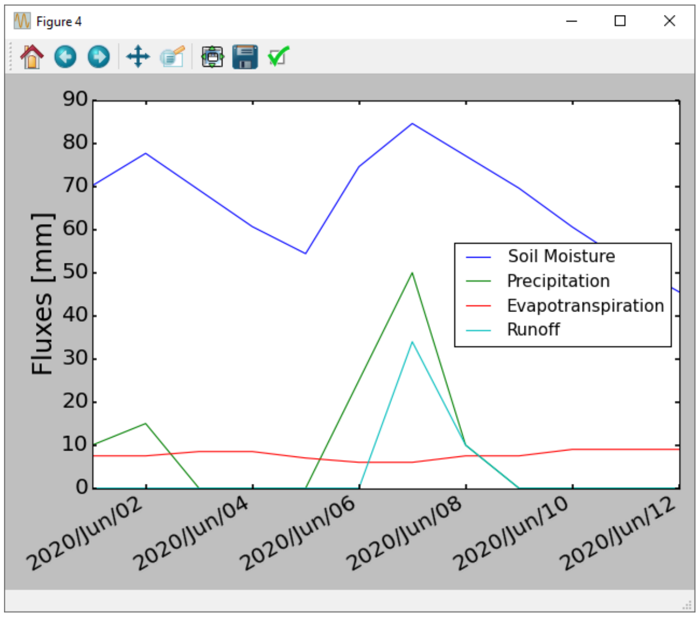


*Figure 75. Complete workflow to execute the One-Layer Water Balance model in CyberWater using the Generic Model Agent Tools when the EVAPOTRANSPIRATION ACTIVE option is set to FALSE.*

# 6. Accessing HPC from CyberWater

This section will guide the user through running their own model on a remote high-performanace computing platform or server through the CyberWater VisTrails HPC module from the generic model agent toolkit. The HPC module provides users with a way to access remote high-performance computing platforms/servers. It is via the SSH protocol to establish a direct connection with a remote platform, for which the user should have a personal access account. In this section, a workflow with WBSusquehanna dataset is taken as an example to indicate how the HPC module works to upload forcing data files, submit a job, and download the result files. The required files for this exercise can be found in the < Examples Data\GT\VIC5> folder on HydroShare. The instructions will assume you download the folder “VIC5” into the directory <C:\temp\CYBERWATER\HPC> such that <C:\temp\CYBERWATER\HPC\VIC5>.

## 6.1 Building the workflow for data preparation

If you already created the workflow for the West-Branch Susquehanna (WBS) following Section 4.6.1 (Figure 47), you can skip this section and just replace the RunModuleAgent module in your existing workflow with the HPC module; then follow instructions in Section 6.2 to set up the HPC module. If you have a HPC account and would like to do this exercise, please don’t forget to set a new directory for the MainAgent folder in the *MainGenerator* module or activate the “Override?” option in the same module so that the new HPC module that you will add doesn’t use the old results already existed in the *MainAgent* folder from your previous exercise’s executions.

- Add a *TimeRange* box for a simulation between 2010/01/01 00:00:00 (timeini) and 2010/03/01 00:00:00 (timeend).
- Add a *SpaceRange* box with the limits of: -76.213, -78.9155, 41.933, and 40.454 (x_max, x_min, y_max, y_min, respectively). Connect the *‘timerange’* output of the *TimeRange* box with the ‘*subrange’* input of the *SpaceRange* box.
- Add *NLDASAgent* box and setup the inputs as follows:
  - convert_to_dataset: True (activate)
  - file_type: NetCDF
  - password: Type your NASA password.
  - username: Type your NASA username.
  - variableName: Add the following 8 variables:
    - Temperature [K],
    - Pressure [Pa],
    - Radiation Flux Long Wave [W/m^2],
    - Radiation Flux Short Wave [W/m^2],
    - Total Precipitation [mm/h],
    - Specific Humidity [kg/kg],
    - u-wind [m/s],
    - v-wind [m/s]
  - From the Outputs panel, activate the 8 output ports that correspond to the previous variables (Temperature [K], Pressure [Pa], Radiation Flux Long Wave [W/m^2], Radiation

Flux Short Wave [W/m^2], Total Precipitation [mm/h], Specific Humidity [kg/kg], u-wind [m/s], and v-wind [m/s]).

- Use the *msmUnitConversion* to transform the units of the Temperature and Pressure datasets. For the Pressure conversion you can use:
  - operation: x/1000
  - new units: kPa

  And for the Temperature conversion:
  - operation: x-273.15
  - new units: C

- Add *msmDatasetOperation* to compute the water vapor pressure. The inputs should be set as follows:
  - operation: (x*y)/0.622
  - units: kPa
  - variable_name: Vapor Pressure
  - dataset_name1: Connect the *NLDAS* output port for *Specific Humidity*.
  - dataset_name2: Connect the *msmUnitConversion* output port for *Pressure in kPa*

- Add another *msmDatasetOperation* to compute the magnitude of the Wind Speed vector, summing its two components. For this, set the inputs as:
  - operation: (x**2+y**2)**0.5
  - units: m/s
  - variable name: Wind Speed
  - dataset_name1: Connect the *NLDAS* output port for *U-Wind*.
  - dataset_name2: Connect the *NLDAS* output port for *V-Wind*.

- Bring a *MainGenerator* box, the main control component of the Generic Model Agent toolkit. The inputs should be set up as follows:
  - 01_Path: C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\MainAgent
    - Please note that you may need to create the MainAgent folder.
  - 02_GPF: C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\Source\vic_global_file_val
  - Activate 7 Datasets:
    - Dataset_01: Connect the *msmUnitConversion* for *Temperature in C*.
    - Dataset_02: Connect the *NLDAS* output port for *Long Wave Radiation in W/m$^2$*.
    - Dataset_03: Connect the *NLDAS* output port for *Precipitation in mm/h*.
    - Dataset_04: Connect the *msmUnitConversion* for *Pressure in kPa*.
    - Dataset_05: Connect the *NLDAS* output port for *Short Wave Radiation in W/m$^2$*.
    - Dataset_06: Connect the *msmDatasetOperation* for *Water Vapor Pressure in kPa*.
    - Dataset_07: Connect the *msmDatasetOperation* for *Wind Speed in m/s*.

- Add an *AreaWiseParamGenerator*, which sets up the model's parameter files. These files need to be previously created by the user. The user must add the required parameter files' paths as inputs through the module information panel on the VisTrails window's right in the AreaWiseParamGenerator module. The user needs to adjust the paths accordingly for their folder structure. Please note that the length of the path has to be under the 255-character maximum imposed by Windows. For this example, these files can be found in < C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\Source>. The files' paths should be added to the *AreaWiseParamGenerator* as follows:

- WD_Path: Connect the 'WD_Path' output from the *MainGenerator* module.
- File_In_00: C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\Source\cell_fractions
- File_In_01: C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\Source\snowbands
- File_In_02: C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\Source\soil_param
- File_In_03: C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\Source\veg_lib
- File_In_04: C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\Source\veg_param
- Parameter_Folder_Name: params

● Add a *ForcingDataFileGenerator*, responsible for the creation of the forcing data of the model. This component takes the information plugged in the *MainGenerator* and saves it into the users' specified folder. The inputs should be set up as follows:
  - WD_Path: Connect the 'WD_Path' output port from the *MainGenerator* module.
  - DataSet_Class: Connect the 'DataSet_Class' output port from the *MainGenerator* module.
  - Forcing_Folder_Name: forcing.

● Add an *InitialStateFileGenerator* to organize the initial state data of a generic simulation. The inputs should be set up as follows:
  - WD_Path: Connect the 'WD_Path' output port from the *MainGenerator* module.
  - File_In_0: C:\temp\CYBERWATER\HPC\VIC5\WBSusquehanna\Source\state_20100101_00000

After these changes, the workflow should look like the image shown in Figure 76.

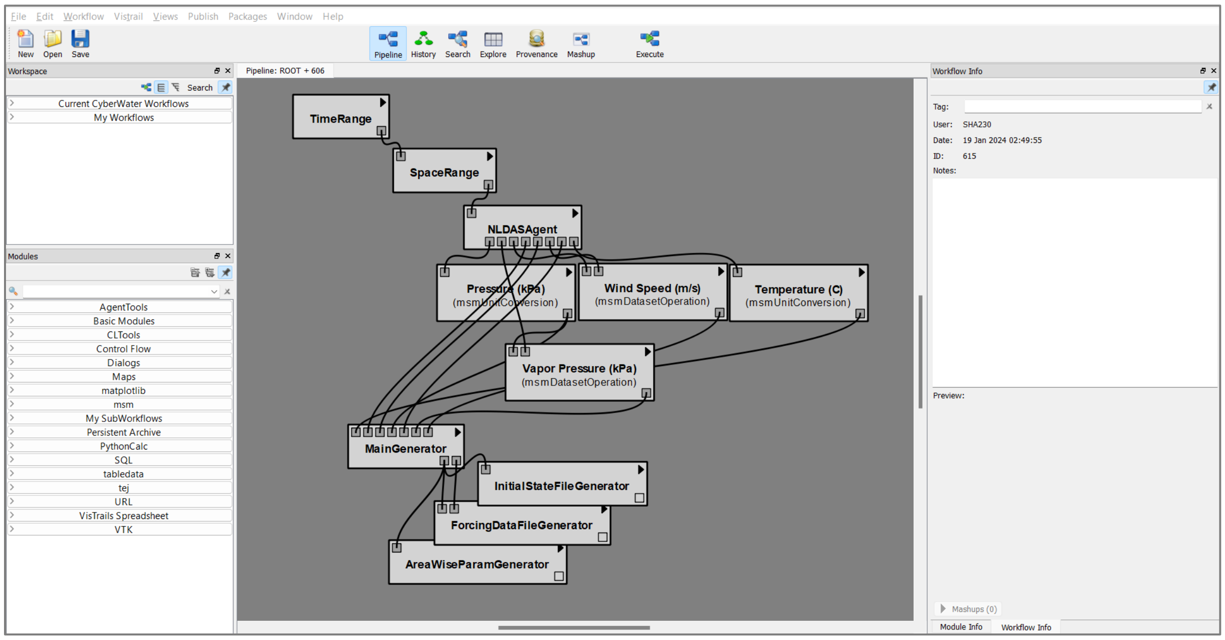


*Figure 76. Workflow with complete data preparation for HPC module.*

## 6.2 HPC Module Configuration

The HPC module is responsible for the execution of a Model on a remote high-performance computing platform or server. For the high-performance computing platform/server selection, 10 plaforms for ssh direct connection are currently provided, including six slurm-based platforms (e.g., bigred3, bridges2, bridges2-shared, stampede2, google cloud, and Jetstream), and 4 bash-based platforms (e.g., sievert, rain, thunder, and lightning). Additionally, users can also add their own HPC platforms/servers (called customized HPC/servers) to connect with and save in the HPC module for future use. Direct connection via SSH with user's individual account.

### 6.2.1 *Direct connection via SSH with user's individual account.*

In this section we assume that users have their own HPC account. If this is not the case, it is still recommended that they set up the HPC module according to the following instructions while skipping (1) Platform Selection and (2) Credential. These two settings will be explained using our community HPC platforms accounts in the next section.

- Add an *HPC*[1] module, which is used to submit the computation job to the remote high-performance computing platforms. The inputs are:
    - (1) Platform Selection:
        - SSH Platform: Bridges2
        - Is_Slurm?: True (activate)
          Note: Bridges2 platform uses Slurm Workload Manager, that's why we set the value of this variable to be true.
    - (2) Credential:
        - Username: [user's username]
        - Password: A password dialog will be popped up during execution of workflow to ask user to enter their corresponding password.
    - (3) Project Configuration:
        - Project Name: CyberWater
        - Email: [user's email]
        - Estimated Runtime: 10
        - Argument: "-g vic_global_file_val"
          Note: Include the double quotation marks in the argument.
        - Result File Prefix: The prefix of all the cells generated by the model. E.g., fluxes
    - (4) Model Source:
        - Executable Program:

          C:\CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_model_agents\VicAgent\VIC5_HPC.exe

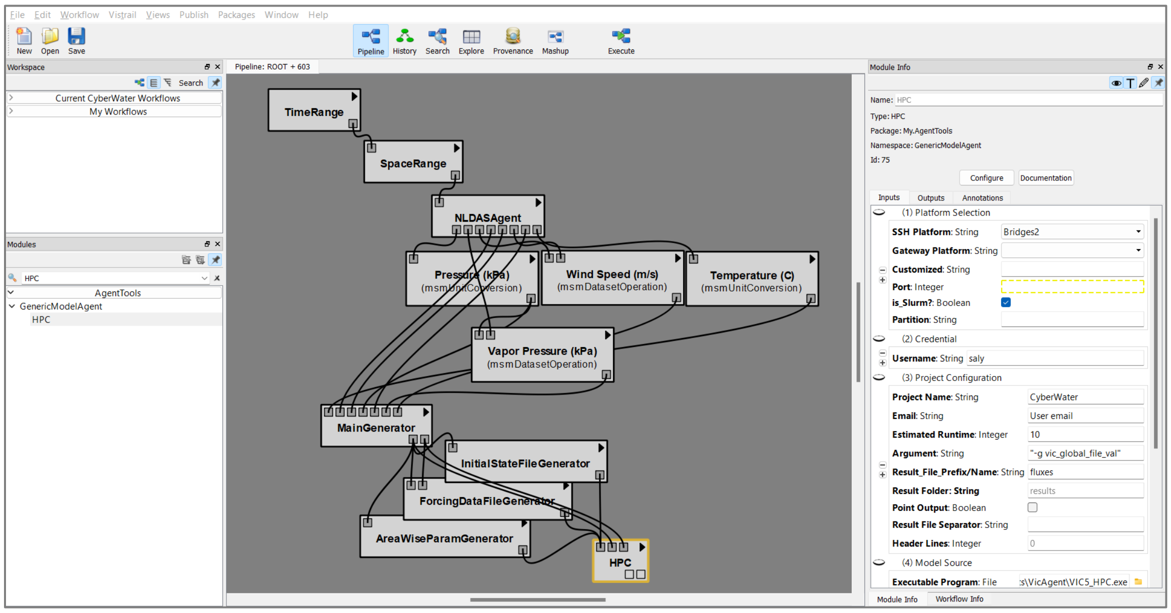
        Notes:

[1] Located at AgentTools>HighPerformanceComputing>HPC

- "VIC5_HPC.exe" is the executable file for VIC5 model on Linux environment.
- We have also included "vicNl.exe" in the same folder which is the executable file of VIC4 model on Linux environment and can be used with the HPC module.
- If a user does not have the executable program available running on the environment of the remote high-performance computing platform/server, the user can submit the source code of the model by entering the path of source code in the text area of `Source Code` below the column `(4) Model Source`. It is very important to note that the folder containing the source code **must** share the same name as the executable that will be generated when the program is compiled. This is how CyberWater finds the executable and naming it otherwise will cause an error.

- (5) Output Datasets:
  Click the "+" button to add a pair of values including output name and file position.
  - File Position: 6
    Output Name: Surface Runoff
  - File Position: 7
    Output Name: Baseflow
- Activate "Boolean" in Convert_to_Dataset
- Ready_List: Connect the output of *ForcingDataFileGenerator, AreaWiseParamGenerator* and *InitialStateFileGenerator* if it exists.
- WD_Path: WD_Path output port of the *MainGenerator*. This is the directory where the simulation files are saved.
- DataSet_Class: DataSet_Class output port of the *MainGenerator*.

The workflow after finishing HPC module configuration is shown in Figure 77.



Name: HPC
Type: HPC
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 75

(1) Platform Selection
**SSH Platform**: String Bridges2
**Gateway Platform**: String
**Customized**: String
**Port**: Integer
**is_Slurm?**: Boolean ✓
**Partition**: String

(2) Credential
**Username**: String saly

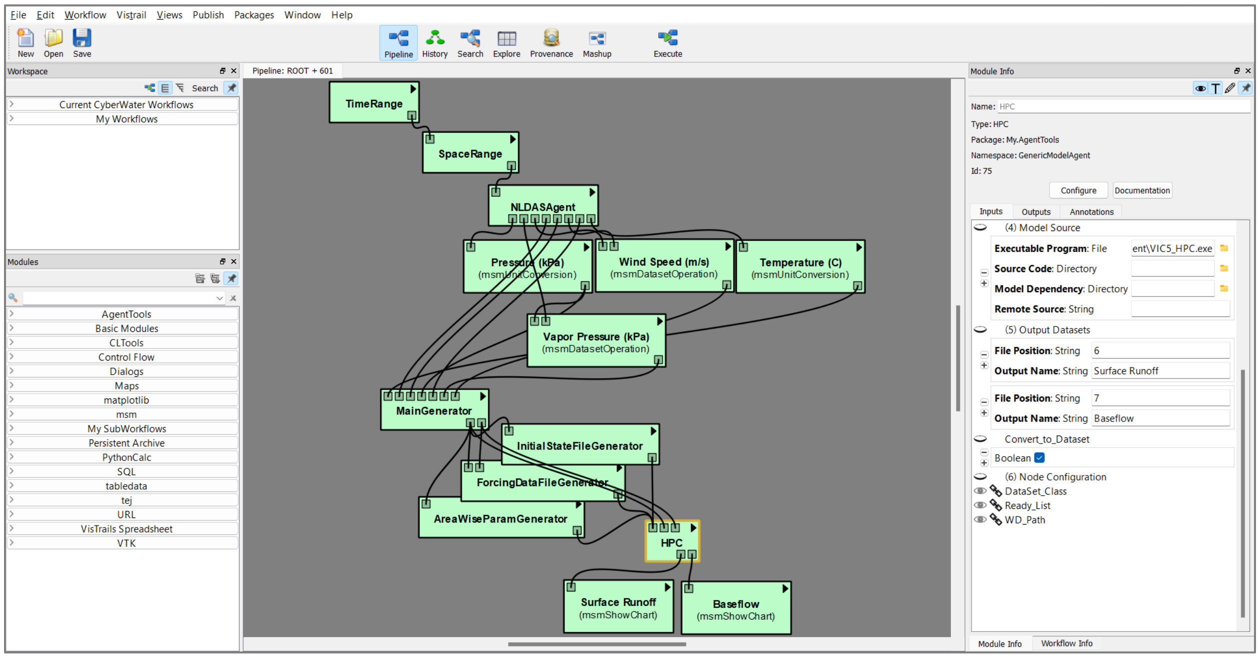
(3) Project Configuration
**Project Name**: String CyberWater
**Email**: String User email
**Estimated Runtime**: Integer 10
**Argument**: String "-g vic_global_file_val"
**Result_File_Prefix/Name**: String fluxes
**Result Folder: String** results
**Point Output**: Boolean ☐
**Result File Separator**: String
**Header Lines**: Integer 0

(4) Model Source
**Executable Program**: File :s\VicAgent\VIC5_HPC.exe

*Figure 77. Workflow with HPC module configuration by selectiong Bridges2 as SSH direct connection platform.*

- Enable the 'Output02' output port on the HPC module by clicking on "Outputs" and then clicking the icon in front of `Output02`. Add 2 *msmShowChart* modules to plot the data, with "Surface Runoff" and "Baseflow" as *variable_name* and with mm/hr as units. Connect the first and second outputs of the HPC. The resulting workflow is shown in Figure 78. The workflow in Figure 78 can be downloaded from CyberWater website <10. VIC5 Model Simulation in High-Performance Computing Platform>.

Click the "Execute" button to run the workflow. The result diagrams are displayed in Figure 79.



Name: HPC
Type: HPC
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 75

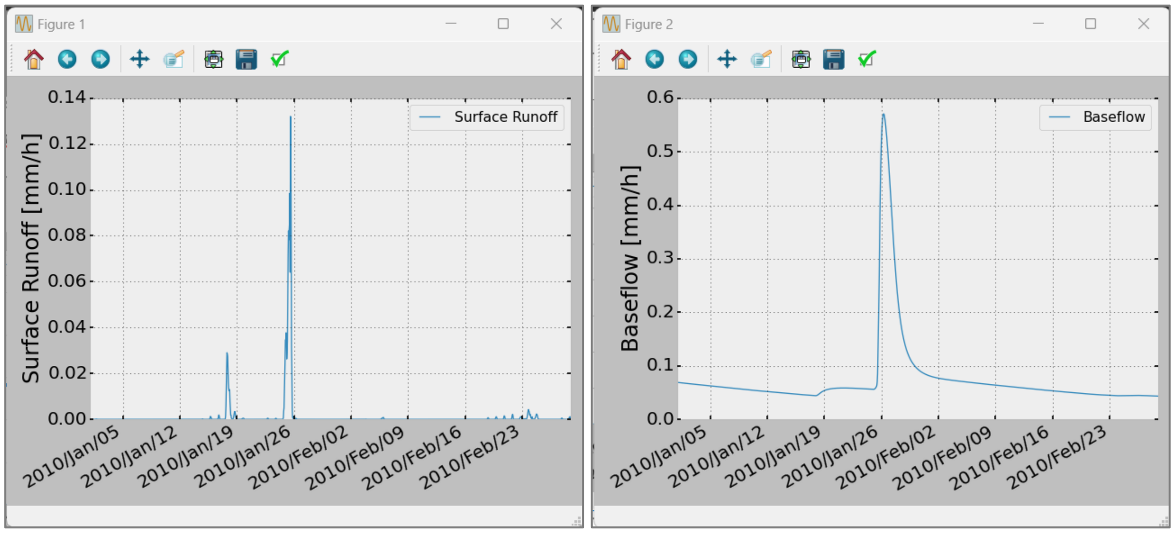
(4) Model Source
**Executable Program**: File ent\VIC5_HPC.exe
**Source Code**: Directory
**Model Dependency**: Directory
**Remote Source**: String

(5) Output Datasets
**File Position**: String 6
**Output Name**: String Surface Runoff
**File Position**: String 7
**Output Name**: String Baseflow

Convert_to_Dataset
Boolean ✓

(6) Node Configuration
DataSet_Class
Ready_List
WD_Path

*Figure 78. Complete workflow with msmShowChart module boxes connected to the first and second output ports of HPC module*



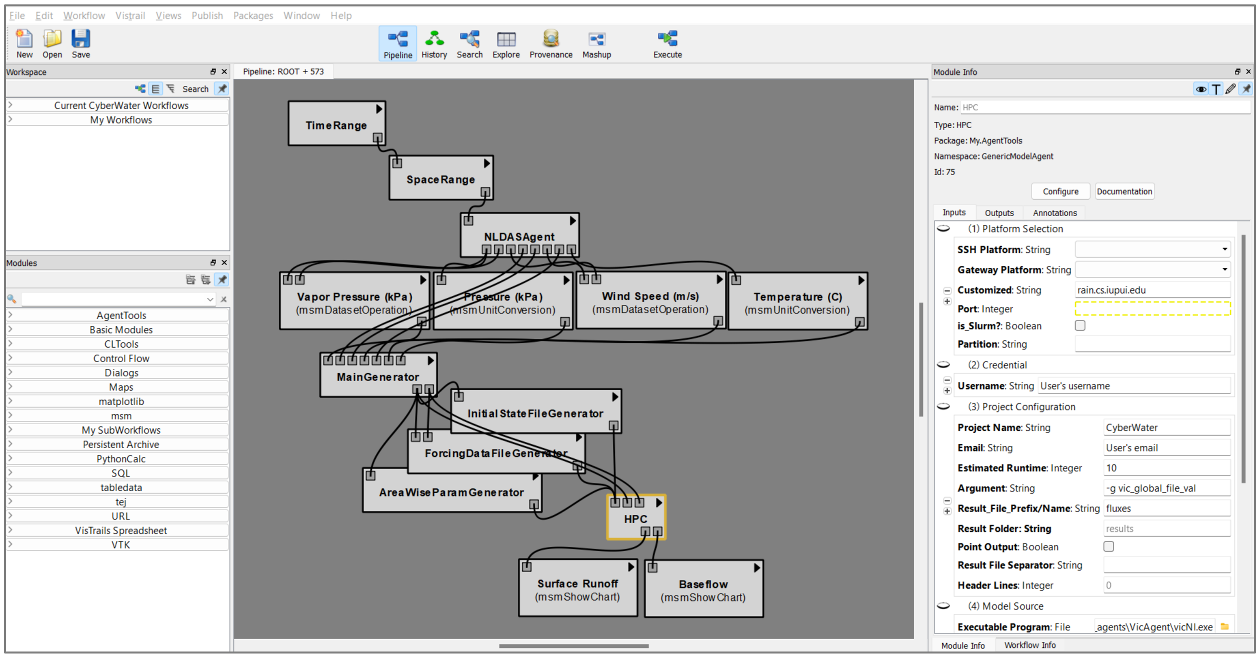
*(a)* *(b)*

*Figure 79. (a) The resulting (a) surface runoff and (b) baseflow from the VIC5 simulation of the West-Branch Susquehanna in PA running on the Bridges2 high-performance computing platform.*

### 6.2.2 *Connection with user's own added (customized) platform/server.*

- Configure *HPC* module by following the instruction stated in Section 6.2.1, keep other entries the same except the selection for *(1) Platform Selection* and *(2) credential*:
  - (1) Platform Selection:
    Click '-' button to delete the previous selection, and then click the column '(1) Platform Selection' to type-in the IP address or the domain name of the server or personal machine to connect.
    - Customized: [The IP address or domain name of user's added server]. E.g., rain.cs.iupui.edu or htc.sam.pitt.edu
    - Slurm-based?: If the added machine (e.g., HPC platform) uses Slurm Workload Manager, set the value of this variable to be true by checking it. For the server, rain.cs.iupui.edu, which is not slurm-based, so keep the checkbox blank.
  - (2) Credential:
    - Username: [user's corresponding username for the customized machine]
    - Password: [user's corresponding password for the customized machine]

After these changes, the workflow should look like the image shown in Figure 80.



Name: HPC
Type: HPC
Package: My.AgentTools
Namespace: GenericModelAgent
Id: 75

(1) Platform Selection

| | |
|---|---|
| **SSH Platform**: String | |
| **Gateway Platform**: String | |
| **Customized**: String | rain.cs.iupui.edu |
| **Port**: Integer | |
| **is_Slurm?**: Boolean | ☐ |
| **Partition**: String | |

(2) Credential

| | |
|---|---|
| **Username**: String | User's username |

(3) Project Configuration

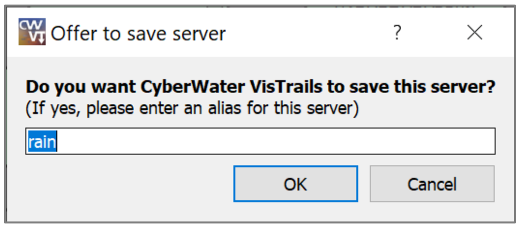
| | |
|---|---|
| **Project Name**: String | CyberWater |
| **Email**: String | User's email |
| **Estimated Runtime**: Integer | 10 |
| **Argument**: String | -g vic_global_file_val |
| **Result_File_Prefix/Name**: String | fluxes |
| **Result Folder: String** | results |
| **Point Output**: Boolean | ☐ |
| **Result File Separator**: String | |
| **Header Lines**: Integer | 0 |

(4) Model Source

| | |
|---|---|
| **Executable Program**: File | _agents\VicAgent\vicNI.exe |

*Figure 80. Workflow with HPC module configuration by entering 'rain.cs.iupui.edu' as the domain name of the customized server.*

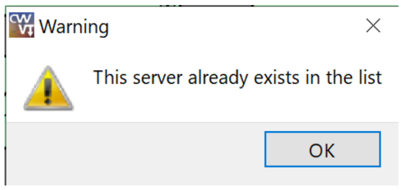
While running the workflow, a query dialog should pop up to ask the user whether to save this server information with a user's provided alias if the user attempts to keep this server for future use, as Figure 81 depicted below. If the provided name of the server already exists, a warning dialog will show up as shown in Figure 82, otherwise, this server information will be saved in the server list. When the user wants to re-connect this customized server, they don't need to re-enter the server information; instead, they can find the alias of the server as an option in the drop list of 'SSH Platform' when reloading the package of 'AgentTools' or reopening the CyberWater VisTrials.


Offer to save server

Do you want CyberWater VisTrails to save this server?
(If yes, please enter an alias for this server)

rain

*Figure 83. The query dialog asks the user whether to save this current server.*


Warning

This server already exists in the list

*Figure 84. The warning message dialog to indicate the server name already exists and the user doesn't need to add it again.*

# 7. WRF-Hydro

The following exercises demonstrate advanced ways to use the Generic Module Agent Toolkit and the High-Performance Computing module. If the user would like additional practice with these tools before engaging with this section, they are invited to work through Sections 4 and 5 to get an overview of the Generic Module Agent Toolkit.

## 7.1 Executing WRF-Hydro for Small-Scale Hydrological Studies.

This section demonstrates how the CyberWater framework can be used to build a complete hydrologic simulation using the WRF-Hydro model. This capability is demonstrated in executing a WRFHydro simulation locally and this demonstration illustrate how the CyberWater framework can be used to construct self-describing, reusable model simulations by leveraging modern cyberinfrastructure and modeling techniques.

## 7.2 Model Description

WRF-Hydro is a terrestrial hydrologic process model that is developed and maintained by the National Center of Atmospheric Research (NCAR). It is designed to provide land surface hydrology and energy states at high spatial resolutions using a physics-based conceptual approaches and is the basis for the NOAA National Water model (NWM). Moreover, the NWM is a special configuration of the WRF-Hydro open-source community model to simulate observed and forecasted streamflow over the continental United States. A detailed description of the physics of the model are thoroughly outlined in the official WRFHydro documentation.

The NWM configuration of WRF-Hydro is typically used for computationally intensive modeling systems at large spatial and temporal scales that can require hundreds or thousands of processors. However, research at local and regional spatial scales provides valuable insight into the physical process representations and parameterizations that can inform larger scale activities. The WRF-Hydro examples outlined in this case study explore how to build such modeling workflows using the CyberWater framework.

The configuration of WRF-Hydro that we'll be using in this case study consists of the following files and datasets:

### 7.2.1 *Domain*

WRF-Hydro DOMAIN datasets consist of geospatial and hydrographic input files that are necessary to define the geographic extent and static parameterization of the model. The CUAHSI Domain Subsetter web application is used to collect these files for our study area via a graphical map interface. This enables us to obtain the same model domain and parameterization that is used in the National Water Model. A description of the files that are obtained from the CUAHSI Domain Subsetter are described in the table below.

| Dataset | Description |
|---|---|
| Fulldom_hires.nc | High resolution full domain file. Includes all fields specified on the routing grid. |
| geo_em.d0x.nc | Defines the land surface model grid and relevant geospatial data. |
| GWBUCKPARM.nc | Groundwater parameter table containing bucket model parameters for each basin |
| hydro2dtbl.nc | Spatially distributed parameter table for lateral flow routing within WRF-Hydro. |
| Route_Link.nc | Channel reach parameters (ComID, gage ID, bottom width, slope, roughness, order, etc.) |
| soil_properties.nc | Spatially distributed land surface model parameters that allow users to specify parameters on the model grid rather than as a single value or function of soil or land cover type. |
| spatialweights.nc | Weights to map between the land surface grid and the pre-defined groundwater basin boundaries. |
| wrfinput_d0x.nc | Initial conditions for the land surface, such as soil moisture, soil temperature, and snow states |

### 7.2.2 *Configuration*

| File Name | Description |
|---|---|
| hydro.namelist | Settings for operating all of the routing components of the WRF-Hydro system. |
| namelist.hrldas | Specifies the model options to be used in the land surface model, e.g. NoahMP. |

### 7.2.3 *Forcing*

WRF-Hydro accepts forcing input data either in an hourly or minute time step . Currently, the WRFHydro team supports pre-processing tools for NLDAS, GLDAS, HRRR, MRMS, GFS, RAP, and WRF meteorological data. In this example, we'll be using NLDAS meteorological input data, however, the process outlined in the document can be followed, with slight modification, for other supported data products. In any case, the required input variables are listed in the table below:

| Variable | Description | Units |
|---|---|---|
| SWDOWN | Incoming shortwave radiation | W/m2 |
| LWDOWN | Incoming longwave radiation | W/m2 |
| Q2D | Specific humidity | kg/kg |
| T2D | Air temperature | K |
| PSFC | Surface pressure | Pa |
| U2D | Near surface wind in the u - component | m/s |
| V2D | Near surface wind in the v-component | m/s |
| RAINRATE | Liquid water precipitation rate | mm/s or kg/m2/s |

### 7.2.4 *Source Code*

While all model source code is publicly available online, we will use several Docker containers to enable model execution on the Windows OS without the need to compile anything. Users are encouraged to compile the WRFHydro model in their free time using the documentation provided by NCAR.

| Docker Container | Description |
|---|---|
| cuahsi/wrfhydro-regrid | Performs forcing pre-processing operations using the ESMF regridding tools. |
| cuahsi/wrfhydro-nwm | Executes the WRF-Hydro model with user provided DOMAIN, FORCING, and namelists. |
| cuahsi/wrfhydro-postprocess | Performs post-processing (data cleaning) operations to make output visualization within CyberWater seamless. |

## 7.3 Executing WRF-Hydro in CyberWater

This section illustrates how to build and configure a WRF-Hydro simulation using Cyberwater and execute a simulation on a local computer using the Windows operating system. Since the WRF-Hydro model is designed and tested on Linux, we'll use Docker and the Windows Subsystem for Linux to support these actions on Windows.

## 7.4 Install and Configure Docker on Windows

1. If the user didn't install the Docker Desktop with CyberWater package, then they can go back and install it using the same installer of CyberWater. Also, to complete the installation, follow instructions in Section 1.2.3. If user install the Docker Desktop successfully, then they can open the CyberWater, and a "Docker" tab will be shown on the top of the interface, and user can "Enable Docker" and "Disable Docker" to start the Docker Engine or close the Docker Engine running in the background.

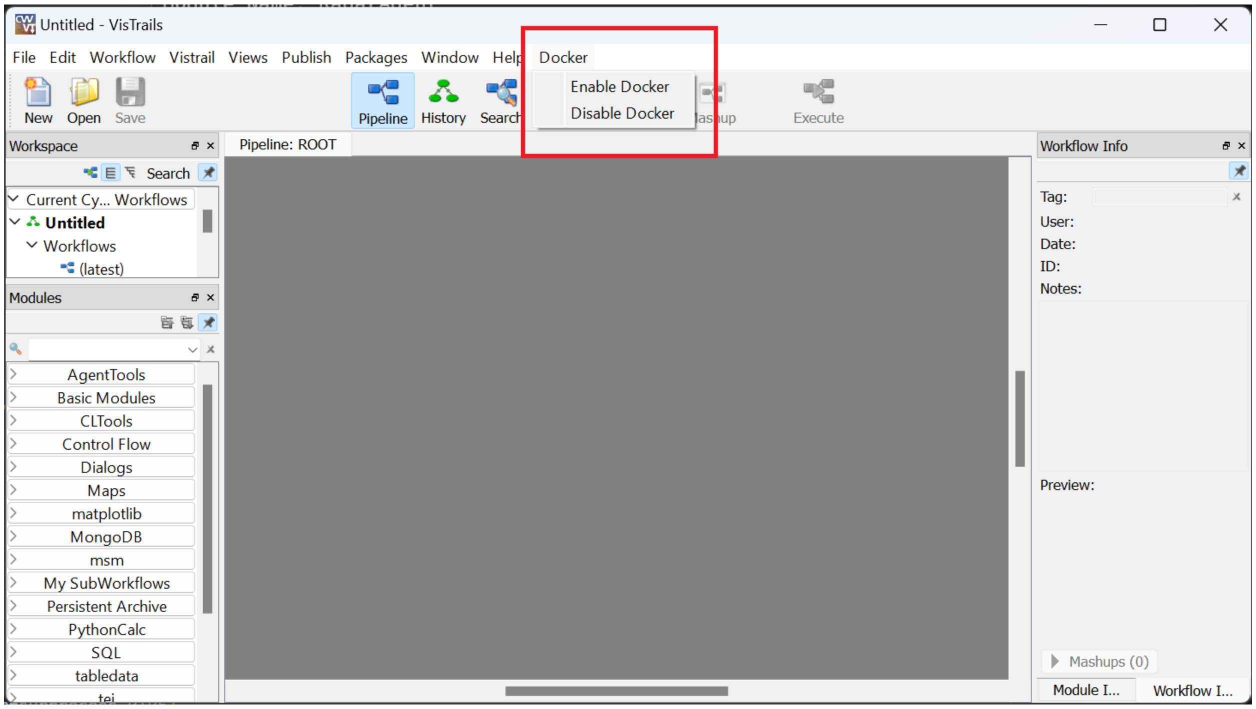


*Figure 85. A CyberWater interface with Docker Desktop installed successfully.*

2. Launch the Windows Subsystem for Linux by searching for "wsl" in the Start Menu.

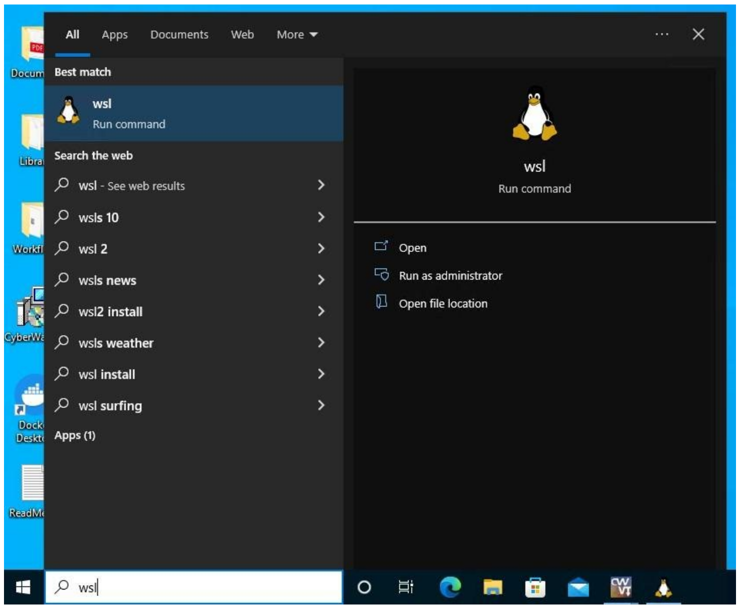


*Figure 86. How to open 'wsl' from Start Menu*

3. Within the WSL terminal, list all Docker images that exist on your system using `docker images` command. If this is your first-time using Docker, then the output of this command will be empty like shown below.

```
root@win10-workshop-:/mnt/c/Windows/system32# docker images

REPOSITORY          TAG     IMAGE ID     CREATED     SIZE
```

4. Test that Docker is configured correctly by downloading and executing a "helloworld" test image. Do so by issuing the following commands:

```
root@win10-workshop-:/mnt/c/Windows/system32# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete Digest: sha256:18a657d0cc1c7d0678a3fbea8b7eb4918bba25968d3e1b0adebfa71caddbc346
Status: Downloaded newer image for hello-world:latest docker.io/library/hello-world:latest
```

5. Now if we list this Docker images on our system, we should see one called "hello-world"

```
root@win10-workshop-:/mnt/c/Windows/system32# docker images

REPOSITORY          TAG     IMAGE ID     CREATED      SIZE hello-world         latest    feb5d9fea6a5   12 months ago   13.3kB
```

6. Execute this image by issuing the command `docker run hello-world:latest`. Docker is properly configured if the output of this command looks like the following:

```
root@win10-workshop-:/mnt/c/Windows/system32# docker run hello-world:latest


Hello from Docker!
This message shows that your installation appears to be working correctly.


To generate this message, Docker took the following steps:
```

```
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.    (amd64)
3. The Docker daemon created a new container from that image which runs the     executable that produces the output you are
   currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it     to your terminal.


To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash


Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/
```

7. Open the CyberWater. If the "Docker" is properly configured if the output of this command looks like the following:

## 7.5 Executing WRF-Hydro Locally

This tutorial outlines the process of constructing and executing a WRF-Hydro simulation on a local Windows computer using the CyberWater modeling framework.

### 7.5.1 *Study Area*

We will be collecting data and executing a simulation of WRF-Hydro for a small watershed located in eastern Iowa at the outlet of the Clear Creek river near Iowa City as shown in Figure 78. This watershed is approximately 270 km$^2$ and contains characteristics that are typical in many midwestern watersheds. This is a highly studied watershed (part of the Critical Zone Observatory network) because of its agricultural use, highly erodible soils, steep slopes, humid climate, and increased urbanization. To learn more about this watershed, see the CZO website.

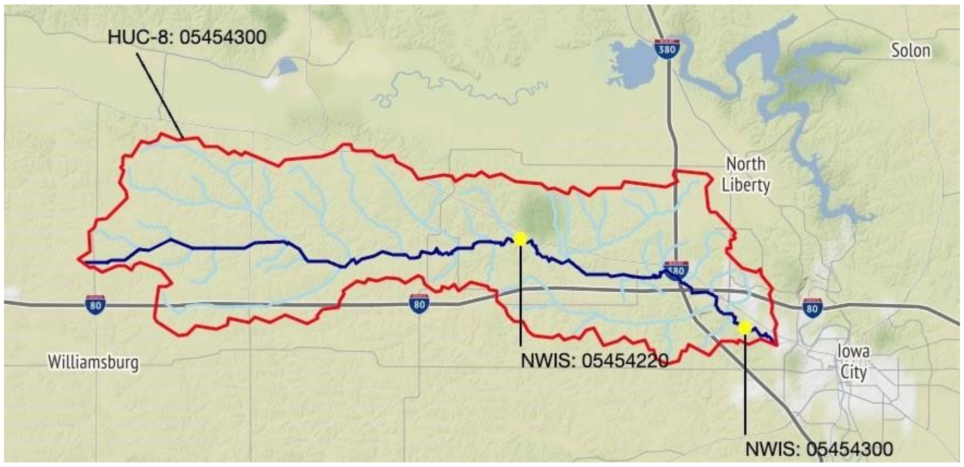


*Figure 87. The study area of WRF-Hydro model*

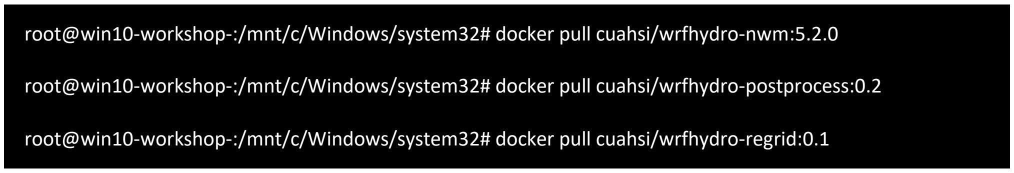
### 7.5.2 *Download Docker Images*

1. Launch the WSL terminal.

2. Download three images that will be used in this tutorial using the WSL terminal.

   i. `docker pull cuahsi/wrfhydro-nwm:5.2.0`

```
ii.   docker pull cuahsi/wrfhydro-regrid:0.1
iii.  docker pull cuahsi/wrfhydro-postprocess:0.2
```

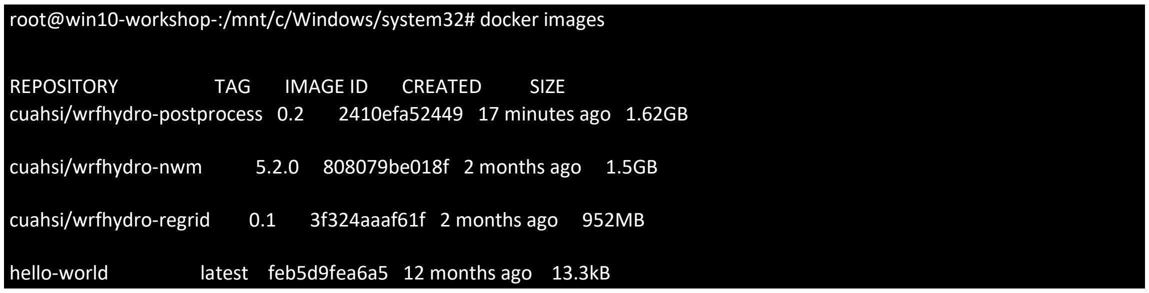


```
root@win10-workshop-:/mnt/c/Windows/system32# docker pull cuahsi/wrfhydro-nwm:5.2.0

root@win10-workshop-:/mnt/c/Windows/system32# docker pull cuahsi/wrfhydro-postprocess:0.2

root@win10-workshop-:/mnt/c/Windows/system32# docker pull cuahsi/wrfhydro-regrid:0.1
```

After issuing the commands above, we should see the following when listing the Docker images on the system (`docker images`):



```
root@win10-workshop-:/mnt/c/Windows/system32# docker images


REPOSITORY                  TAG     IMAGE ID       CREATED         SIZE
cuahsi/wrfhydro-postprocess  0.2     2410efa52449   17 minutes ago   1.62GB

cuahsi/wrfhydro-nwm          5.2.0   808079be018f   2 months ago     1.5GB

cuahsi/wrfhydro-regrid       0.1     3f324aaaf61f   2 months ago     952MB

hello-world                  latest  feb5d9fea6a5   12 months ago    13.3kB
```

Each of these docker images will be used within our simulation to perform forcing data pre-processing, model execution, or output data post-processing.

### 7.5.3 *Building the Model Workflow*

1. The following explanation assumes that the user will download <Examples Data\WRF-Hydro> folder from Hydroshare and save it in the following path *"C:\temp\CYBERWATER\my-simulation\WRF-Hydro"*.

2. Add a **TimeRange** component to the workspace, this is used to define the temporal range for our simulation. Configure it with the following parameters:
   - Inputs

| Variable | Value |
|---|---|
| 01_timeini | 2017/01/01 00:00:00 |
| 02_timeend | 2017/02/01 00:00:00 |

3. Add an **NLDASAgent** component to the workspace. This component will collect NLDAS meteorological forcing data for the time range specified in the previous step. You'll need to create a free account on earthdata.nasa.gov to download data. Once you've created an EarthData account, configure the **NLDASAgent** with the following parameters:
   - Inputs

| Variable | Value |
|---|---|
| file_type | GRIB |

| | |
|---|---|
| username | <your earthdata username> |
| password | <your earthdata password> |
| variableName | All |

Note: User can download the forcing data directly from Hydroshare to save some time. These pre-downloaded data files are compressed and available at:
<Examples Data/Forcing Data/NLDAS/All>
Unzip this file and put the folder "All" inside the "downloads" folder at:
<C:\CyberWater\VisTrails\vistrails\packages\msm\utils\uploaded\user_data_agents\NLDASAgent\downloads>
Also note that if you followed the steps in Section 4.5, you already copied all the folders found in the downloaded "NLDAS" folder and pasted them in the "downloads" folder so you will find the folder "All" already exists.

- Outputs

| Visibility | Variable |
|---|---|
| 👁 | *All |

*Enable the "All" output port by clicking the icon next to the specified value in the table.

4. Connect the TimeRange and NLDASAgent components together:

| From Component | From Port | To Component | To Port |
|---|---|---|---|
| TimeRange | Timerange | NLDASAgent | 01_case_study |

5. Add a **MainGenerator** component to the workspace. This component will create a working directory for the simulation that will be used by future components. Configure it with the following parameters:
   - Inputs

| Variable | Value |
|---|---|
| 01_Path | <directory to save files> E.g. "*C:\temp\my-simulation*" |
| 03_Override? | False (unchecked) |
| 👁 | *Dataset_01 |

*Enable the Dataset_01 input port by clicking the icon next to the specified value in the table.

6. Connect the **NLDASAgent** and **MainGenerator** components together:

| From Component | From Port | To Component | To Port |
|---|---|---|---|
| NLDASAgent | All | MainGenerator | Dataset_01 |

7. Add the **ForcingDataFileGenerator** component to the workspace. Configure it with the following parameters:

| Variable | Value |
|---|---|
| Forcing_Folder_Name | <Folder name to save forcing data> E.g. *"INPUT"* |

The *Forcing_Folder_Name* is a directory that will be created within the path specified in the **MainGenerator** component. This directory will contain NLDAS data that will be downloaded. A good name for this directory might be "NLDAS-raw".

8. Add the **CUAHSISubsetter**[1] component to the workspace. The component will collect WRF-Hydro static domain data from the subset.cuahsi.org. Configure it with the following parameters:
   - Inputs

| Variable | Value |
|---|---|
| Folder_Name | <Folder name to save domain files> E.g. *"DOMAIN"* |
| llat | 192814.30490000173 |
| ulat | 203185.30050000036 |
| llon | 399965.7248999998 |
| ulon | 436818.2032000007 |

The CUAHSI Model Domain Subsetter is a collaborative effort for preparing, publishing, and sharing subsets of the Continental U.S. (CONUS) model input datasets at small and regional watershed scales. This is made possible with a combination of modern cyberinfrastructure techniques and state-of-the-science modeling tools. Researchers and educators can access to subsets of these large-scale, intensive efforts, that would otherwise require extensive computational and human resources. Moreover, this service provides a mechanism for the water science community to align research efforts with the large scale modeling initiatives that are being undertaken by modelers, national laboratories, and federal institutions. We will use the CUAHSI Model Domain Subsetter to collect static domain data and parameters that are used by the operational National Water Model (a special configuration of WRF-Hydro) for the Clear Creek watershed.

In the *Folder_Name* variable, we can specify a directory name for our static domain data files to be saved. A good name to use would be DOMAIN. We must also provide the latitude and longitude bounding box of the watershed area. This bounding box (upper and lower latitiude, upper and lower longitude) are used to subset static domain parameters for executing WRF-Hydro configured as the National Water Model. It is important to note that these coordinates must be provided in a WRF-Hydro specific Lambert Conformal Conic projection.

9. Connect the outputs of the **MainGenerator** to the input ports of the **CUAHSI Subsetter** and **ForcingDataFileGenerator** components as shown in Figure 88.

| From Component | From Port | To Component | To Port |
|---|---|---|---|
| MainGenerator | WD_Path | CUAHSISubsetter | WD_Path |

[1] Located in AgentTools>Subsetter>CUAHSISubsetter

| | | | |
|---|---|---|---|
| MainGenerator | WD_Path | ForcingDataFileGenerator | WD_Path |
| MainGenerator | DataSet_Class | ForcingDataFileGenerator | DataSet_Class |

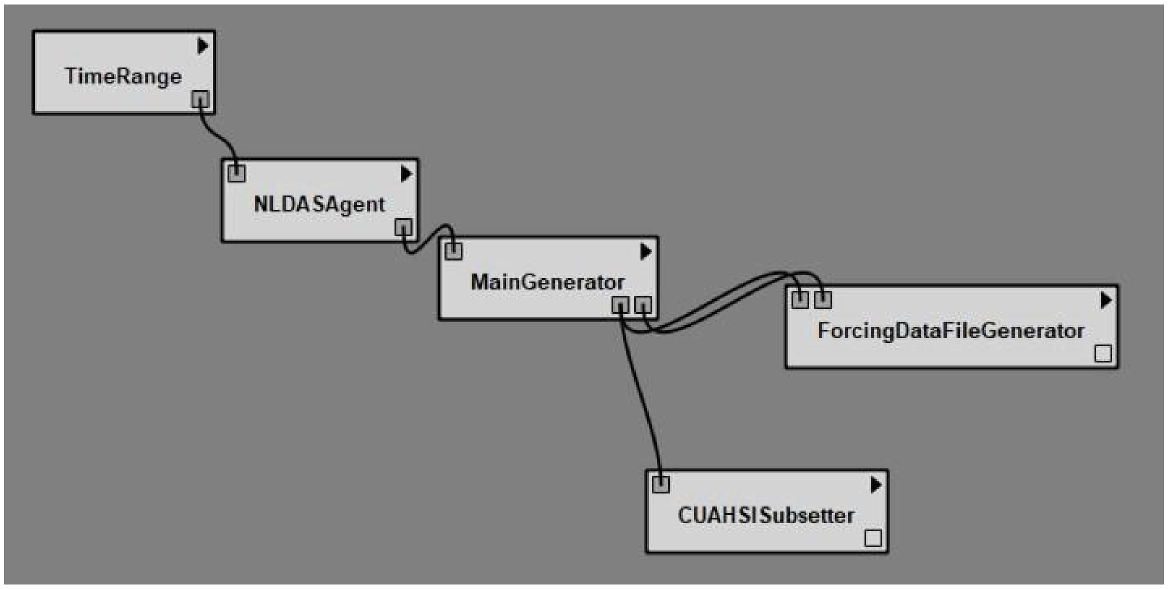


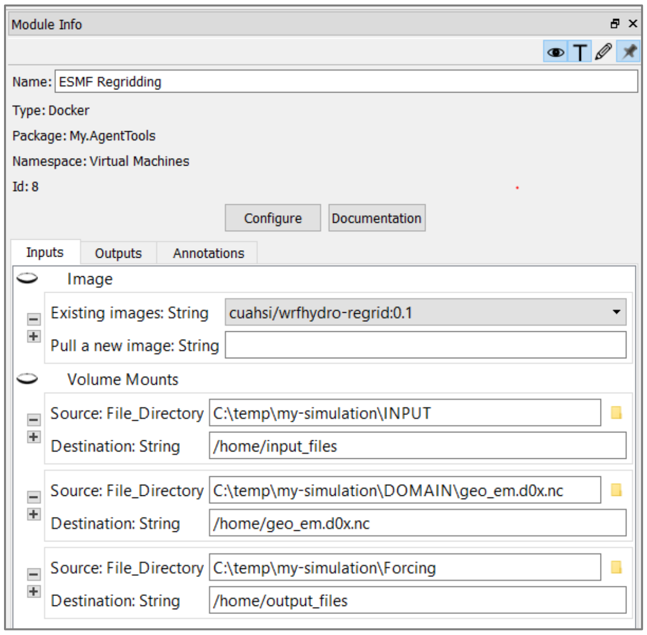
*Figure 88. The workflow connecting the MainGenerateor and CUAHSISubsetter module*

10. Add a **Docker** component to the workspace (`AgentTools/Virtual Machines/Docker`). Make sure you have started the Docker Engine by clicking the "Enable Docker" option under the "Docker" tab on the top of the interface, otherwise the "Virtual Machines" tab will not be shown under the "AgentTools" package in the input panel. It might take 1~2 minutes start Docker Engine, until you see the message in CyberWater terminal saying Docker is running, which signs the Docker Engine has been started, and then turn to the input panel, the "AgentTools" package will be auto-updated and display the "Virtual Machines" tab. You can "Disable Docker" when you finish the WRF-Hydro model simulation in CyberWater to avoid Docker is still running in the background to impact your machine performance. This generic component can be used to execute any Docker container. We'll configure it to execute the forcing data regridding operations. To do so, configure it with the following parameters:

- Inputs (shown in Figure 89):
  First of all, rename the module to be "ESMF Regridding" instead of Docker from the Module info in the upper right of the screen. Click the (+) button next to Volume Mounts to add three new items, then fill the source and destination paths with the values listed in the table below.

| **Variable** | **Value** |
|---|---|
| Image/Existing images | cuahsi/wrfhydro-regrid:0.1 |
| | |
| Volume Mounts/Source | <directory to input forcing> |

| | |
|---|---|
| | E.g. "*C:\temp\my-simulation\INPUT*" |
| Volume Mounts/Destination | /home/input_files |
| | |
| Volume Mounts/Source | <directory to domain geo_em.d0x.nc><br>E.g. "*C:\temp\my-simulation\DOMAIN\geo_em.d0x.nc*" |
| Volume Mounts/Destination | /home/geo_em.d0x.nc |
| | |
| Volume Mounts/Source | <directory to save output files><br>E.g. "*C:\temp\my-simulation\Forcing*" |
| Volume Mounts/Destination | /home/output_files |



Name: ESMF Regridding
Type: Docker
Package: My.AgentTools
Namespace: Virtual Machines
Id: 8

Image
Existing images: String cuahsi/wrfhydro-regrid:0.1
Pull a new image: String

Volume Mounts
Source: File_Directory C:\temp\my-simulation\INPUT
Destination: String /home/input_files
Source: File_Directory C:\temp\my-simulation\DOMAIN\geo_em.d0x.nc
Destination: String /home/geo_em.d0x.nc
Source: File_Directory C:\temp\my-simulation\Forcing
Destination: String /home/output_files

*Figure 89. The input panel of ESMF Regridding module box*

- Outputs

| Visibility | Variable |
|---|---|
| 👁 | *DockerImage |

*Enable the DockerImage output port by clicking the icon next to the specified value in the table.

11. Add a **RunModuleAgent** component to the workspace, give it the name "Model Prep". This component will trigger the execution of the model preparation phase of our workflow. Rename the module to be "Model Preparation" instead of RunModuleAgent from the Module info in the upper right of the screen. Then configure it with the following parameters:

| Variable | Value |
|---|---|

| | |
|---|---|
| 03_Model_executable/arg | -f /home/input_files -g /home/geo_em.d0x.nc -m NLDAS -i bilinear |
| 04_Results_format/Results_Folder | Forcing |
| Convert_to_Dataset | False (Unchecked) |
| | |
| 👁 | *Docker |

*Enable the Docker input port by clicking the icon next to the specified value in the table.

- Outputs

| Visibility | Variable |
|---|---|
| 👁 | *Ready |

Note: the *Results_format/Results_Folder* variable should match the folder name mounted to /home/output_files that was provided in the previous step, e.g. NLDAS_regrid.

12. Connect the outputs of the **CUAHSISubsetter**, **ForcingDataFileGenerator**, **MainGenerator**, and **ESMF Regridding** components to the Model Preparation component as shown in Figure 82.

| From Component | From Port | To Component | To Port |
|---|---|---|---|
| CUAHSISubsetter | Ready | Model Preparation | Ready_List |
| | | | |
| ForcingDataFileGenerator | Ready | Model Preparation | Ready_List |
| | | | |
| ESMF Regridding | DockerImage | Model Preparation | Docker |
| | | | |
| MainGenerator | WD_Path | Model Preparation | WD_Path |
| MainGenerator | Dataset_class | Model Preparation | Dataset_class |

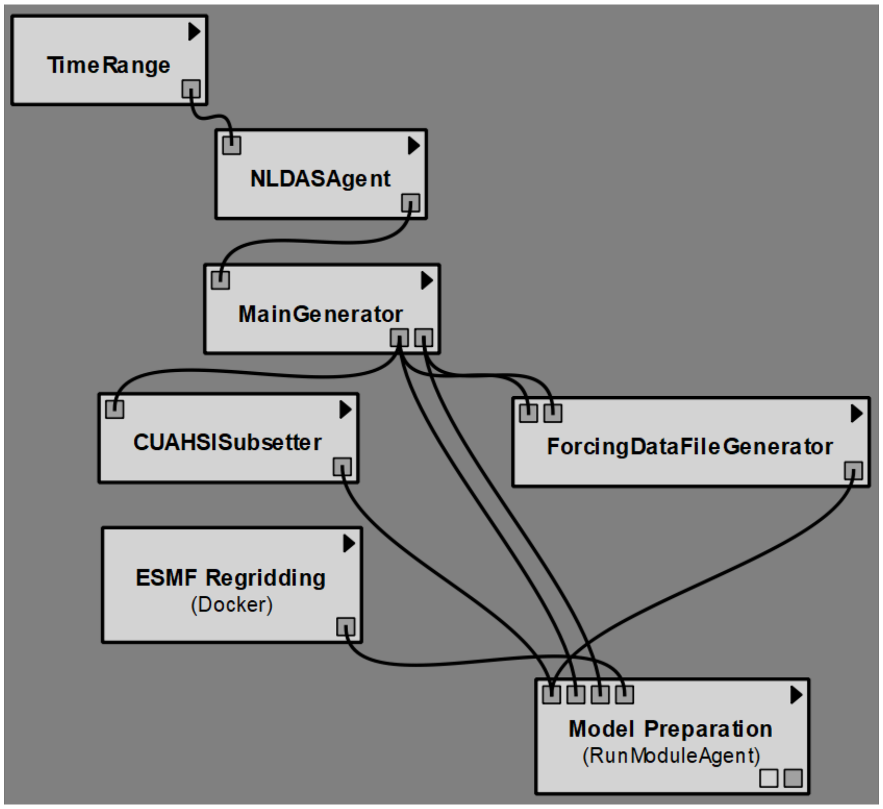


*Figure 89. The workflow connecting ESMF Regridding Docker module and RunModuleAgent*

13. Add a Docker component to the workspace (`AgentTools/Virtual Machines/Docker`). Configure it with the following parameters.
    - Inputs (shown in Figure 90)

    Rename the module to be "WRF-Hydro Execution" instead of Docker from the Module info in the upper right of the screen. Click the (+) button next to Volume Mounts to add five new items, then fill the source and destination paths with the values listed in the table below.

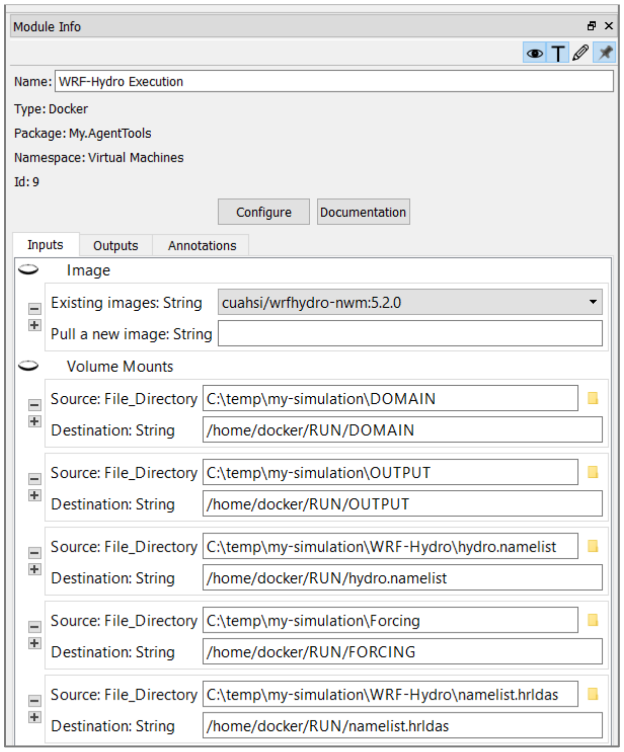
| **Variable** | **Value** |
|---|---|
| Image/Existing images | cuahsi/wrfhydro-nwm:5.2.0 |
| | |
| Volume Mounts/Source | <directory to domain data><br>E.g. "*C:\temp\my-simulation\DOMAIN*" |
| Volume Mounts/Destination | /home/docker/RUN/DOMAIN |
| | |
| Volume Mounts/Source | <directory to save output data><br>E.g. "*C:\temp\my-simulation\OUTPUT*" |
| Volume Mounts/Destination | /home/docker/RUN/OUTPUT |
| | |

| | |
|---|---|
| Volume Mounts/Source | <directory to forcing data><br>E.g. "*C:\temp\my-simulation\Forcing*" |
| Volume Mounts/Destination | /home/docker/RUN/FORCING |
| | |
| Volume Mounts/Source | <path to hydro.namelist><br>E.g. "*C:\temp\my-simulation\WRF-Hydro\hydro.namelist*" |
| Volume Mounts/Destination | /home/docker/RUN/hydro.namelist |
| | |
| Volume Mounts/Source | <path to hrldas namelist><br>E.g. "*C:\temp\my-simulation\WRF-Hydro\namelist.hrldas*" |
| Volume Mounts/Destination | /home/docker/RUN/namelist.hrldas |

- Outputs

| **Visibility** | **Variable** |
|---|---|
| 👁 | *DockerImage |

*Enable the DockerImage output port by clicking the icon next to the specified value in the table.


Module Info

Name: WRF-Hydro Execution
Type: Docker
Package: My.AgentTools
Namespace: Virtual Machines
Id: 9

Image
Existing images: String cuahsi/wrfhydro-nwm:5.2.0
Pull a new image: String

Volume Mounts
Source: File_Directory C:\temp\my-simulation\DOMAIN
Destination: String /home/docker/RUN/DOMAIN
Source: File_Directory C:\temp\my-simulation\OUTPUT
Destination: String /home/docker/RUN/OUTPUT
Source: File_Directory C:\temp\my-simulation\WRF-Hydro\hydro.namelist
Destination: String /home/docker/RUN/hydro.namelist
Source: File_Directory C:\temp\my-simulation\Forcing
Destination: String /home/docker/RUN/FORCING
Source: File_Directory C:\temp\my-simulation\WRF-Hydro\namelist.hrldas
Destination: String /home/docker/RUN/namelist.hrldas

*Figure 90. The input panel ofWRF-Hydro Execution module box*

14. Add a **RunModuleAgent** component to the workspace and give it the name “Model Execution”. This component will trigger the execution phase of our workflow. Configure it with the following parameters:
    - Inputs

| Variable | Value |
|---|---|
| 04_Results_format/Results_Folder | OUTPUT |
| Convert_to_Dataset | False (Unchecked) |
| | |
| 👁 | *Docker |

*Enable the Docker input port by clicking the icon next to the specified value in the table.

- Outputs

| Visibility | Variable |
|---|---|
| 👁 | *Ready |

*Enable the Ready output port by clicking the icon next to the specified value in the table.

15. Connect the outputs of the **Execute WRF-Hydro**, **MainGenerator**, and **Model Prep** components to the **Model Execution** component as shown in Figure 91.

| From Component | From Port | To Component | To Port |
|---|---|---|---|
| WRF-Hydro Execution | DockerImage | Model Execution | Docker |
| | | | |
| Model Preparation | Ready | Model Execution | Ready_List |
| | | | |
| MainGenerator | WD_Path | Model Execution | WD_Path |
| MainGenerator | Dataset_class | Model Execution | Dataset_class |

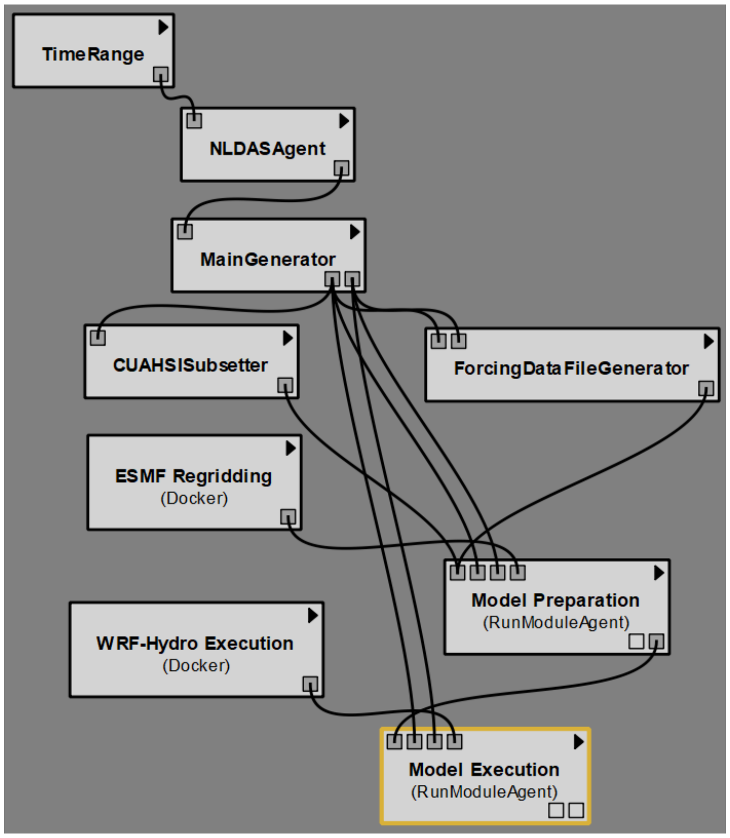


*Figure 91. The workflow connecting WRF-Hydro Execution Docker module and Model Execution RunModuleAgent*

16. Add a **Docker** component to the workspace (`AgentTools/Virtual Machines/Docker`). Make sure you have started the Docker Engine by clicking the "Enable Docker" option under the "Docker" tab on the top of the interface, otherwise the "Virtual Machines" tab will not be shown under the "AgentTools" package in the input panel. It might take 1~2 minutes start Docker Engine, until you see the message in CyberWater terminal saying Docker is running, which signs the Docker Engine has been started, and then turn to the input panel, the "AgentTools" package will be auto-updated and display the "Virtual Machines" tab. You can "Disable Docker" when you finish the WRF-Hydro model simulation in CyberWater to avoid Docker is still running in the background to impact your machine performance. This generic component can be used to execute any Docker container. We'll configure it to execute the forcing data regridding operations. To do so, configure it with the following parameters:Configure it with the following parameters.

    - Inputs (shown in Figure 92)

    Rename the module to be "Result Post-processing" instead of Docker from the Module info in the upper right of the screen. Click the (+) button next to Volume Mounts to add three new items, then fill the source and destination paths with the values listed in the table below.

| **Variable** | **Value** |
|---|---|
| Existing images | cuahsi/wrfhydro-postprocessing:0.2 |

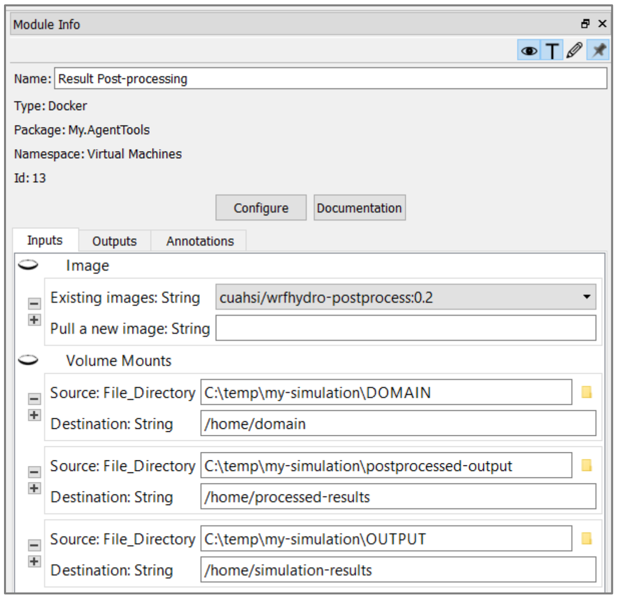
| | |
|---|---|
| Volume Mounts/Source | <directory to domain data><br>E.g. *“C:\temp\my-simulation\DOMAIN”* |
| Volume Mounts/Destination | /home/domain |
| | |
| Volume Mounts/Source | <directory to save processed output data><br>E.g. *“C:\temp\my-simulation\postprocessed-output“* |
| Volume Mounts/Destination | /home/processed-results |
| | |
| Volume Mounts/Source | <directory to wrfhydro results><br>E.g. *“C:\temp\my-simulation\OUTPUT”* |
| Volume Mounts/Destination | /home/simulation-results |

- Outputs

Select the “Outputs” tab and enable the “DockerImage” output port.

| **Visibility** | **Variable** |
|---|---|
| 👁 | *DockerImage |

*Enable the DockerImage output port by clicking the icon next to the specified value in the table.



Name: Result Post-processing
Type: Docker
Package: My.AgentTools
Namespace: Virtual Machines
Id: 13

Image
Existing images: String cuahsi/wrfhydro-postprocess:0.2
Pull a new image: String

Volume Mounts
Source: File_Directory C:\temp\my-simulation\DOMAIN
Destination: String /home/domain
Source: File_Directory C:\temp\my-simulation\postprocessed-output
Destination: String /home/processed-results
Source: File_Directory C:\temp\my-simulation\OUTPUT
Destination: String /home/simulation-results

*Figure 92. The input panel of Result Post-processing module box*

17. Add a **RunModuleAgent** component to the workspace and give it the name “Model Postprocessing”. This component will trigger the post processing phase of our workflow. Configure it with the following parameters:

- Inputs

| Variable | Value |
|---|---|
| 04_Results_format/Results_Folder | postprocessed-output |
| Convert_to_Dataset | True (checked) |
| | |
| 👁 | *Docker |

*Enable the Docker input port by clicking the icon next to the specified value in the table.
Note: the *04_Results_format/Results_Folder* variable should match the folder name mounted to /home/processed-resultsthat was provided in the previous step, e.g. postprocessed-output.

18. Connect the outputs of the WRF-Hydro Post Processing to the new RunModuleAgent as shown in Figure 93.

| From Component | From Port | To Component | To Port |
|---|---|---|---|
| Result Post-processing | DockerImage | Model Post-processing | Docker |
| | | | |
| Model Execution | Ready | Model Post-processing | Ready_List |
| | | | |
| MainGenerator | WD_Path | Model Post-processing | WD_Path |
| MainGenerator | Dataset_class | Model Post-processing | Dataset_class |

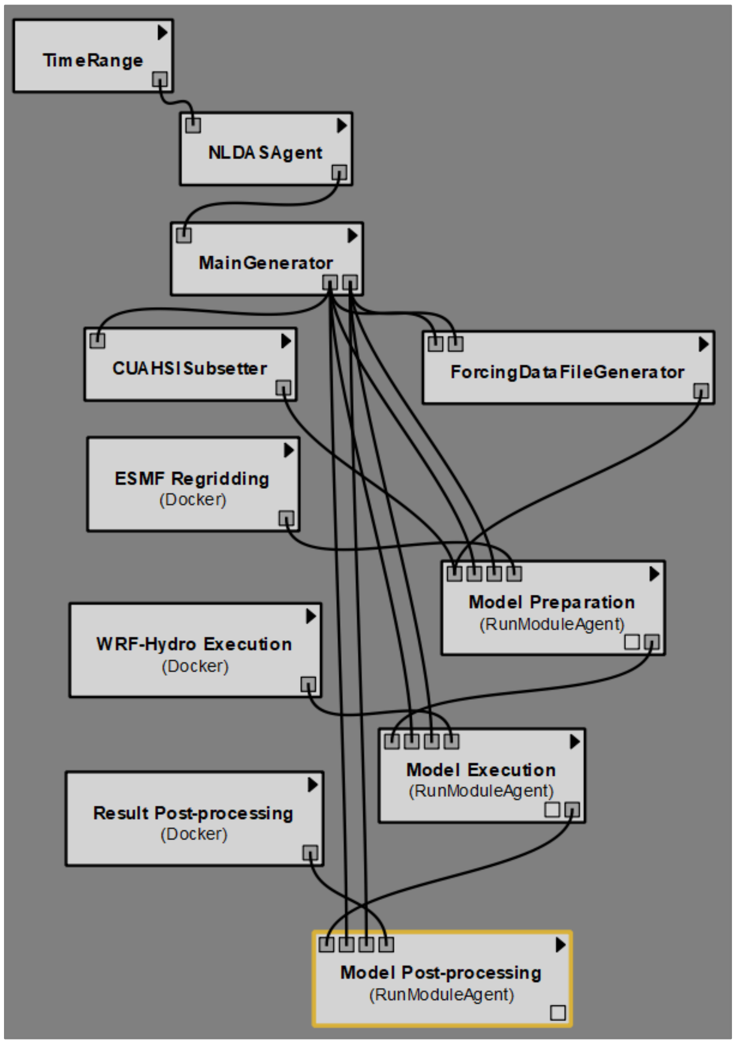


*Figure 93. The workflow connecting Result Post-processing Docker module and Model Post-processing RunModuleAgent*

19. Add a **UsgsAgent** component to the workspace. We will use this component to collect streamflow observations at the outlet of our watershed and compare them with our simulation results. Connect the *TimeRange* module to its input port then configure it with the following parameters:

- Inputs

| Variable | Value |
|---|---|
| desired_site_code | 05454300 |
| unit_conversion_factor | 0.028317 |

20. Add a **msmShowChart** component to the workspace. This component will generate a time series plot of our simulated streamflow along with USGS streamflow observations. Configure it with the following parameters:

- Inputs

| Variable | Value |
|---|---|
| units | m$^3$/s |
| variable_name | Discharge |

21. Connect the output of the **UsgsAgent** and **Model Post-processing** components to the input of the **msmShowChart** component:

| From Component | From Port | To Component | To Port |
|---|---|---|---|
| UsgsAgent | dataset_names | msmShowChart | Dataset_names_mainAxis |
| | | | |
| Model Postprocessing | Output01 | msmShowChart | Dataset_names_mainAxis |

The final model configuration should resemble the workflow in Figure 94 which can be downloaded from CyberWater website <11. WRF-Hydro Model Simulation with Docker>.

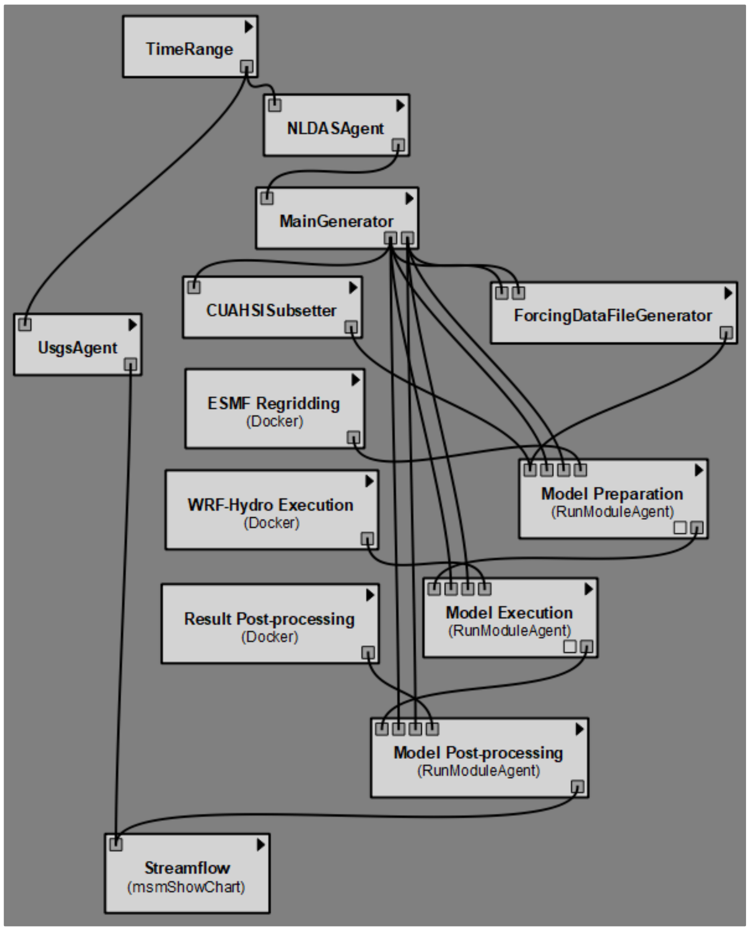

TimeRange
NLDASAgent
MainGenerator
CUAHSISubsetter
ForcingDataFileGenerator
UsgsAgent
ESMF Regridding (Docker)
WRF-Hydro Execution (Docker)
Model Preparation (RunModuleAgent)
Result Post-processing (Docker)
Model Execution (RunModuleAgent)
Model Post-processing (RunModuleAgent)
Streamflow (msmShowChart)

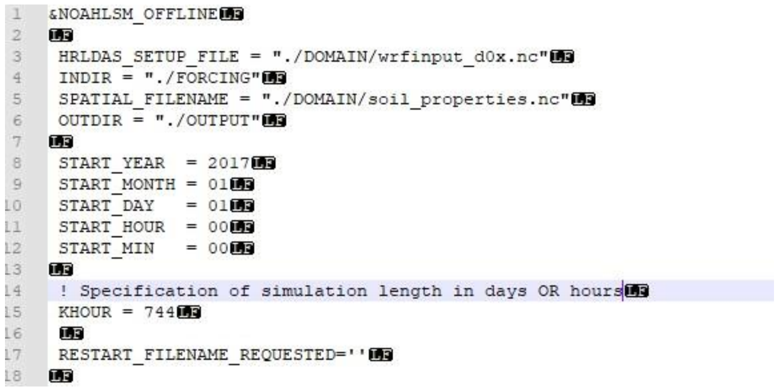
*Figure 94. The overall workflow to execute the WRF-Hydro model*

22. Before executing the simulation, we need to make sure that our WRF-Hydro namelists (i.e. configuration files) are defined correctly. Specifically, we need to make sure that the *simulation*

*start time* and *simulation duration* defined in the namelist.hrldas file match what was specified in the *TimeRange* component.

Update the *namelist.hrldas* file with the desired start time and simulation duration as shown in Figure 95:

| Variable | Value |
|---|---|
| START_YEAR | 2017 |
| START_MONTH | 01 |
| START_DAY | 01 |
| START_HOUR | 00 |
| START_MIN | 00 |
| | |
| KHOUR | 744 |



```
&NOAHLSM_OFFLINE

 HRLDAS_SETUP_FILE = "./DOMAIN/wrfinput_d0x.nc"
 INDIR = "./FORCING"
 SPATIAL_FILENAME = "./DOMAIN/soil_properties.nc"
 OUTDIR = "./OUTPUT"

 START_YEAR  = 2017
 START_MONTH = 01
 START_DAY   = 01
 START_HOUR  = 00
 START_MIN   = 00

 ! Specification of simulation length in days OR hours
 KHOUR = 744

 RESTART_FILENAME_REQUESTED=''

```

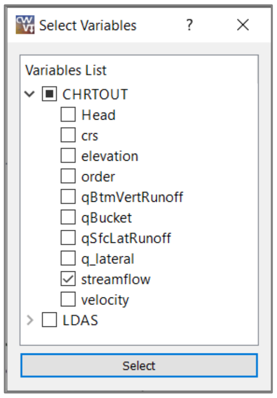
*Figure 95. The namelist.hrldas file*

- IMPORTANT NOTE

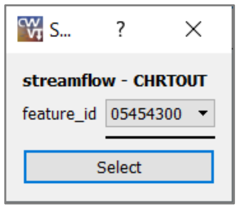
  We recommended that the Notepad++ text editor is used to edit these files to ensure that Windows line endings are not appended to each line of the document. Since we'll be submitting these documents to execute on a Linux computer, we need to make sure that our text documents do not contain Windows carriage returns. This can be done with the following steps (see this website for more information):

  a. Select **View** > **Show Symbol** > **Show End of Line**
  b. Use the menu item **Edit** > **EOL Conversion,** then select **Unix (LF)**.

23. Execute the simulation which would take around 30 minutes on personal computers. When the simulation is complete, a dialog will appear allowing us to choose output variables that we can plot. Since we've configured the **USGSAgent** to collect NWIS streamflow at gage location 05454300, we should select the corresponding reach in our simulation. Select "streamflow" from "CHRTOUT" as shown in Figure 96(a). Then a second drop list will pop out, select "05454300" from it as shown in Figure 96(b). The resulting diagram is shown in Figure 97.



(a)



(b)

*Figure 96(a)The prompt dialog to choose output variables to plot, (b)The prompt dialog to choose the gage location.*

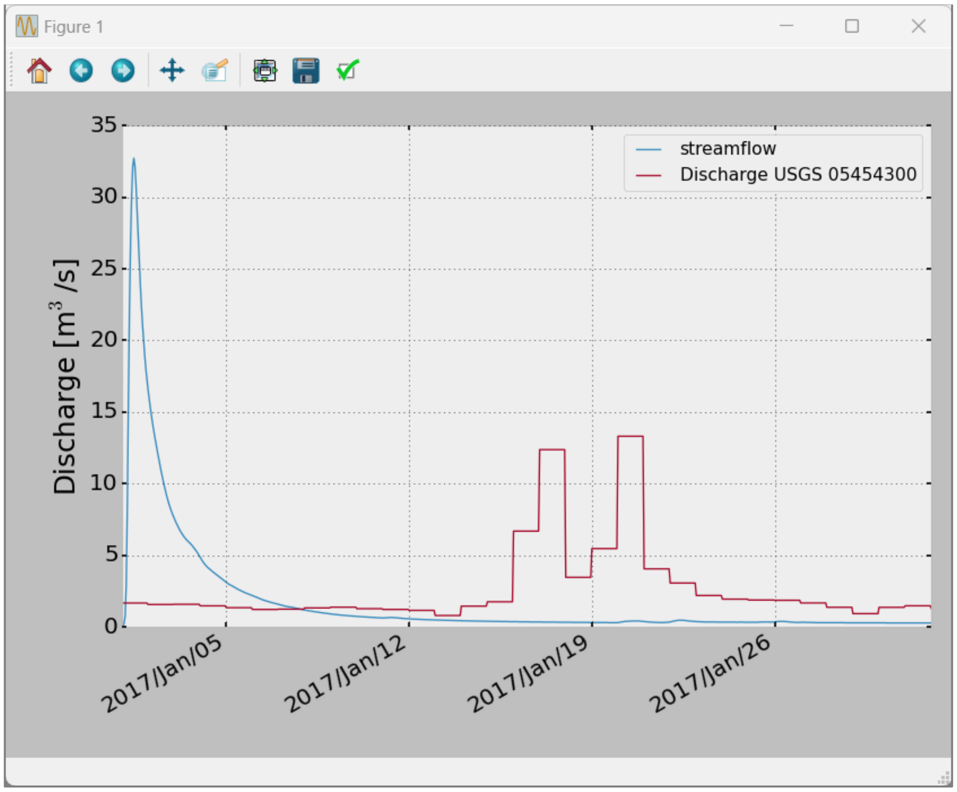


*Figure 97. The resulting diagram of WRF-Hydro model compared with USGS data at the gage location of 05454300.*

Notice that the WRF-Hydro simulated streamflow does not closely match the observed discharge at NWIS 05454300. This may be due to many factors such as, the short simulation duration, a cold start simulation (i.e. initial state values do not reflect reality), or it could be a product of our meteorological forcing data. While it's beyond the scope of this case study, we encourage users to explore this by extending the simulation duration, swapping forcing data, or using model restarts to provide accurate initial conditions.