**WBM Instruction Manual**

Danielle S. Grogan, Shan Zuidema, Alex Prusevich, Wil Wollheim, Stanley Glidden, Richard B. Lammers

Earth Systems Research Center, Institute for the Study of Earth, Oceans, and Space, University of New Hampshire, Durham, NH, 03824, US

This instruction manual accompanies the University of New Hampshire Water Balance Model (WBM) open source code v1.0.0 available via GitHub (https://github.com/wsag/WBM), and described in:

Grogan, D.S., Zuidema, S., Prusevich, A., Wollheim, W., Glidden, S., Lammers, R.B. (2022) WBM: A scalable gridded global hydrologic model with water tracking functionality. *Geoscientific Model Development*, x, xxxxxx.

Contacts:

Danielle.grogan@unh.edu

Shan.zuidema@unh.edu

Richard.lammers@unh.edu

Alex.prusevich@unh.edu

Stanley.Glidden@unh.edu

unh.wbm@unh.edu

**Table of Contents**

General model overview

Installation: Singularity container

Workflow structure

Step 1. Prepare input data, metadata, and parameter files

Step 2. Write a model setup file

Step 3. Test setup file

Step 4. Execute the model code

Step 5. Post-processing

Directory structure and required files

Tools for pre-processing data inputs

Other useful tools

References

1. Timeline

   Description automatically generated**General model overview**

**Figure 1. Water balance model schematic showing major fluxes and storages.**

The University of New Hampshire Water Balance Model (WBM) is a process-based, gridded hydrologic model that simulates spatially and temporally varying water volumes and quality. WBM represents all major land surface components of the hydrologic cycle, and tracks fluxes and balances between the atmosphere, above-ground water storages (e.g. snowpack), soil, vegetation, groundwater, and runoff. A digitized river network connects each grid cell to the next, enabling simulation of flow through river systems. WBM includes domestic and industrial water requirements and use, agricultural water requirements and use (irrigation and livestock), and hydro-infrastructure (dams, canals, and inter-basin transfers). While the model is considered global, it can be run for any region and any spatial resolution given available input data at the appropriate resolution.

The open source WBM code described here is composed of three main files: wbm.pl, which is the main model script; WBM.pm, a module providing WBM specific functionality; and RIMS.pm, a module providing geospatial and temporal transformation utilities. The entire modeling framework is dependent upon other software: perl, PDL, gdal, ogr, and NetCDF. The model input data repository (<https://wbm.unh.edu/>) also includes a Singularity container which has pre-installed the required operating system and software dependencies for ease of model use by the research community.

1. **Installation: Singularity container**

**2.1 Singularity General Notes**

These instructions assume you have a working installation of Singularity version >= 3.7.1 (Available from https://sylabs.io/docs) on your system. Other versions may work, but all testing was done with versions >= 3.7.1.

Using Singularity, WBM can be run on Windows, MacOS, or Linux systems. Singularity runs on Linux natively and uses virtual machines (VMs) on Windows and Mac.

These instructions, the Singularity image of WBM and model input data are available from the University of New Hampshire Water Balance Model Ancillary Data Download Website, [https://wbm.unh.edu](https://wbm.unh.edu/).

**2.2 Downloading the Ancillary files**

The default setup for WBM utilizes a read-only files system within the Singularity container. The initialization, input, output, spool and WBM executable files are placed on a read-write enabled volume external to the Singularity container. To facilitate the ease with which a WBM model run can be set up, several files and folders are provided contained in a file called wbm\_storeage\_v1.0.0.tar.gz that need to be placed on the host file system. A location with sufficient disk space and permissions for WBM to read model input, read/write spool files and to write output must be selected. The approximate minimum disk space requirement for the external files is 30 gigabytes.

For the purposes of these instructions, we assume there exists a folder called /userdata/mywbm on the host computer system and that this is where you are going to be installing WBM. Adjust this location appropriately to match your host computer's file system and the location you want to install the Singularity version of WBM.

**2.3 WBM Storage**

Change directory to /userdata/mywbm or the appropriate folder that you are going to use as the external data storage folder for WBM.

Download the wbm\_storage\_v1.0.0.tar.gz file from the web site using a web browser. Alternately, many OS distributions have the wget command installed or available (Linux, MacOSX, Windows PowerShell) which can be used instead. To use wget you would make sure you are in the /userdata/mywbm folder and type:

wget "https://wbm.unh.edu/v1.0.0/wbm\_storage\_v1.0.0.tar.gz''

Once you have the compressed tar file, you need to uncompress and untar this file. Type the following command:

>tar -zxvf wbm\_storage\_v1.0.0.tar.gz

There are other software tools to expand the file, such as 7zip, however the tar utility may be available by default.

This will expand the tar file into nine folders. A listing of the directory would look like this:

>ls /userdata/mywbm/wbm\_storage\_v1.0.0

***data/ data\_init/ gdal\_test\_files/ model/ spool/ utilities/ wbm\_init/ wbm\_output/ WBM\_run\_state/***

Later we will use the Singularity -B flag to mount these external folders and make the files accessible from within the container.

**2.4 WBM Ancillary Data**

In the previous section (WBM storage) the typical run environment directory structure for WBM was setup. Continuing with our example within that file structure, you should now have a folder called 'data' /userdata/mywbm/wbm\_storage\_v1.0.0/data). From the WBM Ancillary Data Download website, download the file WBMAncillaryData\_v1.0.0.tar.gz in the same manor outlined in the previous section (using a web browser or wget) and uncompress the contents into that directory. Do this using the file system tools of your choice (tar, 7zip, etc) which was also outlined in the previous section. The data directory will now contain the custom data and initialization files associated with the WBM Open Source Release Version 1.0.0.(DOI:https://dx.doi.org/10.34051/d/2022.2). The data provided is only a subset of what is needed in order to execute the two WBM model initialization scripts provided in the wbm\_init folder (/userdata/mywbm/wbm\_storage\_v1.0.0/wbm\_init). These scripts are called Global\_tracking.init and Wyoming\_tracking.init and will be discussed in the last section. Additional publicly available data files are required for successful execution of these two model runs and the sources for obtaining the additional data sets will be listed in the associated paper once published. There will be a link to the paper off of the Ancillary Data web site.

**2.5 Downloading the Singularity Image file**

Here are the steps to download the Singularity image wbm\_opensource\_v1.0.0.sif from the WBM Ancillary Data Download Website. This is the Singularity .sif file that supports a read-only file system, so the .sif file will be immutable. Similar to earlier when you downloaded the external data files, change directory to /userdata/mywbm or an appropriate folder that you have write access to.

Download the wbm\_opensource\_v1.0.0.sif file from the web site (wbm.unh.edu) using a web browser. Alternately, many OS distributions have the wget command installed or available (Linux, MacOSX, Windows PowerShell) which can be used instead. To use wget you would make sure you are in the /userdata/mywbm folder and type:

wget "https://wbm.unh.edu/v1.0.0/wbm\_opensource\_v1.0.0.sif''

* 1. **Starting an Instance from the Singularity Image file**

**2.6a) Typical Read-only Method**

If you do not plan to make any changes to the wbm code, you can use this default read-only method to create a Singularity instance. To start an instance issue the command:

Note: You probably need to adjust the path to the folders external to the Singularity image, which is the path to the left of the : in the -B (bind) arguments. The path to the right of the colon is the path within the singularity container and should not be modified.

singularity instance start \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/data:/wbm/data \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/data\_init:/wbm/data\_init \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/spool:/wbm/spool \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/wbm\_output:/wbm/wbm\_output \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/WBM\_run\_state:/wbm/WBM\_run\_state \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/wbm\_init:/wbm/wbm\_init \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/model:/wbm/model \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/utilities:/wbm/utilities \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/gdal\_test\_files:/wbm/gdal\_test\_files \

wbm\_opensource\_v1.0.0.sif wbm\_os\_instance1

This will start a read-only singularity instance.

To open a shell to this instance from which you can run wbm, type:

singularity shell instance://wbm\_os\_instance1

The prompt should now look similar to this:

Singularity:~>

**2.6b) Persistent Read/Write Method using a built in Overlay**

If you plan to make any changes to the wbm code that will persist, you will need download the .sif file from the wbm.unh.edu website that has an overlay added. The file is called wbm\_opensource\_rw\_v1.0.0.sif. In addition to using the.sif with the overlay, you need to use the --writable flag to utilize the provided overlay that was built into the sif file, into which your modifications will be stored.

Issue the following command:

*Note: You probably need to adjust the path to the folders external to the Singularity image, which is the path to the left of the : in the -B (bind) arguments. The path to the right of the colon is the path within the singularity container and should not be modified.*

singularity instance start \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/data:/wbm/data \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/data\_init:/wbm/data\_init \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/spool:/wbm/spool \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/wbm\_output:/wbm/wbm\_output \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/WBM\_run\_state:/wbm/WBM\_run\_state \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/wbm\_init:/wbm/wbm\_init \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/model:/wbm/model \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/utilities:/wbm/utilities \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/gdal\_test\_files:/wbm/gdal\_test\_files \

--writable wbm\_opensource\_v1.0.0.sif wbm\_os\_instance1

This will start a read/write singularity instance.

To open a shell to this instance from which you can run wbm, type:

singularity shell instance://wbm\_os\_instance1

The prompt should now look similar to this:

Singularity>

Note: If you would rather open a singularity shell rather then start a singularity instance, simply replace ‘instance start’ with ‘shell’ in the above examples. Using instances allows Singularity runs to be exited and remain running.

**2.7 Executing WBM from the Singularity Image, without starting a shell**

Note: You probably need to adjust the path to the folders external to the Singularity image, which is the path to the left of the : in the -B (bind) arguments. The path to the right of the colon is the path within the singularity container and should not be modified.

#To execute wbm from the command line, rather than open a shell

singularity exec \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/data:/wbm/data \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/data\_init:/wbm/data\_init \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/spool:/wbm/spool \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/wbm\_output:/wbm/wbm\_output \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/WBM\_run\_state:/wbm/WBM\_run\_state \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/wbm\_init:/wbm/wbm\_init \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/model:/wbm/model \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/utilities:/wbm/utilities \

-B /userdata/mywbm/wbm\_storage\_v1.0.0/gdal\_test\_files:/wbm/gdal\_test\_files \

wbm\_opensource\_v1.0.0.sif /wbm/model/wbm.pl

The above command should execute the wbm.pl script and display the usage information.

**2.8 Checking the setup of the external volumes needed by WBM**

Now that the container is running, you can check if the external folders are mounted properly. To check this type the following from within the running Singularity container:

ls /wbm/data

you should see the data folders and one file:

*WBMAncillaryData\_Readme.txt* ***crop/ data\_init/*** ***glaciers/ irrigation/ soil/***

**2.9 Executing WBM**

**2.9.1 Testing the setup**

There is a test initialization file (test.init) packaged with WBM. This is a good place to start in order to test if your system is running WBM properly. Executing WBM on the test initialization file will produce a local copy of the output (both printed to screen and saved in a .log file) that can be compared to the following standard result:

 Summary for Year 2000:

   Rff-Gl/Precip  Yr = (8.24 / 15.84), 52.03 %

   Runoff-Dischg  Yr = (8.24 - 8.21), 0.03 km3

   ET Non/Irr/Rfd Yr = (7.60 / 0.00 / 0.00) km3

   Surface W Storage =  0.03 km3

   Grndwater Storage =  0.65 km3

Any discrepancy between the two results will indicate an installation problem with WBM.

Change directory to the Singularity root directory / if you opened a shell to the singularity container. To execute the test model, type the following:

/wbm/model/wbm.pl -v /wbm/wbm\_init/wbm\_test.init

On a typical system this model will run for less than one minute.

If the model completes with no errors, you can now compare the output in the test.log file with the summary output provided above. The path to the newly-generated output folder from within the container is /wbm/wbm\_output/wbm\_test/. From outside the container the path will be /userdata/mywbm/wbm\_storage\_v1.0.0/wbm\_output.

**2.9.2 Executing the model runs Global\_tracking and Wyoming\_tracking**

If no errors were encountered while executing the test model run (test.init ), you should now be able to execute the model runs Global\_tracking and Wyoming\_tracking that were provided in the download. As stated earlier, there are a number of publicly available datasets required to be downloaded for the successful execution of these two model runs. The data initialization files for the two model runs has been provided in the download of wbm\_storage\_v1.0.0 and should now be located in the data\_init folder (/userdata/mywbm/wbm\_storage\_v1.0.0/data\_init) under the appropriate subfolder. The subfolders are named for the type of data they are to contain. These initialization scripts contain the citation, metadatalink and downloadlink for all of the required data.

Once the additional data files are downloaded and in place, execution of the scripts would be similar to the test.init script run ealier.

To execute the Global\_tracking.init model type:

/wbm/model/wbm.pl -v /wbm/wbm\_init/Global\_tracking.init

To execute the Wyoming\_tracking model type

/wbm/model/wbm.pl -v /wbm/wbm\_init/Wyoming\_tracking.init

1. **WBM Workflow**

The WBM workflow has 5 basic steps. These are:

Step 1. Prepare input data, metadata, and parameter files

Step 2. Write a model setup file

Step 3. Test setup file

Step 4. Execute the model code

Step 5. Post-processing

**Step 1. Prepare input data, metadata, and parameter files**

*Input Data*

For almost all input files, WBM can utilize any type of GDAL-readable file format. These include: rasterized geotiffs, ascii grids, netCDF (2 or 3 dimensions), and shapefiles.

In addition to GDAL-readable files, WBM can also take input databases as .csv files for Dams & Reservoirs and for Inter-basin Transfers. See the HydroConDams v2.0 (Zuidema & Morrison, 2020) database for an example of the required Dams & Reservoirs database format. See Zaveri et al. (2016) Supplemental Info for an example of the required Inter-basin Transfers database format.

See the file on here: https://github.com/wsag/WBM/tree/main/instructions/WBM\_init\_file\_instructions.csv

for a description of all optional model inputs, and the file formats that WBM can accept for those inputs.

To reproduce model results from Grogan et al. (2022), please refer to the input data list in Table 1. All of these data are available for download online, either directly from the original source or from <https://wbm.unh.edu/>. If you cannot access any required data, please contact the authors.

Table 1: Input data required to reproduce model results in Grogan et al. (2022)

|  |  |  |  |
| --- | --- | --- | --- |
| **Input data type** | **Input data** | **Download link or website** | **Citation** |
| River network | MERIT 5-minute river network | http://hydro.iis.u-tokyo.ac.jp/~yamadai/MERIT\_Hydro/ | Yamazaki et al. (2019) |
| Precipitation (daily) | MERRA 2 (prectotcorr variable) | https://gmao.gsfc.nasa.gov/reanalysis/MERRA-2/ | Gelaro et al. (2017) |
| Temperature (daily average) | MERRA 2 | https://gmao.gsfc.nasa.gov/reanalysis/MERRA-2/ | Gelaro et al. (2017) |
| Dams & reservoirs | HydroConDams v2.0 and GrAND v1.3 | https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/5YBWWI  https://globaldamwatch.org/grand/ | Lehner et al. (2011); Zuidema & Morrison (2020) |
| Soil available water capacity | Harmonized world soil database v1.2 | https://www.fao.org/soils-portal/data-hub/soil-maps-and-databases/harmonized-world-soil-database-v12/en/ | Fischer et al. (2008) |
| Root depth\* | Effective rooting depth from Yang et al. (2016), gap-filled with the  FAO/UNESCO digital soil map of the world v3.6 | https://doi.org/10.4225/08/5837b3aa9cb90  <https://www.worldcat.org/title/digital-soil-map-of-the-world-and-derived-soil-properties/oclc/52200846>  and:  <https://wbm.unh.edu/> | FAO/UNESCO (2003); Yang et al. (2016) |
| Glacier runoff, volume and area\* | GloGEM glacier model | <https://wbm.unh.edu/> | Huss & Hock (2015) |
| Crop maps & calendars\* | MIRCA2000 v1.1 | <https://www.uni-frankfurt.de/45218023/MIRCA>  and:  <https://wbm.unh.edu/> | Portmann et al. (2010) |
| SW:GW ratio\* | FAO AQUASTAT | <https://www.fao.org/aquastat/statistics/query/index.html;jsessionid=71F6F6340C470CFBE92D71489546AA39>  and:  <https://wbm.unh.edu/> | FAO (2015) |
| Irrigation Efficiency | Rasterized data from Table 1 of Döll and Siebert (2002) | https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2001WR000355 | Döll & Siebert (2002) |
| Rice paddy percolation rate\* | Derived from the FAO/UNESCO soil map of the world | <https://www.fao.org/soils-portal/soil-survey/soil-maps-and-databases/faounesco-soil-map-of-the-world/en/>  and:  <https://wbm.unh.edu/> | FAO/UNESCO (2003), with derived data described by Wisser et al. (2008) |

\*Primary data was processed for formatting, gap-filling, or to generate a calculated product; the resulting formatted files are provided for download at <https://wbm.unh.edu/> (https://dx.doi.org/10.34051/d/2022.2) for simulation reproducibility.

*Metadata*

Each spatial data set and database must be described in a metadata file with the extension *“.init”*. These files are .txt files in the format of a perl-readable hash. See the file <https://github.com/wsag/WBM/tree/main/instructions/data_init_instructions.init> for how to write this file with correct formatting, and the optional and required hash keys.

Input file unit conversions (e.g., converting temperature data from ºC to ºF) do not need to be performed prior to running WBM. Rather, the user can define a conversion slope and intercept for linear transformations within the metadata “.init” files, and WBM will automatically calculate the new units through the RIMS.pm module.

Example of a metadata.init file with unit conversions:

{

Code\_Name => 'merra2\_prectotcorr\_d', # This is the dataset ID

Data\_Cube => 'merra2\_prectotcorr',

Project => 'MERRA2',

Time\_Series => 'daily',

Start\_Date => '1980-01-01',

End\_Date => '2018-12-31',

Name => 'MERRA2 Total Surface Precipitation, bias corrected, Daily',

Param\_Name => 'Precipitation',

Var\_Name => 'PRECTOTCORR', # Variable name in the source data files, e.g. NetCDF

Orig\_Units => 'kg/(m^2 s)', # Units in the original data files

Units => 'mm/day', # Units in the output binary files to be used by WBM

Var\_Scale => 86400, # Unit conversion: Scaler (slope)

Var\_Offset => 0, # Unit conversion: Offset (intercept)

Bands => 1,

Projection => 'epsg:4326',

File\_Path => '/wbm/data/climate/precip/\_YEAR\_/MERRA2.PRECTOTCORR.\_YEAR\_-\_MONTH\_-\_DAY\_.nc;',

DownloadLink => 'https://gmao.gsfc.nasa.gov/reanalysis/MERRA-2/',

citation => 'The Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2), Ronald Gelaro, et al., 2017, J. Clim., doi: 10.1175/JCLI-D-16-0758.1',

DOI => ‘10.1175/JCLI-D-16-0758.1'

}

All metadata.init files associated with the data listed in Table 1, which are required to reproduce results from Grogan et al. (2022), are provided in the data\_init directory of the data download here: <https://wbm.unh.edu/>

*Parameter Files*

Parameter files are .csv files that either directly list parameters, or point to a set of files containing spatially- or temporally-varying parameters.

Crop parameters:

WBM uses the set of crop parameters described in Siebert and Döll (2010), which are meant for use with the MIRCA2000 (Portmann et al., 2010) global crop database. WBM requires the crop parameters to be formatted as gridded time series; due to the extensive formatting required to use the MIRCA2000 crop data in WBM, we provide the gridded time series of aggregate crop parameters for the four crop categories used in Grogan et al. (2022) here: <https://wbm.unh.edu/>. In the provided data download, see the file data\_init/crops/MIRCA2000\_average\_landCoverParameters.csv for the required format of the crop parameter file.

Livestock parameters:

WBM uses the parameters from Steinfeld et al. (2006) to calculate livestock water use by animal type, along with a gridded field of livestock density. See Grogan et al. (2022) Table 2 for default parameter values.

The parameter file is a .csv file with entries in columns with the following required column headers:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stock | SlopeValue | InterceptValue | ServiceWater | AnimalGrowthRate | DensityFile |
| *Character string: Type of animal* | *Numeric* | *Numeric* | *Numeric* | *Numeric* | *Character string: path to file* |

**Step 2. Write a model setup file with extension “.init”**

In Step 2, the model user writes a model setup file with the extension *“.init”* that lists all model inputs as well as other key parameters such as the start year, end year, list of output variables to save, and output directory location. This setup file points directly to the input metadata .init files, and includes options to directly define parameter values and set binary on/off flags for particular modules. Most important is the identification of the digital river network. The input river network file determines the model simulation grid spatial resolution, spatial extent, projection, and defines non-land grid cells (which are set to a no data value). Other input datasets will automatically be clipped (extent reduced) and re-gridded (either through resampling or aggregation) to match the extent and grid cell resolution of the input river network file. This means that the model user does not need to do these spatial transformations prior to starting the model.

See the file on here: <https://github.com/wsag/WBM/tree/main/instructions/WBM_init_file_instructions.csv>

for instructions on how to write a model setup file, including formatting and all input, parameter, and on/off flag options.

The model setup files used for the simulations in Grogan et al. (2022) are provided here: <https://wbm.unh.edu/>. In the provided data download, see the files wbm\_init/Global\_tracking.init and wbm\_init/Wyoming\_tracking.init.

**Step 3. Test setup file**

In Step 3, the user tests the model setup and produces an optional input data pre-processing script. Test mode and “noRun” mode call the input data reading functions from RIMS.pm and set up the model run’s output directory. This step is used to identify any errors in the model setup, which are commonly issues such as incorrect file paths, syntax errors in the “*.init*” files, or formatting errors in the raw data files. Executing wbm.pl in test and noRun mode also automatically generates a custom build\_spool.pl script (written to the model run’s output directory) that can optionally be executed prior to Step 4 to pre-process all input data files that require spatial clipping, re-gridding, or unit conversions.

If build\_spool.pl is executed (check its “-h” help options first where a user can set number of CPUs to use and output to netCDF files), the results of input data pre-processing are saved in the spool/ directory as binary files that are read directly by WBM; these files can also optionally be saved as netCDF files for ease of analysis, so the user can evaluate the results of the processing step. If the custom build\_spool.pl script is not executed prior to starting the model in Step 4, it will automatically be executed in the model’s run time. Note, this automatic option only produces binary files, and does not output any netCDF files.

**Step 4. Execute the model code**

In Step 4, the model user executes wbm.pl via direct command line entry.

**IMPORTANT:** When wbm.pl is called, it first checks to see if the binary files associated with a given input file already exist in the spool/ directory. If they do, then wbm.pl does not re-write these files, but rather uses the existing binary files. If the user changes values in the input file, but the file keeps the same name, wbm.pl will not recognize that the binary file needs to be updated. ***We highly recommend that users ensure unique names go with each version of data input files. If a file is updated and you are not sure if the binary file still reflects the correct input data, then delete the binary file and re-make it.*** You can also use the -rmSpool flag (see below) to automatically delete ALL binary files associated with a given river network when you execute wbm.pl.

The code wbm.pl has several flag options, including -h for help, -v for verbose mode, and others described in the instruction manual. The model setup file is the only required argument to wbm.pl.

Usage:

wbm.pl [-h] [-v] [-bd|by|bm] [-test] [-rm] [-rmSpool] [-noOutput] [-noState] [-dState] [-m] [-t THREADS] [-tz THREAD\_SIZE] [-sl] [-idump] [-saveDams] [-err] [-spoolDir SPOOL\_DIR] [-stateDir STATE\_DIR] MODELSETUP.init

MODELSETUP.init is the model setup file created in Step 2.

|  |  |
| --- | --- |
| **Options:** |  |
|  |  |
| h | Display this help. |
| v | Verbose mode. |
| bd | Perform code benchmarking for daily intervals. |
| by | Perform code benchmarking for yearly intervals. |
| bm | Perform code benchmark mapping. Pause using "bm.lock" file. |
| test | Test mode: no output, WBM runs one time step only. |
| rm | Remove and do not update existing ouput files. |
| noRun | No run mode: Initialization only, creates auxiliary files in the output directory |
| rmSpool | Remove all of this run’s input spool files. |
| noOutput | Do not write output files. |
| noState | Do not read or write run state files. |
| dState | Write daily run state files. Forces "spinup\_state\_ID". |
| m | Memory saving mode. Default is to automatically use this option. |
| t | Number of threads to use. Default is 4. |
| tz | Minimum thread size. Default is 1 Mb. |
| sl | Use PDL slicing (faster when using many crops). |
| saveDams | Save a .csv file listing the subset of dams within this run’s spatial domain to the output directory. |
| err | Print all GDAL relevant STDERR to screen for debugging. |
| spoolDir | Overwrite path to spool directory defined in the model setup init file. |
| stateDir | Overwrite path to run state directory defined in the init file. |

**Step 5. Post-processing**

Step 5 is the most application- and user-specific. The raw daily model output is rarely the final product of analysis; temporal and spatial aggregation or point-location time series extraction are most commonly required to evaluate output and produce research results. The model setup file has a binary on/off option that enables automatic temporal aggregation from daily to monthly and yearly time steps, as well as an input field for automatic spatial aggregation. Perl utilities for these operations are included in the model GitHub repository in the utilities/ folder.

1. **Directory structure and required files**

**Required files (all are included in the Singularity container and/or data download here:** <https://wbm.unh.edu/>**):**

1. WBM.conf

WBM.conf is a configuration file that lists file paths (absolute or relative) to other required libraries and files. If you move any of these required files, you must update WBM.conf

1. WBM\_varAttr

The path to this file is identified in WBM.conf

This file lists all possible WBM output variables, along with their units and long-form names

The default path to this file on the WSAG servers is:

/net/nfs/zero/data3/WBM\_TrANS/speradsheets/WBM\_dataCube\_expand.csv

1. Gdal test files

The path to the directory with this collection of files is identified in WBM.conf

Files included:

* air.2m.1948.nc
* aqThis\_05min\_clip.tif
* MERRA.processed.rh2m.2000.nc
* MERRA.prod.assim.tavg1\_2d\_slv.Nx.19790101.SUB.nc
* Wrfout\_d03\_T2\_2006-02-01.nc
* test\_gdal.pl

These files are used to test that the gdal software is working properly for all WBM input data types.

**Directory Structure**

Within the Singularity file, the directory structure is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Level 1** | **Level 2** | **Level 3** | **Description** |
| data/ |  |  | All input data |
|  | climate/ |  |  |
|  |  | airTemp/ |  |
|  |  | precip/ |  |
|  | crops/ |  |  |
|  | dams/ |  |  |
|  | glaciers/ |  |  |
|  | irrigation/ |  |  |
|  | masks/ |  |  |
|  | network/ |  |  |
|  | RootDepth/ |  |  |
|  | soil/ |  |  |
| data\_init/ |  |  | All input metadata |
|  | climate/ |  |  |
|  | crops/ |  |  |
|  | dams/ |  |  |
|  | glaciers/ |  |  |
|  | irrigation/ |  |  |
|  | soil/ |  |  |
| gdal\_test\_files/ |  |  | For testing gdal |
| model/ |  |  | Contains model code |
|  | RIMS/ |  |  |
|  | \_Inline/ |  |  |
|  |  | build/ |  |
|  |  | lib/ |  |
| spool/ |  |  | Binary files are written here for direct use by WBM |
|  | 05min\_flwdir/ |  |  |
|  | Wyoming\_05min\_flwdir/ |  |  |
| utilities/ |  |  | Contains useful perl scripts for pre- and post-processing |
| wbm\_init/ |  |  | Model setup files |
| wbm\_output/ |  |  | Model output files |
|  | Global\_tracking/ |  |  |
|  |  | daily/ | Primary model output |
|  |  | monthly/ | Aggregated in run-time |
|  |  | yearly/ | Aggregated in run-time |
|  |  | climatology/ | Aggregated in post-processing |
|  |  | cell\_area/ |  |
|  |  | init\_files | Copy of input metdata and setup files |
|  | Wyoming\_tracking/ |  |  |
|  |  | daily/ | Primary model output |
|  |  | monthly/ | Aggregated in run-time |
|  |  | yearly/ | Aggregated in run-time |
|  |  | climatology/ | Aggregated in post-processing |
|  |  | cell\_area/ |  |
|  |  | init\_files |  |
| WBM\_run\_state/ |  |  | Run state: saved files of the model state at the end of each year |

1. **Tools for pre-processing data inputs**

**Network Tools:**

The most important data pre-processing step is preparation of the digital river network. WBM requires a digital river network input in a gridded gdal-readable format (e.g. tif, GridASCII, 2D netCDF, etc.), following the standard flow direction convention used by STN-30p (Vörösmarty et al., 2011), MERIT (Yamazaki et al., 2019), and HydroSHEDS (Lehner et al., 2008). If downscaling of climate data is required to match the elevation of a river network (which is recommended when the river network is of higher resolution than the climate input), then WBM also requires a gridded gdal-readable file that contains the average grid cell elevation of each river network grid.

The river network is used by WBM to define both the spatial resolution and the spatial extent of the simulation. For sub-global simulations, the user will need to subset a global digital river network to the desired spatial extent. All other spatial input files are rasterized, re-sampled, and clipped to match the spatial resolution and extent of the river network.

For the purposes of post-processing, the user may want the river network grid cells identified by river basin, and they may find data on the area upstream of any given grid cell to be useful.

We provide a utility that can perform all of the above tasks here:

<https://github.com/wsag/WBM/tree/main/utilities/networkTools.pl>

With instructions on how to use this tool for network subsetting here:

<https://github.com/wsag/WBM/tree/main/utilities/networkTools_manual.init>

**Spool file preparation tools:**

Before a WBM simulation can be started, all input data must be written as binary files and saved in a spool/ directory. While this step will be done automatically when wbm.pl is called, there are two reasons a user may want to build these binary files before calling wbm.pl:

(1) the user wants to evaluate the results of automatic spatial clipping, rasterizing, resampling, and unit conversion of input data prior to running the model;

(2) input files are large, and building spool files takes a long time.

In the case of very large input files (for example, global 5-minute gridded time series), users may be able to make use of multiple cores/CPUs, multiple servers, or other computational efficiencies that are not called automatically by wbm.pl, and thereby save time.

The model run specific build\_spool\_batch.pl script is automatically generated and saved to the output directory when a user runs wbm.pl first time with “-noRun“ option. It is a wrapper around build\_spool.pl (described below) to generate all required spool files for a given WBM simulation. We recommend using it as a preferred method instead of build\_spool.pl.

We provide a utility to build the binary spool files:

<https://github.com/wsag/WBM/tree/main/model/build_spool.pl>

Note, this utility must be in the model/ directory because it is called directly by wbm.pl. build\_spool.pl builds the binary files for time series data. Static data (not time series) do not require binary formats.

Usage:

build\_spool.pl [-h] [-v] [-rm] [-sd YYYY-MM-DD] [-ed YYYY-MM-DD] [-nc] [-f FORKS] [-r RESAMPLE\_INT] [-p PATCH\_VALUE] [-pp PPATCH\_VALUE] [-spDir SPOOL\_DIR] [-proj NET\_PROJ] NETWORK\_PATH METADATA.init

Options:

h Display this help.

v Verbose mode.

rm Remove existing spool files.

f Number of forks to use. Default is 8.

sd Start date for the time series.

ed End date for the time series.

proj River Network projection. Default is "epsg:4326".

r Resample method as Integer number. See GDAL docs. Default is 1 (bilinear).

('near'=>0,'bilinear'=>1,'cubic'=>2,'cubicspline'=>3,'lanczos'=>4,'average'=>5,

'mode'=>6,'max'=>7,'min'=>8,'med'=>9,'Q1'=>10,'Q3'=>11)

p Apply PATCH\_VALUE for bad values over the network grid.

It can be a number or secondary dataset ID. Default is undef.

pp Primary PPATCH\_VALUE for bad values over the network grid.

spDir Spool directory for the output files.

mt File path to an alternative Magic Table file.

nc Build additional NetCDF copy of spool binary data.

NETWORK\_PATH is the river network file. METADATA.init is the metadata.init file for the data you want to convert to binary format.

1. **Other useful tools**

WBM co-author Grogan maintains a GitHub repository of R code used in post-processing WBM model output here: <https://github.com/daniellegrogan/WBMr>.

Singularity container installation instructions and user guide can be found here: <https://sylabs.io/guides/3.5/user-guide/>

**References**

Döll, Petra, & Siebert, S. (2002). Global modeling of irrigation water requirements: GLOBAL MODELING OF IRRIGATION WATER REQUIREMENTS. *Water Resources Research*, *38*(4), 8-1-8–10. https://doi.org/10.1029/2001WR000355

FAO/UNESCO (2003). Digital soil map of the world and derived soil properties. [Electronic resource]. Rome, Italy: FAO.

FAO (2015). AQUASTAT Database. AQUASTAT [Website](https://www.fao.org/nr/water/aquastat/data/query/index.html?lang=en)  accessed on 16/07/2015.

Fischer, G., Nachtergaele, F., Prieler, S., van Velthuizen, H. T., Verelst, L., & Wiberg, D. (2008). Global Agro-ecological Zones Assessment for Agriculture (GAEZ 2008). *IIASA, Laxenburg, Austria and FAO, Rome, Italy.*

Gelaro, R., McCarty, W., Suárez, M. J., Todling, R., Molod, A., Takacs, L., et al. (2017). The Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2). *Journal of Climate*, *30*(14), 5419–5454. <https://doi.org/10.1175/JCLI-D-16-0758.1>

Huss, M., & Hock, R. (2015). A new model for global glacier change and sea-level rise. *Frontiers in Earth Science*, *3*. https://doi.org/10.3389/feart.2015.00054

Lehner, B., Verdin, K., & Jarvis, A. (2008). New Global Hydrography Derived From Spaceborne Elevation Data. *Eos, Transactions American Geophysical Union*, *89*(10), 93–94. https://doi.org/10.1029/2008EO100001

Lehner, B., Liermann, C. R., Revenga, C., Vörösmarty, C., Fekete, B., Crouzet, P., et al. (2011). High‐resolution mapping of the world’s reservoirs and dams for sustainable river‐flow management. *Frontiers in Ecology and the Environment*, *9*(9), 494–502. <https://doi.org/10.1890/100125>

Portmann, F. T., Siebert, S., & Döll, P. (2010). MIRCA2000-Global monthly irrigated and rainfed crop areas around the year 2000: A new high-resolution data set for agricultural and hydrological modeling: MONTHLY IRRIGATED AND RAINFED CROP AREAS. *Global Biogeochemical Cycles*, *24*(1), n/a-n/a. https://doi.org/10.1029/2008GB003435

Siebert, S., & Döll, P. (2010). Quantifying blue and green virtual water contents in global crop production as well as potential production losses without irrigation. *Journal of Hydrology*, *384*(3–4), 198–217. <https://doi.org/10.1016/j.jhydrol.2009.07.031>

Steinfeld, H., Gerber, P., Wassenaar, T., Castel, V., Rosales, M., & de Haan, C. (2006). Livestock’s long shadow. Retrieved from <http://www.fao.org/docrep/010/a0701e/a0701e00.HTM>

Vorosmarty, C.J., B.M. Fekete, F.G. Hall, G.J. Collatz, B.W. Meeson, S.O. Los, E.Brown De Colstoun, and D.R. Landis. 2011. ISLSCP II River Routing Data (STN-30p). ORNL DAAC, Oak Ridge, Tennessee, USA. https://doi.org/10.3334/ORNLDAAC/1005

Wisser, D., Frolking, S., Douglas, E. M., Fekete, B. M., Vörösmarty, C. J., & Schumann, A. H. (2008). Global irrigation water demand: Variability and uncertainties arising from agricultural and climate data sets. *Geophysical Research Letters*, *35*(24), L24408. https://doi.org/10.1029/2008GL035296

Yamazaki, D., Ikeshima, D., Sosa, J., Bates, P. D., Allen, G. H., & Pavelsky, T. M. (2019). MERIT Hydro: A High‐Resolution Global Hydrography Map Based on Latest Topography Dataset. *Water Resources Research*, *55*(6), 5053–5073. https://doi.org/10.1029/2019WR024873

Yang, Y., Donohue, R. J., & McVicar, T. R. (2016). Global estimation of effective plant rooting depth: Implications for hydrological modeling: GLOBAL HYDROLOGICAL EFFECTIVE ROOTING DEPTH. *Water Resources Research*, *52*(10), 8260–8276. https://doi.org/10.1002/2016WR019392

Zaveri, E., Grogan, D. S., Fisher-Vanden, K., Frolking, S., Lammers, R. B., Wrenn, D. H., et al. (2016). Invisible water, visible impact: groundwater use and Indian agriculture under climate change. *Environmental Research Letters*, *11*(8), 084005. https://doi.org/10.1088/1748-9326/11/8/084005

Zuidema, S., & Morrison, R. (2020). Hydrologically Consistent Dams Database (version 2.0) [Data set]. Harvard Dataverse. https://doi.org/10.7910/DVN/5YBWWI