# WTools

## WTools and WFlow

WTools is a set of open-source python-based tools to derive WFlow models from a set of topographic layers (see Figure 1). The tool aims to:

* Simply model setup. Less knowledge about GIS-processing is required since maps are processed automatically once topographic layers are derived. Model setup is an easier and faster process.
* Uniform model setup. All WFlow models will have common characteristics, as they all have the same origin.
* Multiple catchments in one model. Models can cover more than one catchment if required.

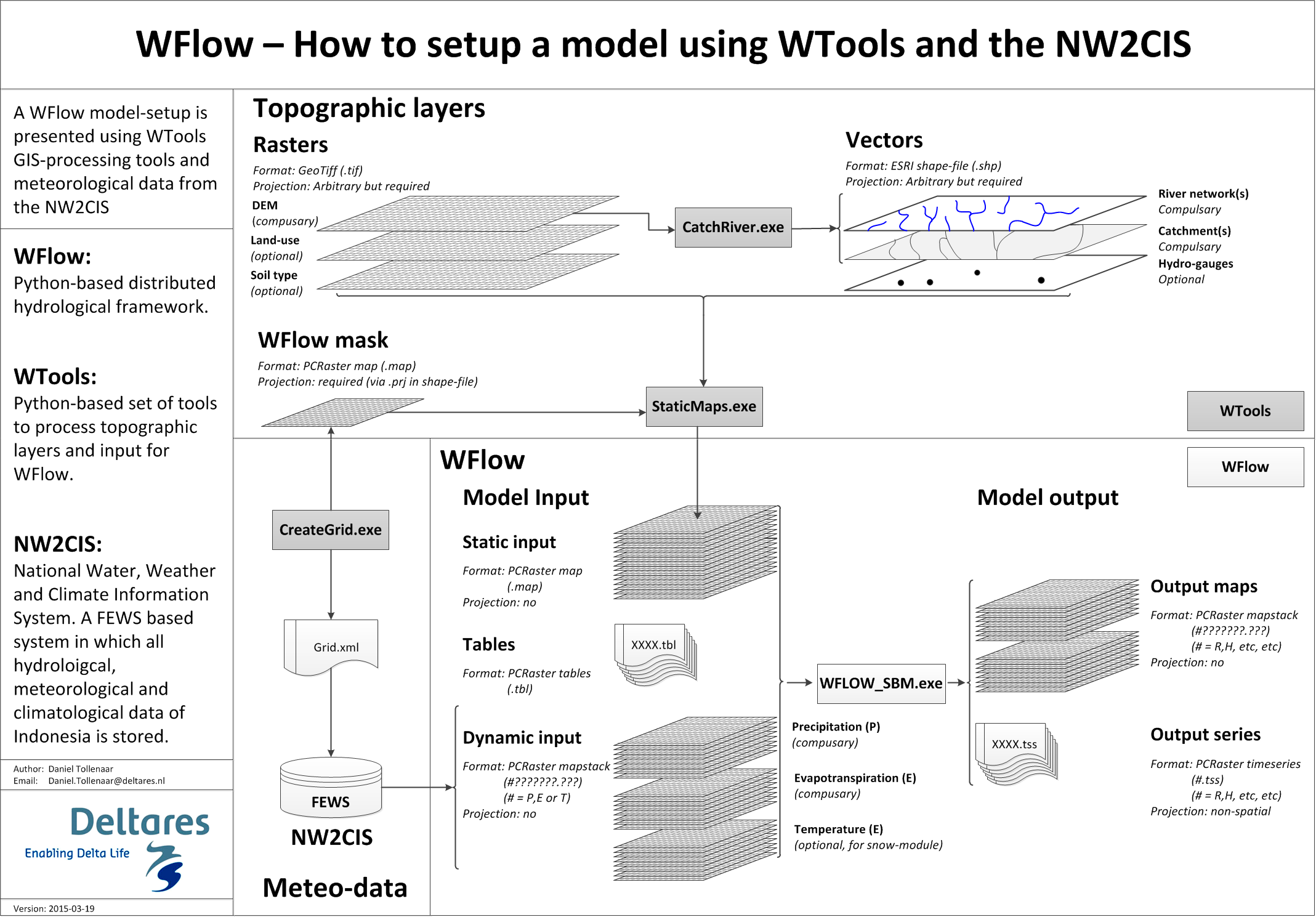


Figure – How to derive a WFlow model with WTools

The current version of WTools (19-03-2015) allows automatic generation of WFlow *static input* via ‘command-line tools’. It also simplifies the generation of *dynamic input*, as it generates a grid-definition for FEWS.

* Improve StaticMaps.exe to include reclassification of raster input, etc, etc.
* Automatic integration of WFlow models in FEWS
* Automatic coupling of WFlow with Ribasim and Sobek
* Integration of WTools in QGIS and DeltaShell

## Installation

The folder WTOOLS consists of:

* *\wtools:* containing the WTools executables
* *\test\_citarum:* this folder is to be used to test the tools and play around with Citarum data to understand how the tools work
* *\documentation:* this folder contains documentation
* *README.txt*: A quick startup guide
* *StaticMaps.ini:* an example ini-file to be used for running StaticMaps.exe (see section B.5)

The folder WTOOLS can be located anywhere on your hard-drive. To run the programs PCRaster and QGIS should be installed. For the installation of PCRaster follow these steps:

1. Download PCRaster (see: <http://pcraster.geo.uu.nl/downloads/latest-release/>; last visited 22 March 2015).
2. Put the PCRaster folder anywhere on your hard-disk

Now you have to update your environment variables:

1. Right-click on Start, and then Open Windows Explorer.
2. Right-click on Computer, then Properties.
3. In the System Control Panel, clickAdvanced System Settings.
4. Click the Advanced tab.
5. ClickEnvironment Variables (see Figure 2)
6. Be careful with the following two steps, and do not delete or modify one or more of the existing entries (unless you really know what you are doing)
7. In the User variablessection, either select the PATH variable and click edit, or add a new variable and name it PATH (see Figure 2)
8. Add the path of the bin directory of the extracted PCRaster (see Figure 2) and the QGIS package to the Variable value field, e.g.  
   C:\Program Files\pcraster-4.0.2\_x86-64\bin; C:\Program Files\QGIS Valmiera\bin\;

Add the path at the beginning of the field. Note the extra semicolon at the end, it is required to separate various entries. Do not add spaces around the semicolon.

1. Similar to step 8 add a *gdal\_data* variable. This variable should point to the file *gcs.csv*, used for map transformations. Fill for the *Variable value,* e.g:

c:\Program Files\QGIS Valmiera\share\gdal\;

1. Reboot Windows. You should now be able to run on command prompt: *pcrcalc, gdal\_translate, gdalwarp and gdal\_rasterize* (see Figure 3)*.*

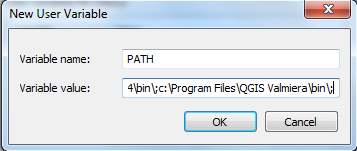
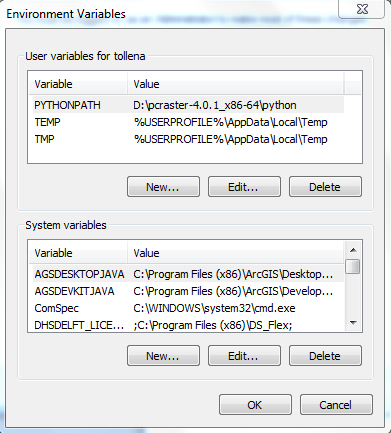
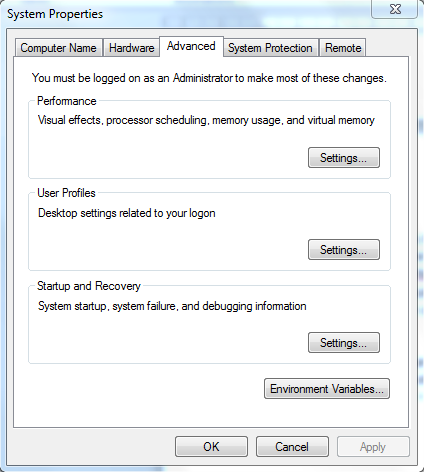


Figure - Adding environment variables

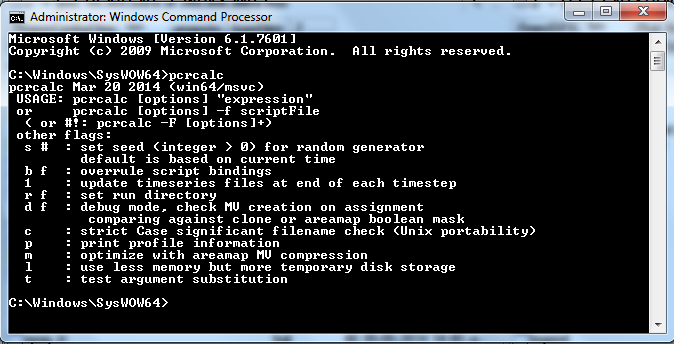


Figure – Running pcrcalc on the command-line

## CatchRiver.exe

CatchRiver.exe can be used to derive (a) river-network(s) and catchment(s). These two products will be stored in the folder *\CatchRiver* as ESRI shape-files. The program requires at least a DEM stored in a GeoTiff with a projection included (if not, no projection will be assigned to the output). The following options can be set:

*-*d *rasterfile (GeoTiff)*

Raster-file with digital elevation model. Projection definition preferred (will be assigned to output shape-files)

-l *shape-file (ESRI shape-file)*

(Optional) shape-file with polylines (rivers\drains) which will be burned in raster before catchment deliniation

-p *shape-file (ESRI shape-file)*

(Optional) shape-file with points (outlets) which will be burned in raster before catchment delineation

-F *integer*

(Optional) factor by which DEM will be rescaled before it is processed into a river network. Using a higher scale-factor will speed-up process-time, but reduce the scale of the output (default=1).

-O *integer*

(Optional) minimum strahler-order which will be included in the output river shape-file (default=3)

-s (Optional) setting to snap points (-p) to lines (-l) prior to burning (default=no)

-R *shape-file (ESRI shape-file)*

(Optional) name for output shape-file (default=river.shp)

-C *shape-file (ESRI shape-file)*

(Optional) name for output shape-file (default=catchment.shp)

-B *integer*

(Optional) burn value which will be used to lower dem at polylines, points and areas if -l, -p or -a are set

-S (Optional) if set, the local drainage direction network (ldd) will not be recreated. Use this option e.g. if you want to change output-names, or strahler-order and your dem, burn layers and burn-value haven’t changed (-l, -p, -a, -B)

-K (Optional) setting to keep all rivers in the domain of the input DEM (default=no)

Example 1:

*wtools\CatchRiver.exe -d input\srtm\_Citarum\_90m.tif -O 6 -F 3*

rivers will be derived on a scaled version of the dem (180x180 meter instead of 90m). Only rivers with a strahler-order larger than 6 will be included in the output shape-file

Example 2:

*wtools\CatchRiver.exe -d input\srtm\_Citarum\_90m.tif -p input\outlets.shp -l input\lines.shp -R river\_burned.shp -C catchment\_burned.shp -O 6*

Prior to the generation of the river network, the outlets and lines will be burned to the DEM. The river network will be stored in *\CatchRiver\river\_burned.shp* and the catchment in *\CatchRiver\catchment\_burned.shp* .Only rivers with a strahler-order larger than 6 will be included in the output shape-file

Example 3:

*wtools\CatchRiver.exe -d input\srtm\_Citarum\_90m.tif -p input\outlets.shp -l input\lines.shp -R river\_burned\_snap.shp -C catchment\_burned\_snap.shp -O 6 -s*

Same as example two, but prior to the burning process, the points will be ‘snapped’ on the lines.

## CreateGrid.exe

CreateGrid.exe is used to derive the grid of WFlow. It generates output in the folder \mask. Here the mask.map file can be found, which will be used to clip the topographical layers with the StaticMaps program. The StaticMaps program will use the mask.shp to read the model projection. In Grid.xml the mask grid-definition can be found which can be used to generate dynamic input from FEWS. The following options have to be set:

*-*f *ESRI shape-file or rasterfile (GeoTiff)*

Input-file from which extent will be derived. Note, the extent will be derived from the file-properties, not from the data-extent.

-c *integer*

Cellsize of the output in the units of the input-file (metres of degrees).

Example:

*CreateGrid.exe -f CatchRiver\catchment\_burned\_snap.shp -c 0.01*

For the extent of the file *CatchRiver\catchment\_burned\_snap.shp (projected in WGS84, lat lon) a grid will be generated with a cellsize of 0.01* *(degrees).*

## StaticMaps.exe

StaticMaps.exe is used to generate all static input for WFlow in the \staticmaps folder. The following options can be set:

*-*i *text-file (see staticmaps.ini as example)*

File containing settings for StaticMaps.exe

-r *polyline-file (ESRI shape-file)*

File containing the river network(s) for WFlow

-c *polygon-file (ESRI shape-file)*

File containing the catchment(s) for WFlow

-d *rasterfile (GeoTiff)*

File containing the DEM for WFlow

-g *point-file (ESRI shape-file)*

(Optional)file containing gauges for WFlow. If not specified, only the outlet will be provided as gauge.

-l *rasterfile (GeoTiff)*

(Optional) file containing the land-use land-cover data to be used in WFlow. If not specified, one map will be generated with one class (1)

-s *rasterfile (GeoTiff)*

(Optional) file containing the soil type data to be used in WFlow. If not specified, one map will be generated with one class (1)

-C (Optional) option to remove all .xml files from the \staticmaps folder (default=no)

-A (Optional) option to burn catchments “all touching”. All cells inside or crossed will be assigned to the catchment. This option can be used when a large cell-size is chosen and the catchment itself is narrow. When a catchment is ‘narrow’ and the cell-size is large, the option can avoid ‘broken’ drainage paths. (default=no)

Example 1:

*StaticMaps.exe -i StaticMaps.ini -d input\srtm\_Citarum\_90m.tif -r CatchRiver\river\_burned\_snap.shp -c CatchRiver\catchment\_burned\_snap.shp -C*

Static maps will be derived from an ini-file (-i), dem raster (-d), river network shape-file (-r) and catchment shape-file (-c). After processing, all xml-files in the \staticmaps folder will be deleted (-C)

Example 2:

*StaticMaps.exe -i StaticMaps.ini -d input\srtm\_Citarum\_90m.tif -r CatchRiver\river\_burned\_snap.shp -c CatchRiver\catchment\_burned\_snap.shp -l input\BIG\_LandUse\_0507\_Citarum\_resampled.tif -C -A*

Same as example 1. Instead of one class (1) the staticmaps\wflow\_landuse.map will now have the classes from a land-use raster (-l). The catchment is rasterized ‘all touched’ (-A)