TARGET v1.0 user manual

1. License and code availability

TARGET is distributed under the Creative Commons Attribution-NonCommercial-ShareAlike 2.0 Generic (CC BY-NC-SA 2.0) license. TARGET code cannot be used for commercial purposes. The code can be downloaded from github at…

<https://bitbucket.org/mothlight/target_java>

(And add a Zenodo link.)

2. System requirements

* Java JRE 8 or greater
* NetCDF 4.6.11 (jar file is provided in repository)
* SLF4J 1.7.14 (jar file is provided in repository)
* python-sympy

Code has been tested on Ubuntu 16.04 but should compile and run on any OS on any average system with adequate memory and CPU.

To test whether python-sympy is installed, try :

Target-java/target\_java$ python TbRurSolver.py

If you receive:

ImportError: No module named 'sympy'

then python-sympy needs to be installed (i.e. pip install sympy)

3. Installation

Java code can be downloaded from Github. Or by cloning the Git repository.

git clone https://mothlight@bitbucket.org/mothlight/target\_java.git

Project was developed in Eclipse Neon.3 and should open and run in Eclipse.

Or can be built and run from the command line like:

$ cd mothlight-target\_java-85fc86f776fa/   
$ cd src/   
[$](mailto:kerryn@mu00038561:.../mothlight-target_java-85fc86f776fa/src$) cd Target/   
[$](mailto:kerryn@mu00038561:.../src/Target$) javac -cp ../../netcdfAll-4.6.11.jar \*.java   
[$](mailto:kerryn@mu00038561:.../src/Target$) cd ..   
[$](mailto:kerryn@mu00038561:.../mothlight-target_java-85fc86f776fa/src$) java -cp ../netcAll-4.6.11.jar:. Target.RunToolkit /home/user/Target\_Runs/Scenario1/controlfiles/Scenario1/Scenario1 .txt

4. Input datafiles

*4.1 Meteorological data*

TARGET requires reference meteorological data to drive the model and calculate microscale urban air temperature. The following meteorological variables are required: incoming shortwave (solar) radiation, incoming longwave (terrestrial), relative humidity, reference wind speed (typically at 10 m), and air temperature. The user must define the height above ground of reference wind speed and air temperature measurements. Ideally, meteorological data should be representative of a nearby urban site. However, the nearest airport weather station will suffice. Ideally, the ground surface characteristics (i.e. % of plan area of model land cover types) below the reference weather (i.e within 125 m radius of station) station should also be defined. If this is unknown assume reference station land cover is 100% dry grass.

Meteorological data should be stored in csv format with the following column headers:

datetime – date and time, e.g. 1/1/2011 12:00

Ta – air temperature in degrees Celsius

RH – relative humidity in percentage

WS – wind speed in meters per second

P – atmospheric pressure in mb/hPa

Kd – incoming shortwave radiation in Wm-2

Ld – incoming longwave radiation in Wm-2

See example meteorological data file attached.

*4.2 Land cover data*

Land cover data should be provided for the entire domain. The planar fraction of land cover types and building heights and widths for each grid cell are defined in a csv file will the following format:

FID – cell identifier, unique numerical number for each grid point.

roof – fractional roof planar area

road – fractional road planar area

watr – factional water planar area

conc – fractional concrete planar area

Veg – fractional tree planar area

dry – fractional dry grass area

irr – fractional irrigated grass/low vegetation area

H – average building height (m)

W – average street width, distance between buildings (m)

These data can be derived at resolution of the user choice. A resolution of 30 – 50m is recommended to captures microscale effects. If the user has exiting raster land cover datasets, an easy way to tabulate these data is (1) define a fishnet with resolution 30 m and (2) use ArcGIS function “tabulate area”.

See example land cover data file for further guidance.

5. Model output

The model output is netCDF format. By default, all variables are written to the output. To suppress some variables, they can be added to the control file, as below

disableOutput=Utb,Fid,modUTaRef,TbRur,HttcCan,HttcUrbNew,TsurfWall,TsurfCan,TsurfHorz,Ucan

The variables Tac, Tmrt, and UTCI will always be outputted.

The domain dimensions are set in the control file, for example, as below giving the top left x,y coordinate, the resolution of each grid square, and the x,y dimensions of the domain.

DomainDim=344,235

latEdge=-37.505054

lonEdge=144.647901

latResolution=.00004294

lonResolution=.0021849

6. Executing the code

Code can be run from within Eclipse or from the command line, for example:

$ cd src/   
[$](mailto:kerryn@mu00038561:.../mothlight-target_java-85fc86f776fa/src$) cd Target/   
[$](mailto:kerryn@mu00038561:.../src/Target$) javac -cp ../../netcdfAll-4.6.11.jar \*.java   
[$](mailto:kerryn@mu00038561:.../src/Target$) cd ..   
[$](mailto:kerryn@mu00038561:.../mothlight-target_java-85fc86f776fa/src$) java -cp ../netcAll-4.6.11.jar:. Target.RunToolkit /home/user/Target\_Runs/Scenario1/controlfiles/Scenario1/Scenario1 .txt