User’s Manual for i-Tree Cool River Model

i-Tree Cool River

Program version 1.0

Manual updated Aug. 10, 2018

Description

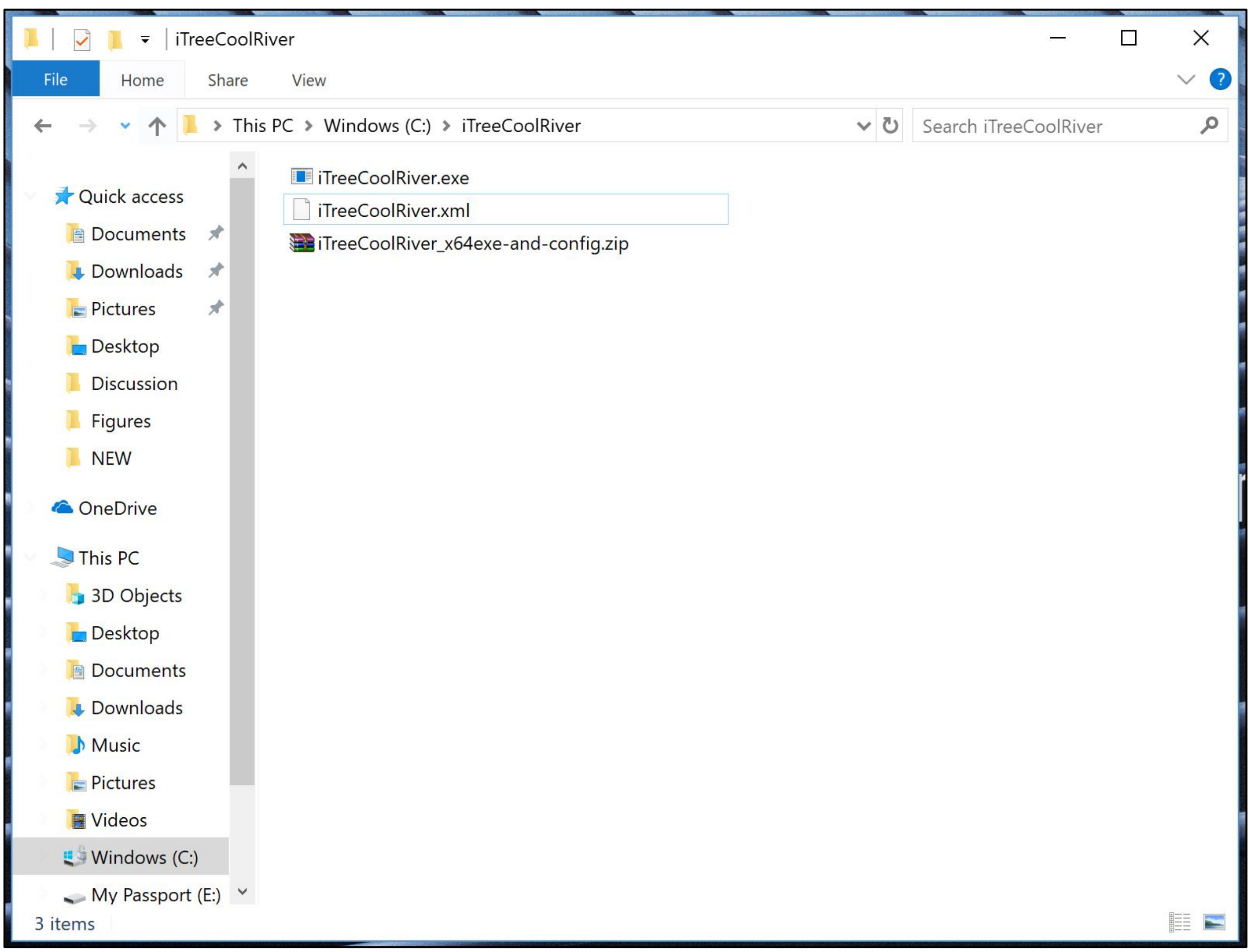
The i-Tree Cool River Model is a one-dimensional river temperature model in steady and unsteady modes in varying time and space using a combination of temperature change driven by the advection, dispersion, energy flux, and mixing process. The model applies upstream boundary condition, diurnal temperature, river geometric data, discharge data, and meteorological data to predict river temperature using a finite difference method.

The i-Tree Cool River Model is designed to allow for flexible shading factor algorithms, unsteady flow of storm sewers, as well as other energy flux and mixing processes. The i-Tree Cool River Model is an open-source model written in C++, and its package contains the C++ routines and an executable file for running the code, which can be downloaded from "www.itreetools.org/research_suite/coolriver" . The i-Tree Cool River Model C++ algorithms can be edited and recompiled with Visual Studio 2017 Community Edition, which is freeware. The outputs of running the model include the predicted river temperature, the volumes and temperatures of mixing processes, and the magnitude of energy fluxes. The current executable is built for x64 platforms. We also provided a Win32 executable.

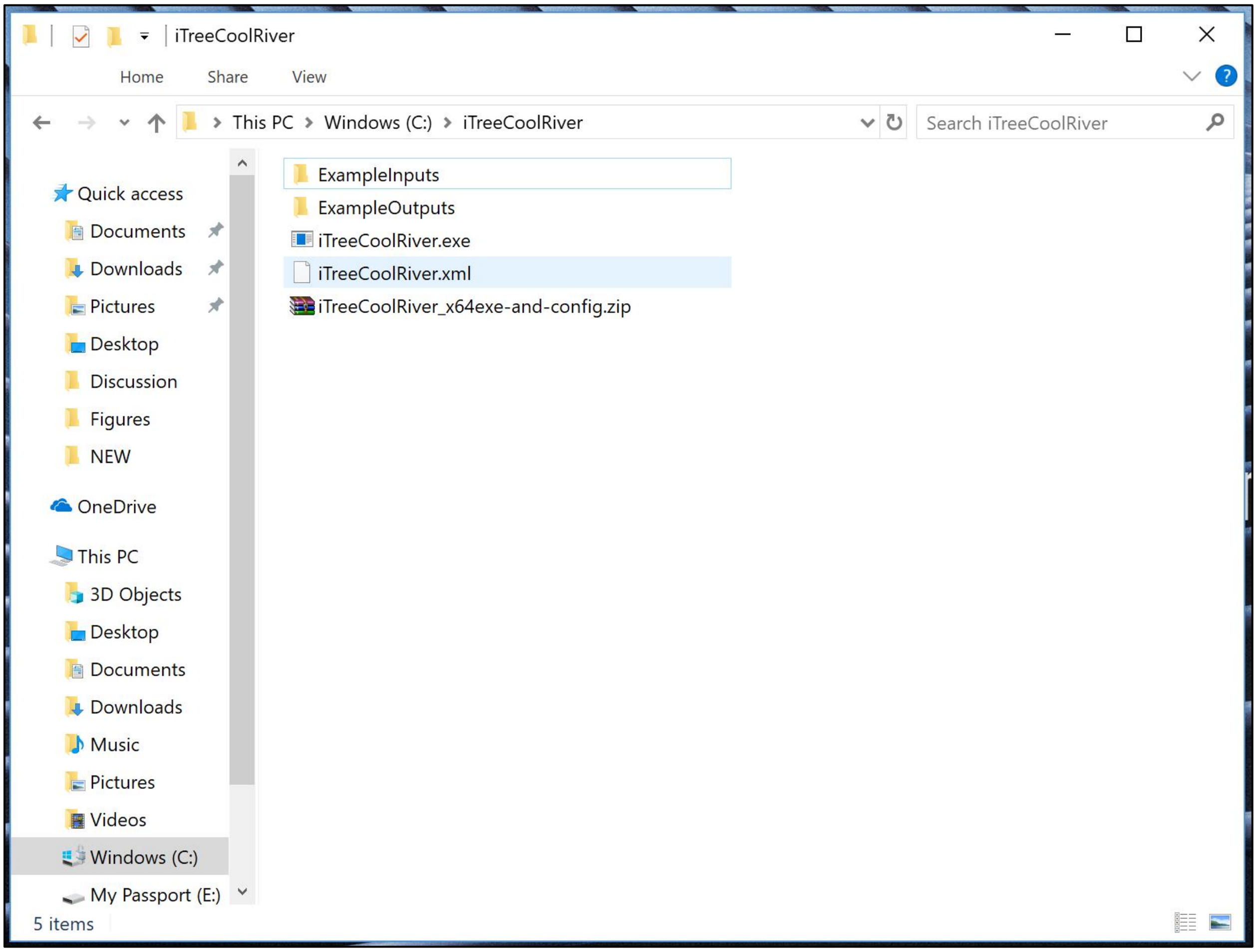
Running the i-Tree Cool River Model Application

We provide an application file (iTreeCoolRiver.exe) in the iTreeCoolRiver_x64/86exe folder. The model executable is called at the command line along with an extensible markup language (XML) file, which includes the required initial information in sample input files folder. The target platform for this application is Windows 10. If you are having trouble running this application for a different platform, consider creating a build for your platform (see Compile and Build iTreeCoolRiver via Visual Studio 2017 section). The easiest way to get the application "up and running" is to download the compressed iTreeCoolRiver.exe file (the iTreeCoolRiver_x64/86exe folder) and extract all to your C: drive. (If you choose a different location, it will be necessary to edit the Inputs and Outputs tag of the iTreeCoolRiver.xml config file accordingly and specify the correct config location when running the application. See Input Data section, Table 2).

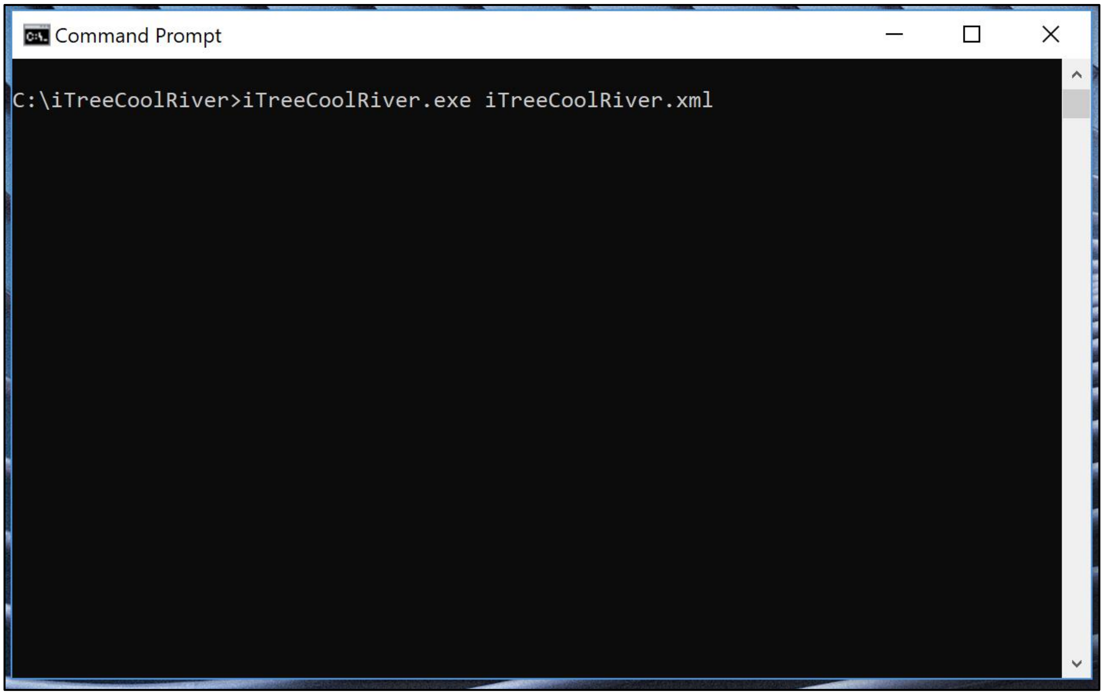
- Uncompress the "iTreeCoolRiver_x64exe-and-config.zip" file to "c:\iTreeCoolRiver



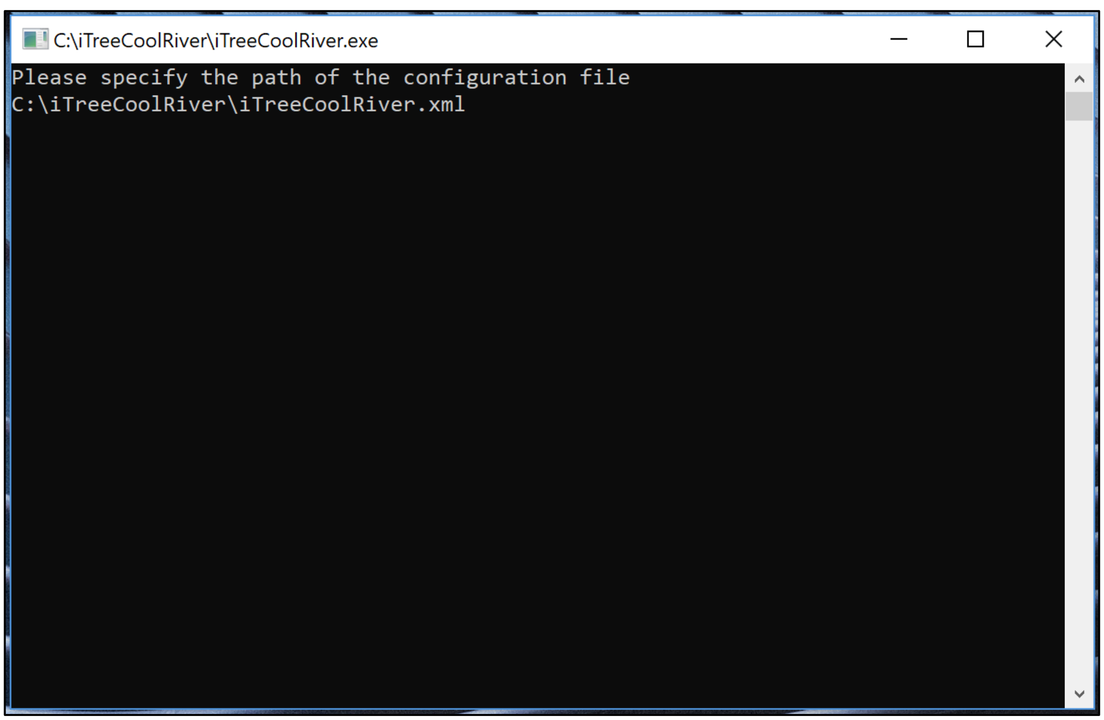
- Uncompress the “iTreeCoolRiver_SampleData.zip” file to “c:\iTreeCoolRiver”, creating “C:\iTreeCoolRiver\ExampleInputs” and C:\iTreeCoolRiver\ ExampleOutputs



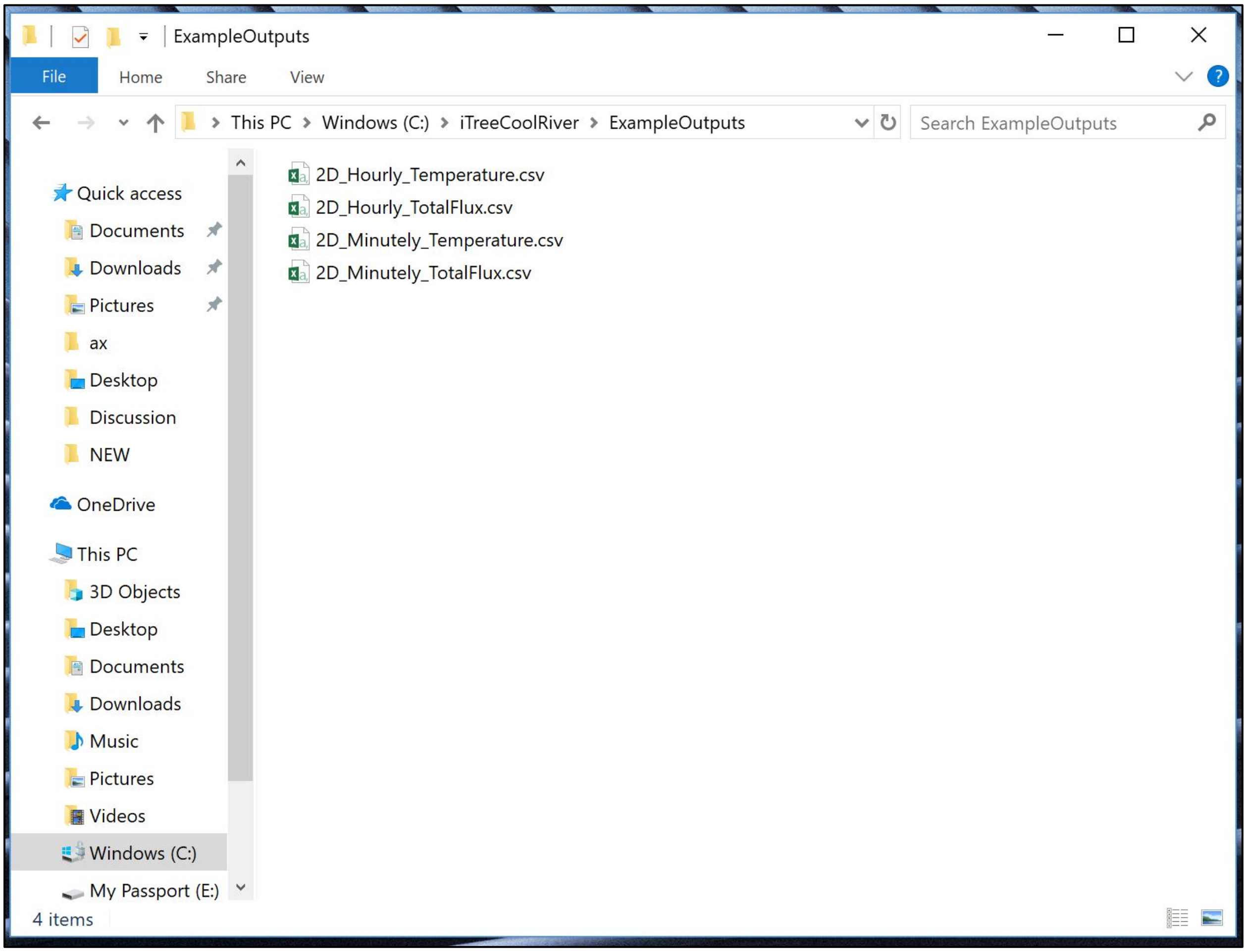
- To run the model, either
  - a) in a DOS Command Prompt navigate to “C:\ iTreeCoolRiver\” and type the name of the executable and the config file and its path: “C:\ iTreeCoolRiver> iTreeCoolRiver.exe C:\ iTreeCoolRiver\iTreeCoolRiver.xml”



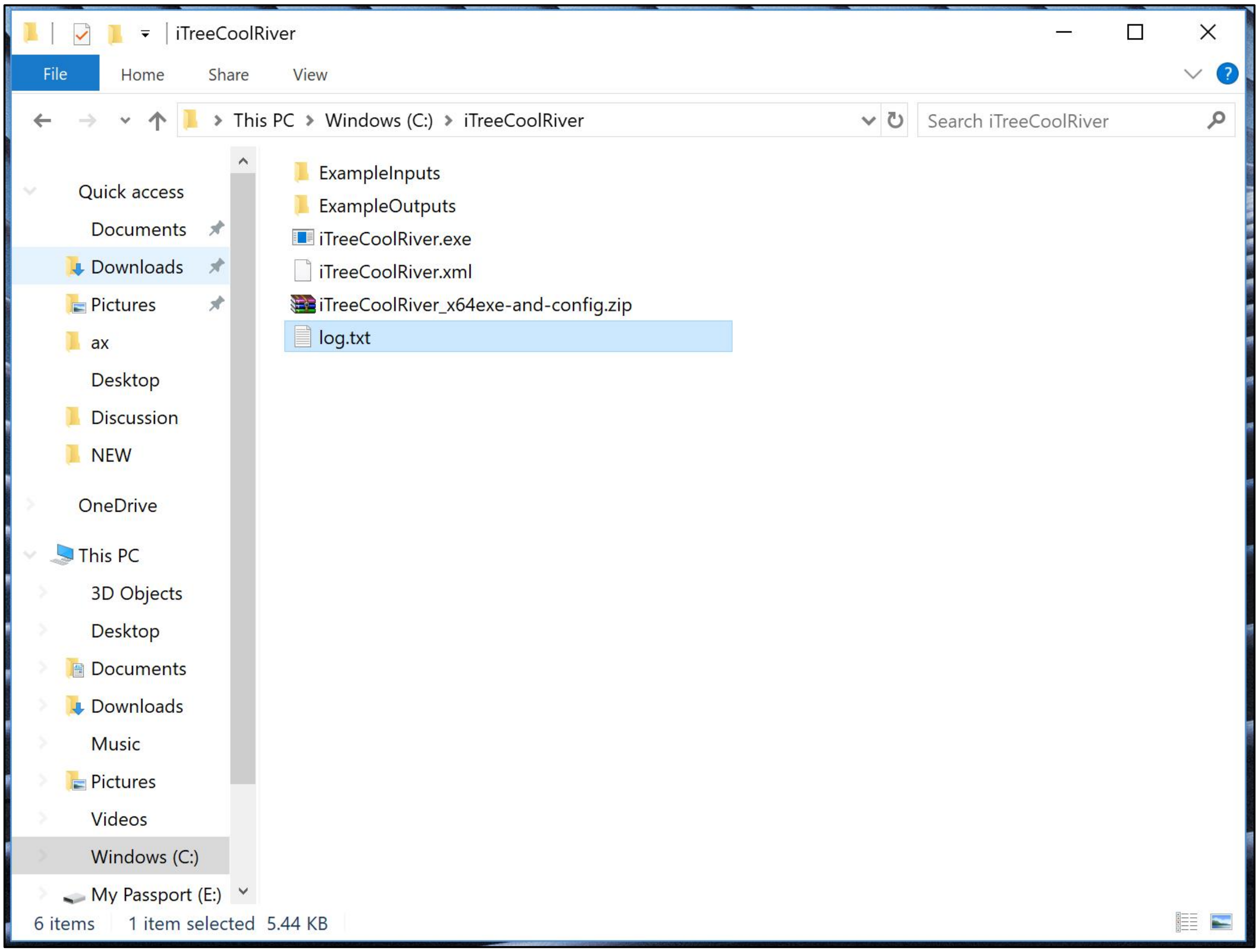
- b) in a Windows Explorer double click on the iTreeCoolRiver.exe file and a DOS Command Prompt will open, type the config file and path: "C:\ iTreeCoolRiver\iTreeCoolRiver.xml



- You will find the output of this run in the ExampleOutputs folder.



You will find a log file that may be useful in troubleshooting. It will be located in the working active directory. Since messages are appended to the file, it is best practice to delete the log after each run. For this example, the log file is located here:



Input data

The i-Tree Cool River Model uses a set of input data including upstream boundary condition, steady or unsteady discharge hydrograph for the river and lateral storm sewer inflows, groundwater and hyporheic exchange data, streambed temperature, meteorological data, etc. for the simulation process, which can be imported to the model using the DAT files (the input data for temperature modeling in time and space) and the XML file (which includes the initial necessary data for the beginning of the simulation). The i-Tree Cool River Model has a function to linearly interpolate the input data based on the defined Δx and Δt and the input files can be imported at different intervals. The name of the required input files and a brief description for each input file represented in Table 1. Input files contain three TXT files including the impervious cover, land cover, and tree cover which are converted from raster to ASCⅡ and are not effective in simulation process of the i-Tree Cool River Model. Table 2 also represents the description of the initial parameters which are imported to the model in the XML file. The other parameters which are not introduced in Table 2 are not effective in i-Tree Cool River Model simulation process.

Table 1. List of the input files required for the simulation process of the i-Tree Cool River Model

Input file	**The parameter name**	**Description**
BedData.dat	Number	The number of the observations indicates the locations of the observed streambed data.
	Distance (m)	Distances through the river reach where the streambed observations are recorded.
	Depth of Measurement (m)	Depth at which groundwater temperatures are recorded in each cross section
	GW_Temp (°C)	Groundwater temperature in downstream.
	Type	Bed-sediment type which can be clay, silt, sand, or gravel.
	Horizontal Bed Conductivity (mm/s)	Horizontal effective thermal conductivity in each observed cross-section.
	Bed Particle Size (mm)	Bed particle size (Bedient and Huber, 1992, Rosgen, 1996) in the observed location.

	Embeddedness (fraction)	Embeddedness in each considered cross section.
DEM.txt	Elevation data for calculating slope and aspect for calculating the hillslope effect on energy flux which can be converted from raster file to ASCII in Arc Map. The raw DEM data can be downloaded from the National Map Viewer.	
Discharge.dat	Number	The number of the observations indicates the locations of the observed groundwater data.
	Distance (m)	Distances through the river reach where the magnitude of groundwater flow is recorded
	Q_GW (cms)	Groundwater discharge.
Inflow.dat*	Number	The number of the observations which indicates the number of the time steps for the hydrographs of the river and lateral inflows.
	Inflow Rate Storm (cms)	Discharge rates of the river in upstream at each timestep defining the hydrograph in steady or unsteady mode.
	Inflow Temp Storm (°C)	Observed stream temperatures corresponding to the river hydrograph timesteps in upstream.
	Inflow Rate 1 (cms)	Discharge rates of the lateral storm sewer inflow at each timestep for the first location defining the hydrograph in steady or unsteady mode.
	Inflow Temp 1 (°C)	Observed stream temperatures corresponding to the first lateral storm sewer inflow hydrograph timesteps.
	Inflow Rate 2 (cms)	Discharge rates of the lateral storm sewer inflow at each timestep for the second location defining the hydrograph in steady or unsteady mode.
	Inflow Temp 2 (°C)**	Observed stream temperatures corresponding to the second lateral storm sewer inflow hydrograph timesteps.

	* The First row of the input file below the headings should be considered as the location of each hydrograph. The river's hydrograph gets 1 m indicating the upstream and other lateral inflows receive their own location from the upstream. ** The number of lateral inflows can be changed in the code by the user.	
Morphology.dat	Number	The number of the observations indicates the locations of the measured geomorphic data.
	Distance (m)	Distances through the river reach corresponding with the cross sections where the geomorphic data are recorded.
	Area (m^2)*	Cross-sectional wetted area of the river channel.
	Width (m)	Stream width.
	Depth (m)*	Wetted depth.
	Discharge (cms)	River discharge magnitude at the location where the geometric data are measured.
	Slope	Channel Slope
	Row#**	The row number in the DEM file where the cross-section is located.
	Column#**	The column number in the DEM file where the cross-section is located.
	Longitude (deg)**	Longitude of the cross-section in the geographic coordinate system.
	Latitude (deg)**	Latitude of the cross-section in the geographic coordinate system.
	Z (m)	The elevation of the cross-section.
	* Measured area and depth are required for running the Crank-Nicolson method in steady state. These parameters are calculated based on the depth using the Newton-Raphson root finding iterative method in explicit finite difference method. ** These input data are required for calculating the slope and aspect of	

	each cell to apply the values on hillslope effect and the shortwave radiation. In case of using fixed magnitudes for the shading factor and view-to-sky values, these values are not effective in the simulation process.	
Shading.dat*	Number	The number of the observations reflecting the locations of the measured shading information.
	Distance (m)	Distances through the river reach corresponding with the cross sections where the shading information are recorded.
	EastBankH (m)	The height of the bankfull at the measured cross section on the Eastside.
	EastTreeH (m)	The height of the canopy at the measured cross section on the Eastside.
	EastBuildingH (m)	The height of the building at the measured cross section on the Eastside.
	EastBankDist (m)	Distance from the bankfull to the edge of the water at the measured cross section on the Eastside.
	EastCanDist (m)	Distance from the canopy to the edge of the water at the measured cross section on the Eastside.
	EastBuildingDist (m)	Distance from the building to edge of the water at the measured cross section on the Eastside.
	EastBufferW (m)	The magnitude of the canopy buffer at the location of the measured cross section on the Westside
	WestBankH (m)	The height of the bankfull at the measured cross section in the Westside.
	WestTreeH (m)	The height of the canopy at the measured cross section on the Westside.

	WestBuildingH (m)	The height of the building at the measured cross section on the Westside.
	WestBankDist (m)	Distance from the bankfull to the edge of the water at the measured cross section on the Westside.
	WestCanDist (m)	Distance from the canopy to the edge of the water at the measured cross section on the Westside.
	WestBuildingDist (m)	Distance from the building to edge of the water at the measured cross section on the Westside.
	WestBufferW (m)	The magnitude of the canopy buffer at the location of the measured cross section on the Westside
	Elevation (m)	The elevation of the cross-section.
	StreamAzimuth (deg)	The stream azimuth at the location of the measured cross section.
	* These input data are required for calculating the topographic, canopy (tree), and building shade angle and view-to-sky factor to apply the values to hillslope effect and the shortwave radiation. In case of using fixed magnitudes for the shading factor and view-to-sky values, these values are not effective in the simulation process.	
ShadingPercent.dat*	Number	The number of the observations reflecting the locations of the shading factors.
	Distance (m)	Distances through the river reach corresponding with the cross sections where the shading factor and the view-to-sky values are calculated.
	ShadeFactor	The value of shading factor in the desired cross-section.
	View-to-Sky	The value of View-to-Sky in the desired cross-section which is 1-shadingFactor

	* In case the topographic, canopy, and building heights and distances are considered for shading calculations, the magnitude of ShadingFactor and View-to-Sky are not effective in the simulation process.	
SolarRadiation.dat* The number of entries in this file should match the attribute value of totTime in the config file (see Table 2)	yyyymmdd	The date of the simulation period.
	Hr: Min: Sec	The time of the simulation period.
	DirSW (W/m^2)	Direct shortwave radiation at the edge of the atmosphere.
	DiffSW (W/m^2)	Diffuse shortwave radiation at the edge of the atmosphere.
	* Source: National Solar Radiation Database (NSRDB)	
Time.dat	Number	The number of the time steps.
	Time (s)	The desired time steps for the output intervals.
Weather.dat* The number of entries in this file should match the attribute value of totTime in the config file (see Table 2)	yyyymmdd	The date of the simulation period.
	Hr: Min: Sec	The time of the simulation period.
	Tair (F)	Air temperature.
	WndSpd (m/s)	Wind speed.
	Precip (m/h)	Precipitation rate.
	Cloudiness	The magnitude of the cloudiness.
	Humidity	Relative humidity.
	obsT_x0 (°C)	Observed river temperature in the upstream.
	sedT (°C)	Riverbed temperature.
	* National Center for Environmental Information	

Table 2. List of the parameters which should be specified in the XML file for the simulation process of the i-Tree Cool River Model

Key tags	**The parameter name**	**Description**
Inputs	Input_Folder	The address of the input folder directory that could be different for every machine.
Outputs	Output_Folder	The address of the output folder directory that could be different for every machine.
Spatial_Domain	cell_size	The magnitude of the cell size of the spatial input data which can be specified in the resampling process using the Arc Map.
	rows	A number of the rows of the spatial input file.
	cols	A number of columns of the spatial input file.
Temporal_Domain	starttime	The date of the starting of the simulation in the yyyymmdd format.
	endtime	The date of the finishing of the simulation in the yyyymmdd format.
	Timeinterval (h)	The intervals of the timesteps in the simulation process.
	startingHour	The starting hour of the simulation process.
	SWMethod	The code of the desired method for calculating the shortwave and longwave radiation energy fluxes: • Number 1 is for solving shortwave and longwave radiation fluxes based on the sky-to-view factor shading angles calculated from the provided heights and distances of the hillslope, vegetation, and building. • Number 2 is for solving shortwave and longwave radiation fluxes based on the fixed values for sky-to-view and shading

		factors.
	initialTemp	The observed river temperature of the upstream cross-section at the first timestep.
	canDensity	The density of the vegetation in the study area.
	roughness	The Manning's roughness coefficient (n).
	LAI	The magnitude of the leaf area index in the study area.
Shading_inputs	tarLat	Representative latitude of the study area.
	tarLong	Representative longitude of the study area.
	tarRow	The row number of the representative cell of the study area.
	tarCol	The column number of the representative cell of the study area.
	standardMeridian	The meridian ranking of the time zone in the study area.
	totDist	The total distance of the river.
	calcMethod	The
	depthOfBed	The code of the desired method for selecting the numerical methodology: • Number 1 is the Crank-Nicolson finite difference method. • Number 3 is for the Explicit finite difference method.
	(h)	A total number of the simulation process in an hour.
	sensMethod	The code of the desired method for calculating the sensible heat flux: • Number 1 is based on:

		• Number 2 is based on: • Number 3 is based on:
Unsteady_inputs	dx	The considered distance (Δx) for numerical finite difference method.
	dt	The considered timestep (Δt) for numerical finite difference method.

Outputs

After the simulation process, the i-Tree Cool River Model generates four comma-separated-value (CSV) files for the predicted temperatures and the net value of the heat flux in minute and hour based intervals (Table 3). The CSV files are two-dimensional vectors in which each row shows each meter of the river reach and each column represents each timestep of the running process. The user can add extra output files in the code as well.

Following figures reflect the possible data visualizations can be generated using the outputs of the i-Tree Cool River Model:

Table 3. List of the default outputs created by simulating the i-Tree Cool River Model

Output file name	**Description**
2D_Hourly_Temperature.csv	A two-dimensional (2D) matrix including the hourly simulated river temperature. The columns reflect the timestep (ΔT) and the rows show the intervals (ΔX).
2D_Hourly_TotalFlux	A two-dimensional (2D) matrix including the hourly simulated total flux as a combination of the longwave, shortwave, latent, sensible, and sediment fluxes. The columns reflect the timestep (ΔT) and the rows show the intervals (ΔX).
2D_Minutely_Temperature	A two-dimensional (2D) matrix including the simulated river temperature. The timestep in this output file is one minute and the spatial intervals are one meter. The columns reflect the timestep (ΔT) and the rows show the intervals (ΔX).
2D_Minutely_TotalFlux.csv	A two-dimensional (2D) matrix including the simulated total flux as a combination of the longwave, shortwave, latent, sensible, and sediment fluxes. The timestep in this output file is one minute and the spatial intervals are one meter. The columns reflect the timestep (ΔT) and the rows show the intervals (ΔX).

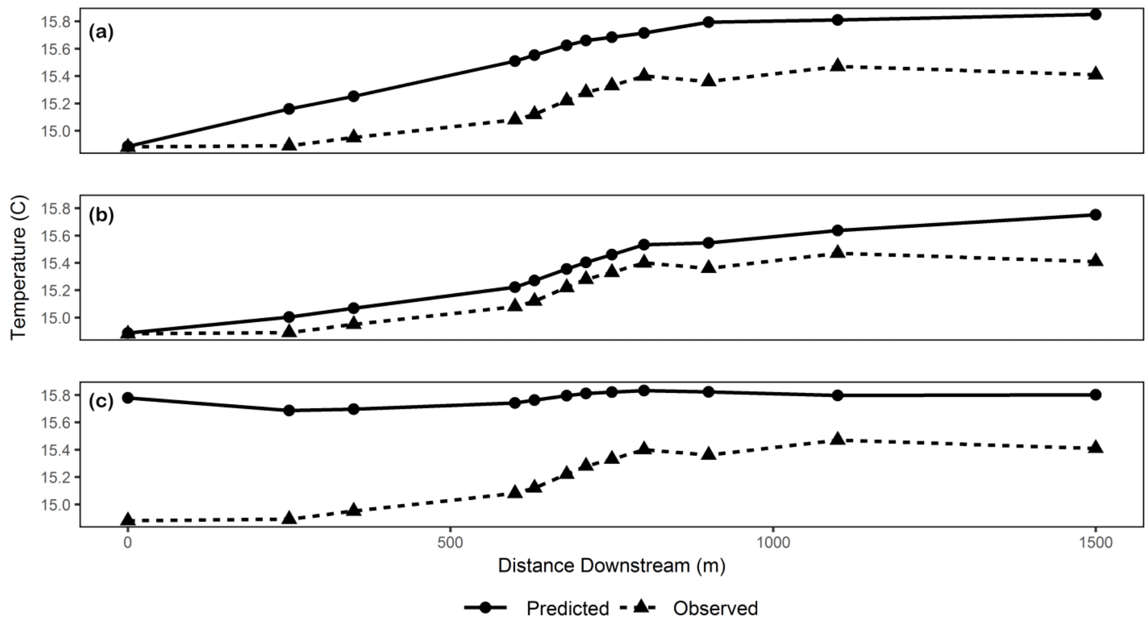


Fig. 1. Observed and predicted average river temperatures in unsteady state using the scenarios reflecting the (a) no shading effect, (b) no groundwater and hyporheic exchange inflows, and (c) no observed boundary condition.

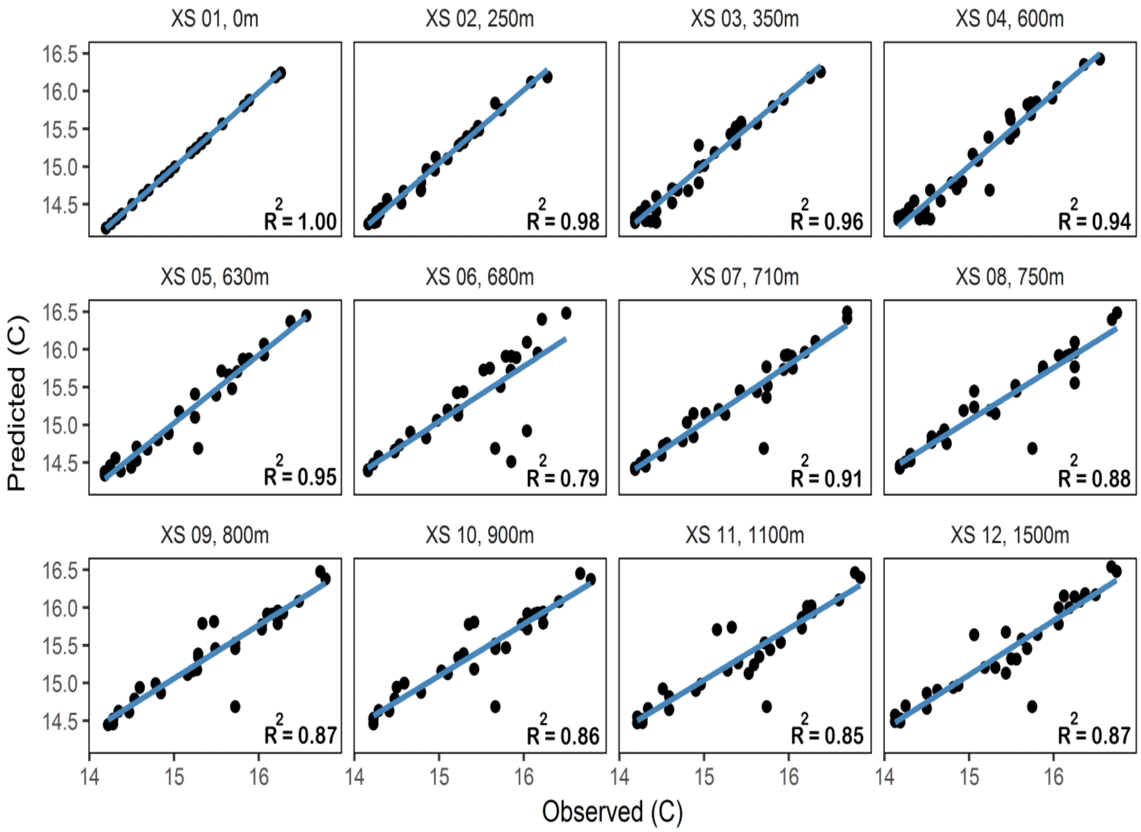


Fig. 2. Scatterplots of observed and predicted river temperature for the 12 cross sections (XS) in unsteady state, indicating their river stations form 0 m to 1500 m. The magnitude of the R^2 for each cross section is shown in the plot.

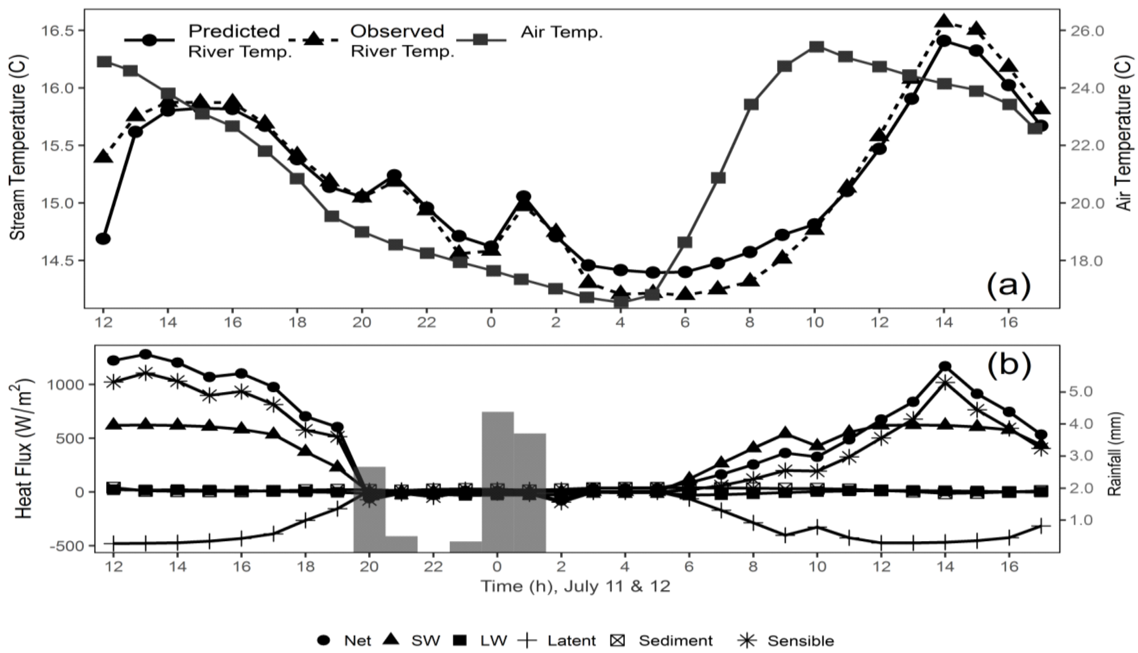


Fig. 3. (a) The air temperature and observed and predicted river temperatures in Sawmill Creek representing the average river temperature through time in unsteady state. (b) Average predicted heat fluxes of Sawmill Creek in time and the rainfall amount.

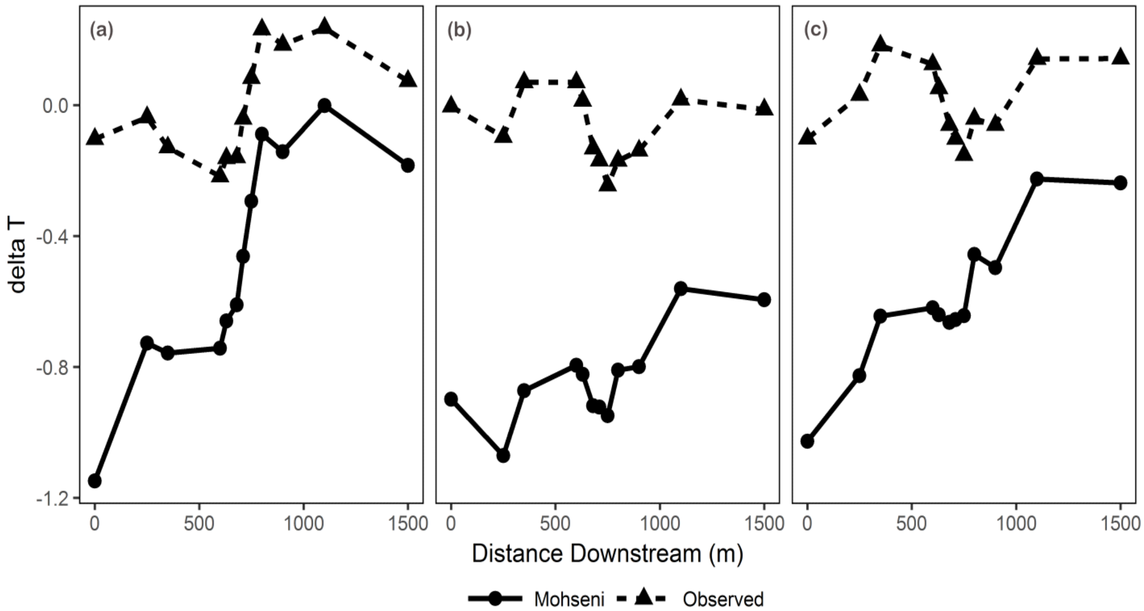


Fig. 4. Temperature differences between the observed and predicted river temperature using the Mohsni et al. (1998) boundary condition (Mohseni), and predicted river temperature using the observed river data (Observed). Three plots represent the differences for the (a) nighttime, (b) daytime, and (c) overall averages.

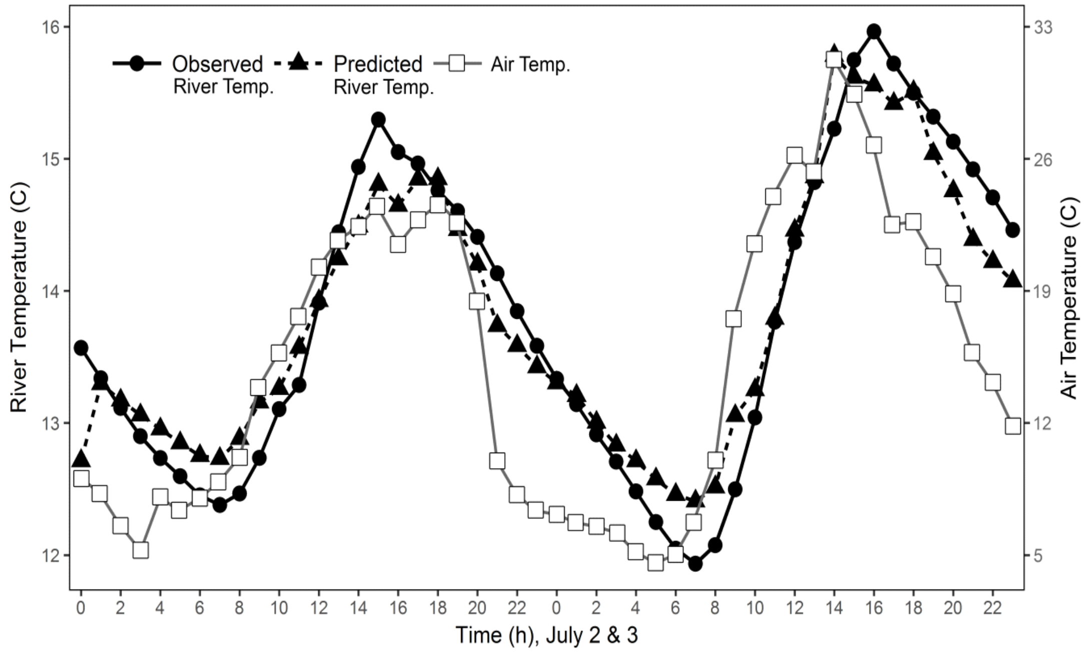
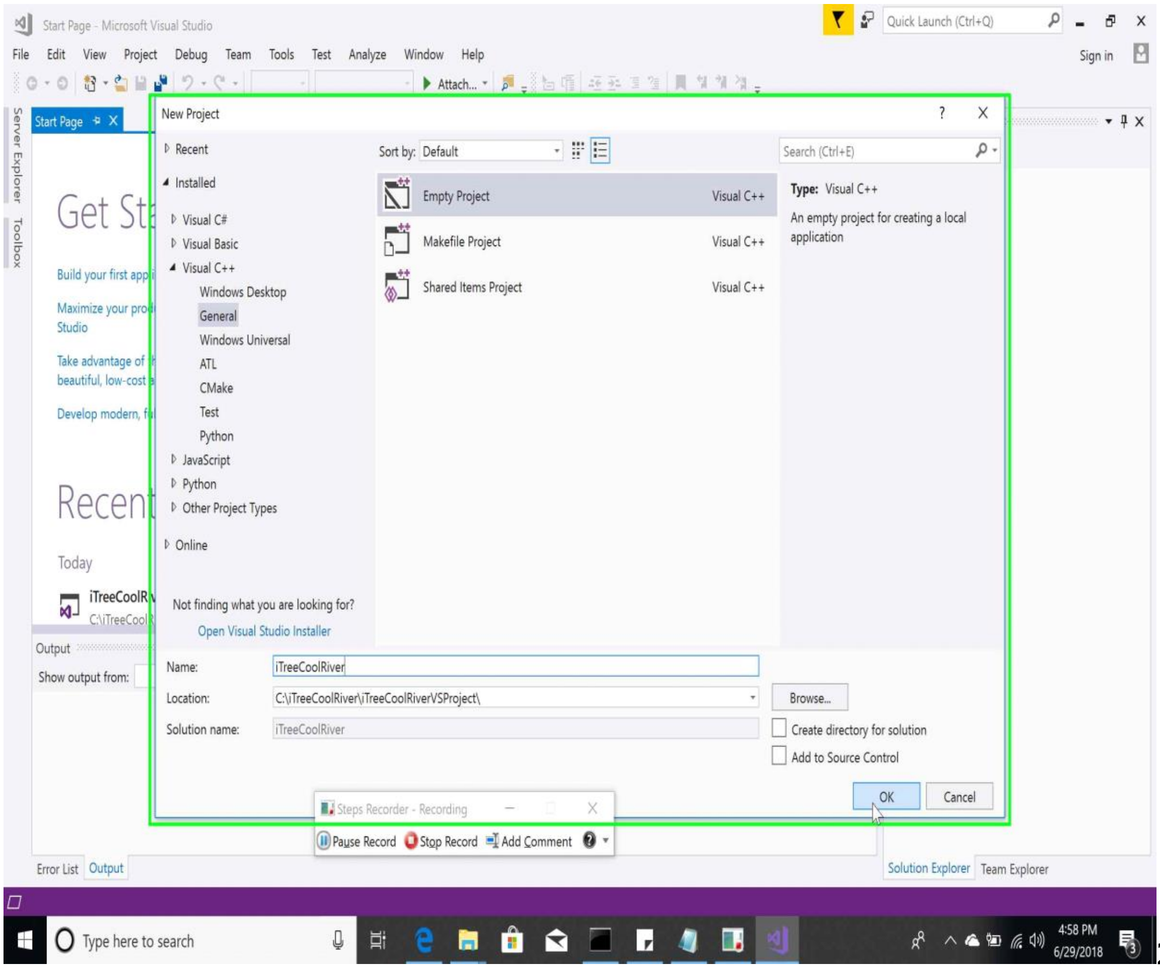


Fig. 5. The air temperature and average observed and predicted river temperatures through time in Sawmill Creek, NY in steady state.

Compile and Build iTreeCoolRiver via Visual Studio 2017

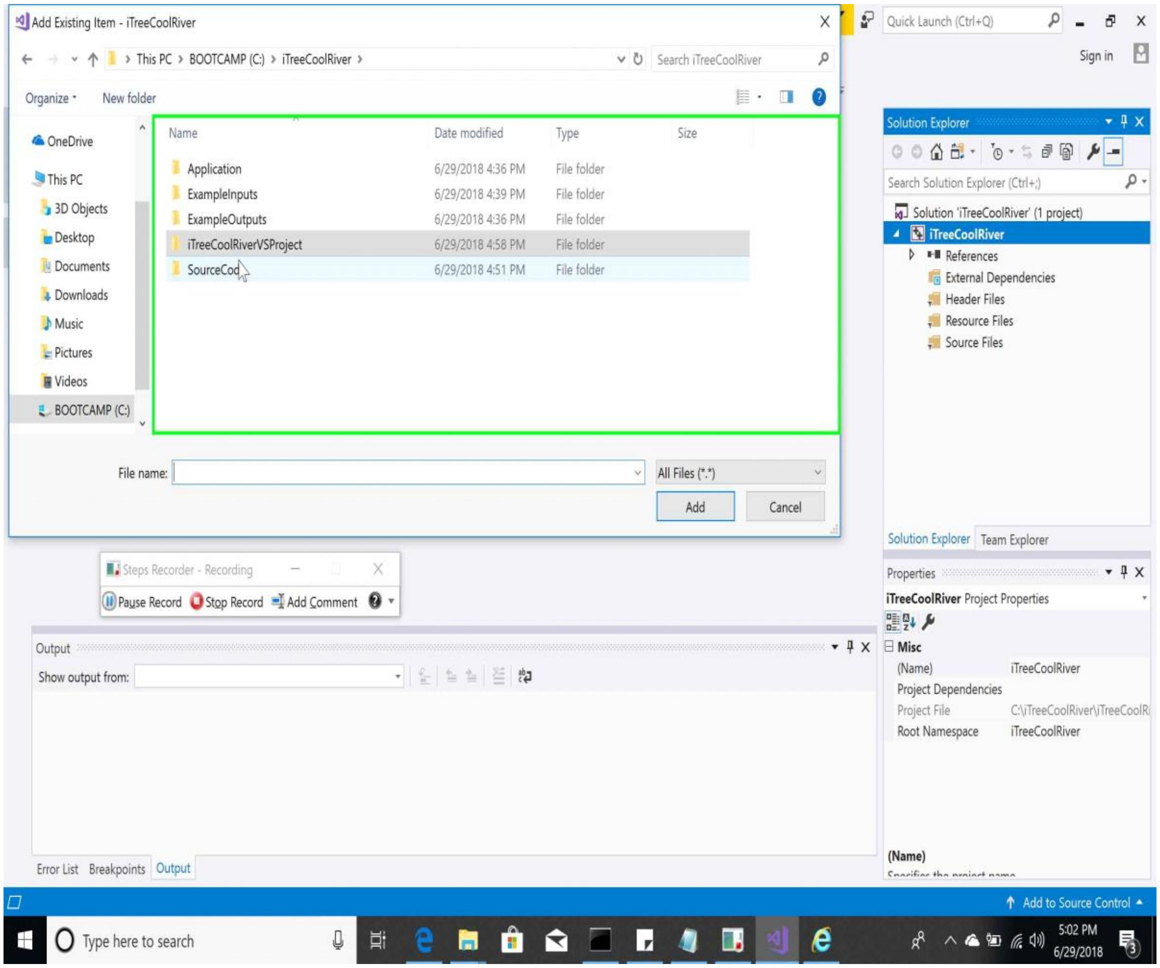
To compile and build iTreeCoolRiver in VisualStudio, follow these steps:

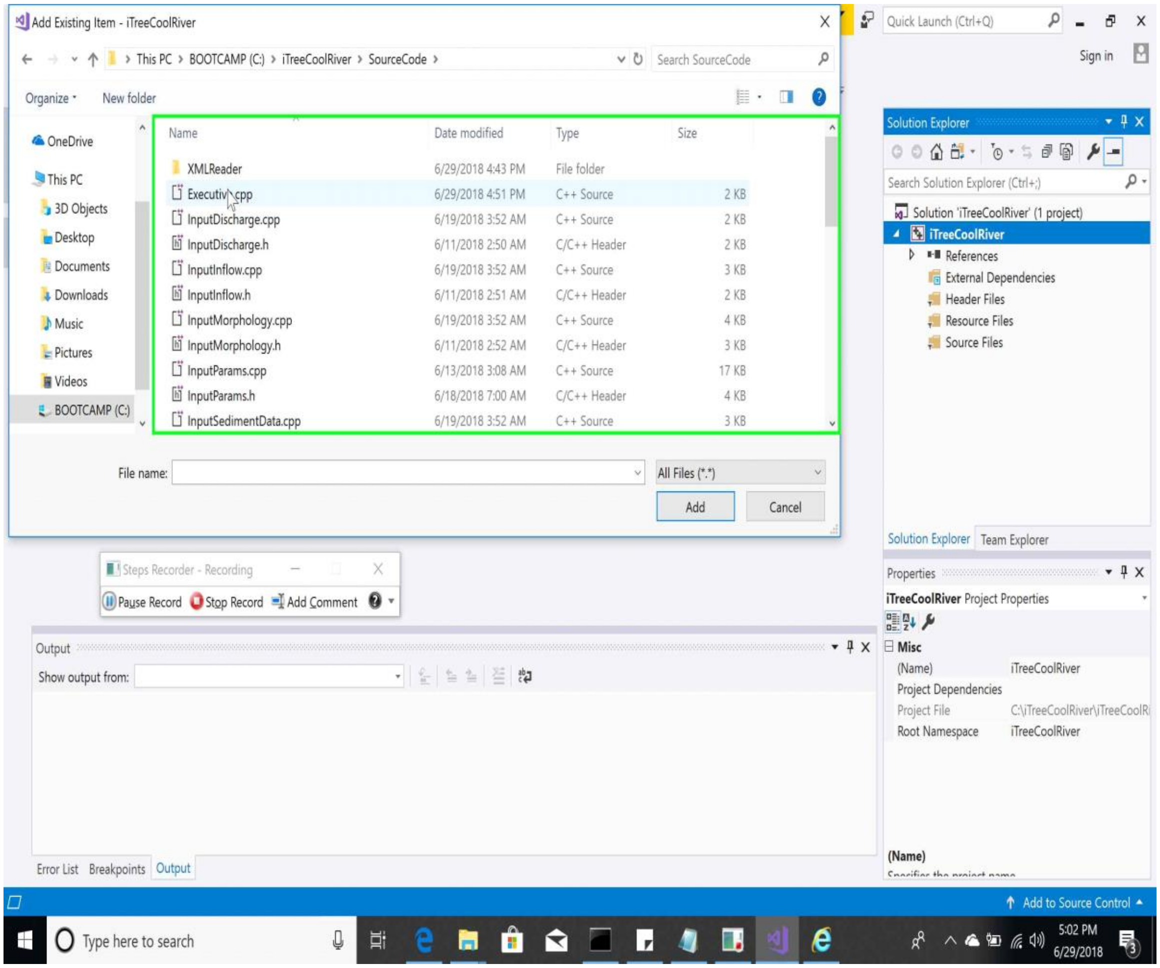
- Create new Visual C++ Empty Project



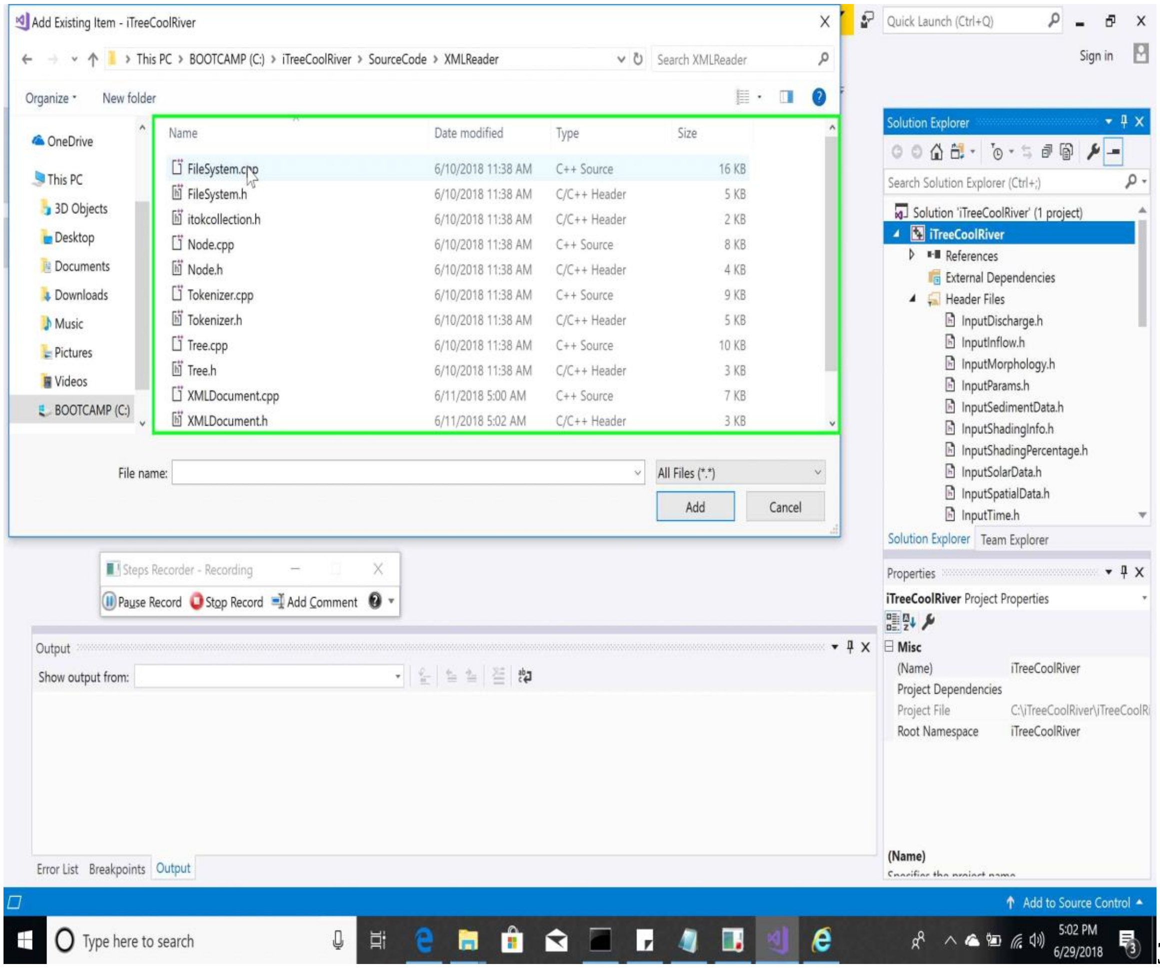
2.

• Add Source Code to Project



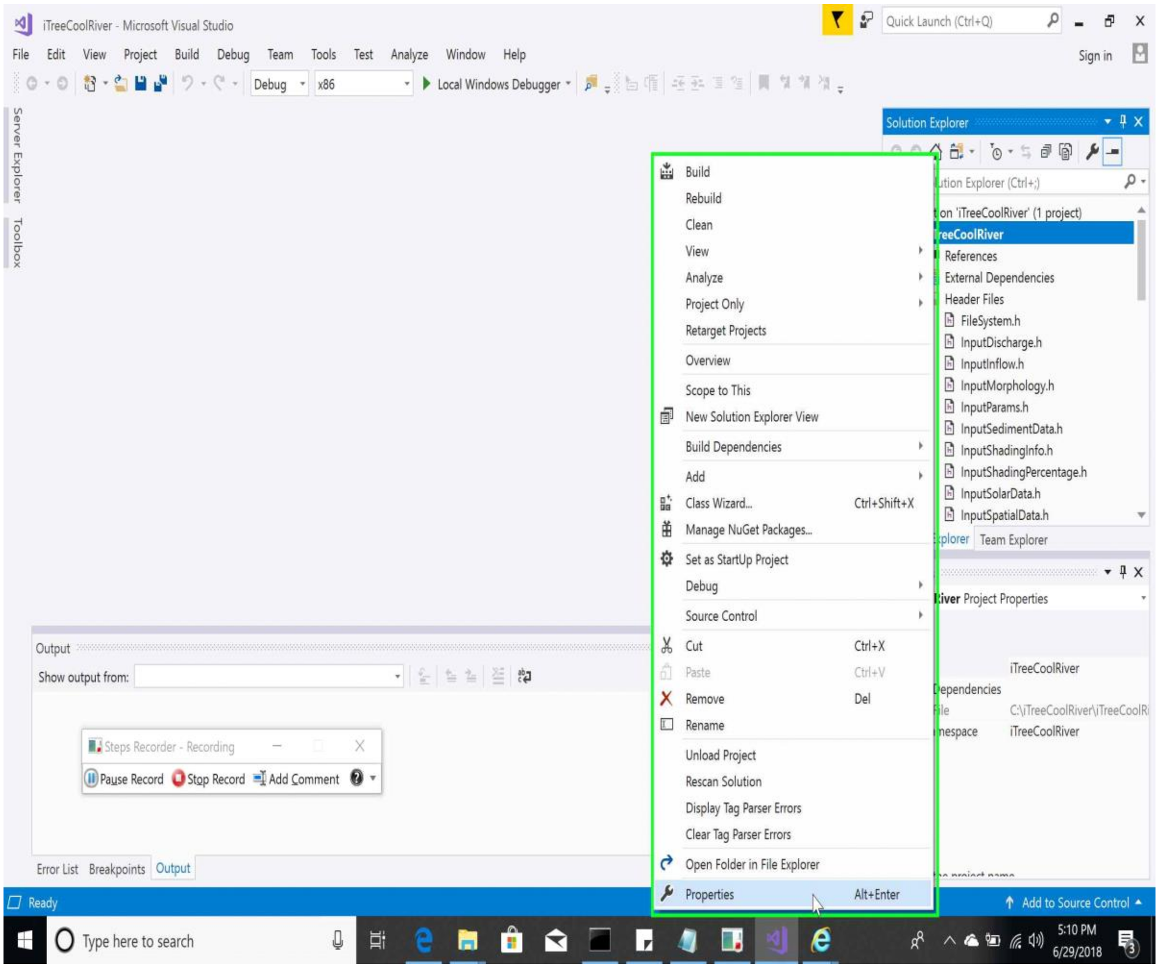

Add Existing Item - iTreeCoolRiver
This PC › BOOTCAMP (C:) › iTreeCoolRiver › SourceCode ›
Search SourceCode
Organize
New folder
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
BOOTCAMP (C:)
Name
Date modified
Type
Size
XMLReader 6/29/2018 4:43 PM File folder
Executive.cpp 6/29/2018 4:51 PM C++ Source 2 KB
InputDischarge.cpp 6/19/2018 3:52 AM C++ Source 2 KB
InputDischarge.h 6/11/2018 2:50 AM C/C++ Header 2 KB
InputInflow.cpp 6/19/2018 3:52 AM C++ Source 3 KB
InputInflow.h 6/11/2018 2:51 AM C/C++ Header 2 KB
InputMorphology.cpp 6/19/2018 3:52 AM C++ Source 4 KB
InputMorphology.h 6/11/2018 2:52 AM C/C++ Header 3 KB
InputParams.cpp 6/13/2018 3:08 AM C++ Source 17 KB
InputParams.h 6/18/2018 7:00 AM C/C++ Header 4 KB
InputSedimentData.cpp 6/19/2018 3:52 AM C++ Source 3 KB
File name:
All Files (*.*)
Add
Cancel
Quick Launch (Ctrl+Q)
Sign in
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'iTreeCoolRiver' (1 project)
iTreeCoolRiver
References
External Dependencies
Header Files
Resource Files
Source Files
Solution Explorer
Team Explorer
Properties
iTreeCoolRiver Project Properties
Misc
(Name) iTreeCoolRiver
Project Dependencies
Project File C:\iTreeCoolRiver\iTreeCoolRi
Root Namespace iTreeCoolRiver
(Name)
Steps Recorder - Recording
Pause Record
Stop Record
Add Comment
Output
Show output from:
Error List
Breakpoints
Output
Add to Source Control
Type here to search
5:02 PM
6/29/2018

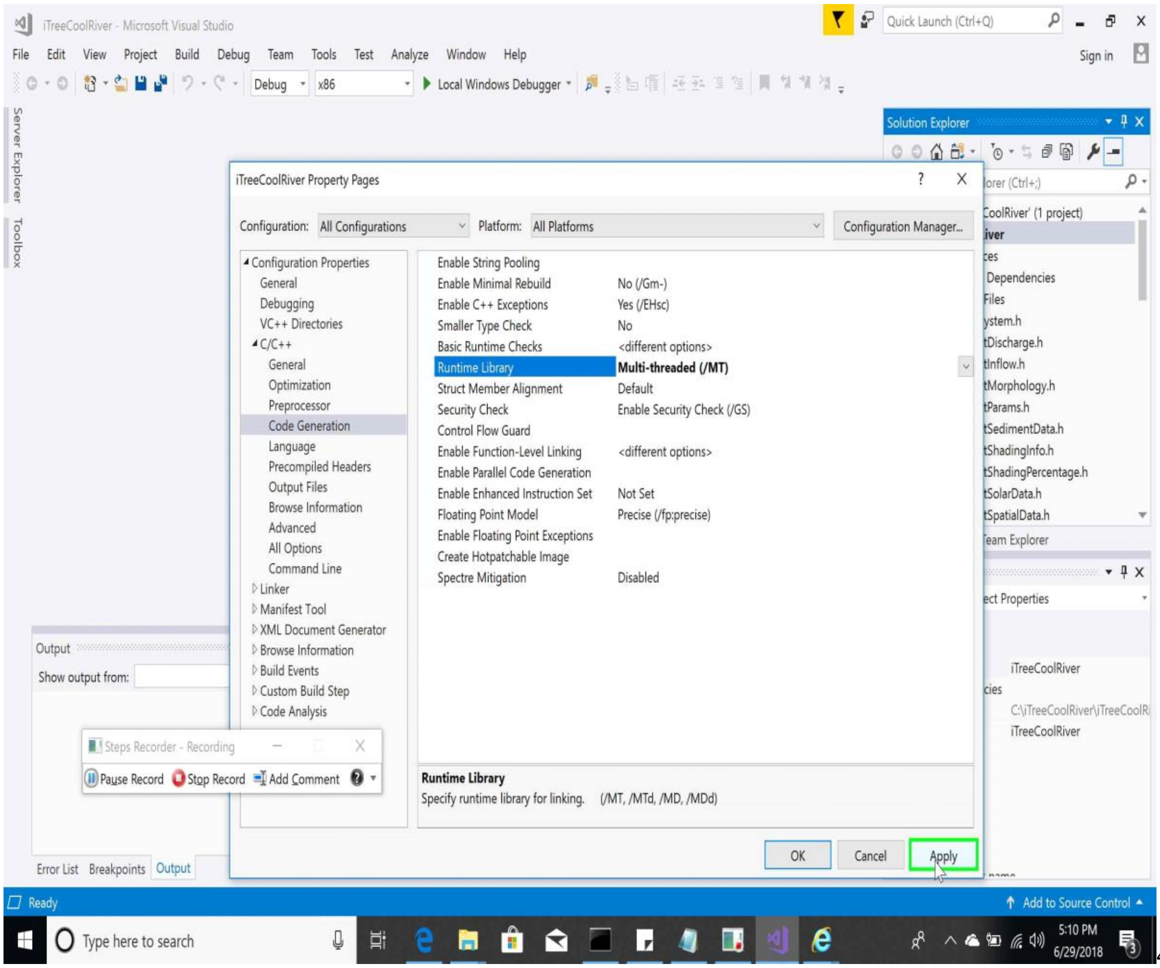
- Don't forget the files in the XmlReader folder.



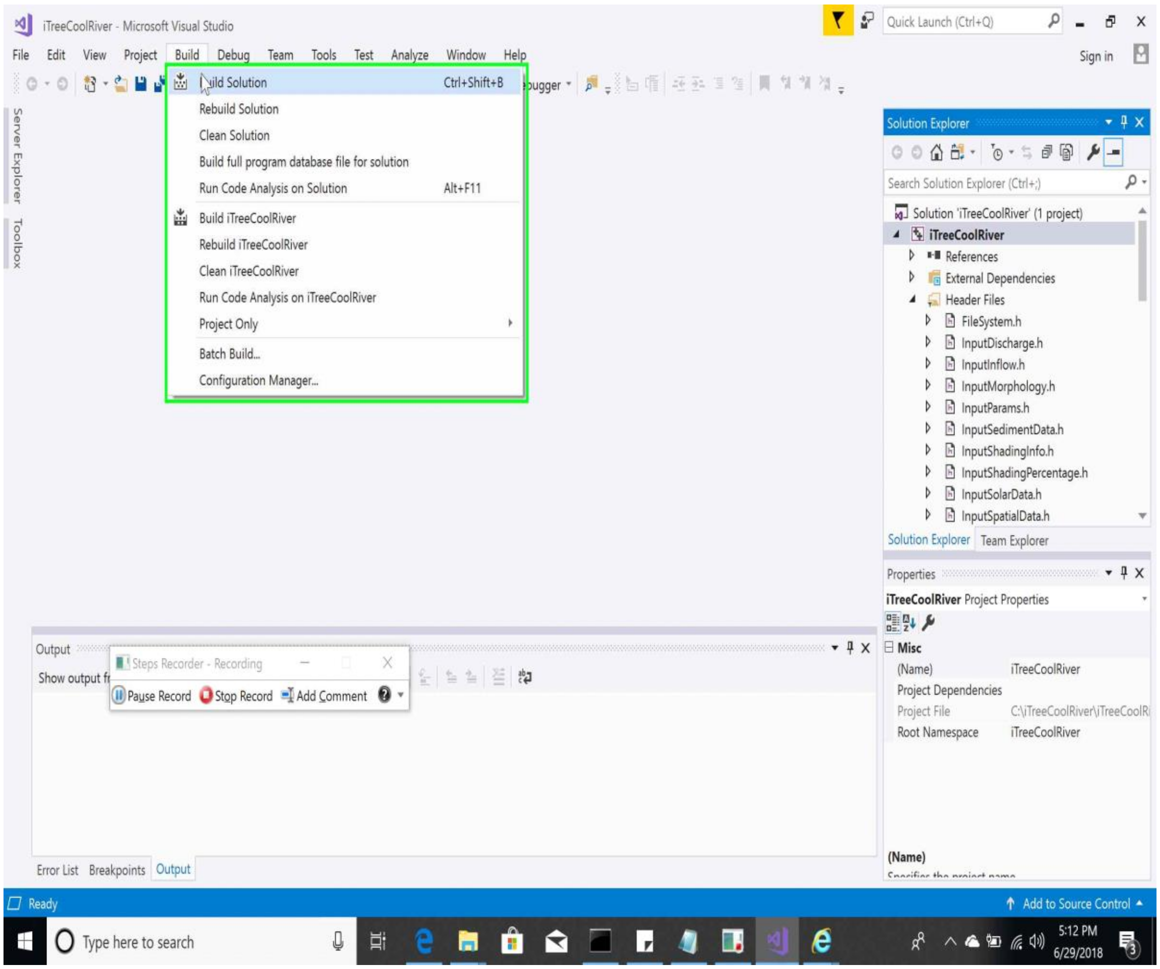
3.

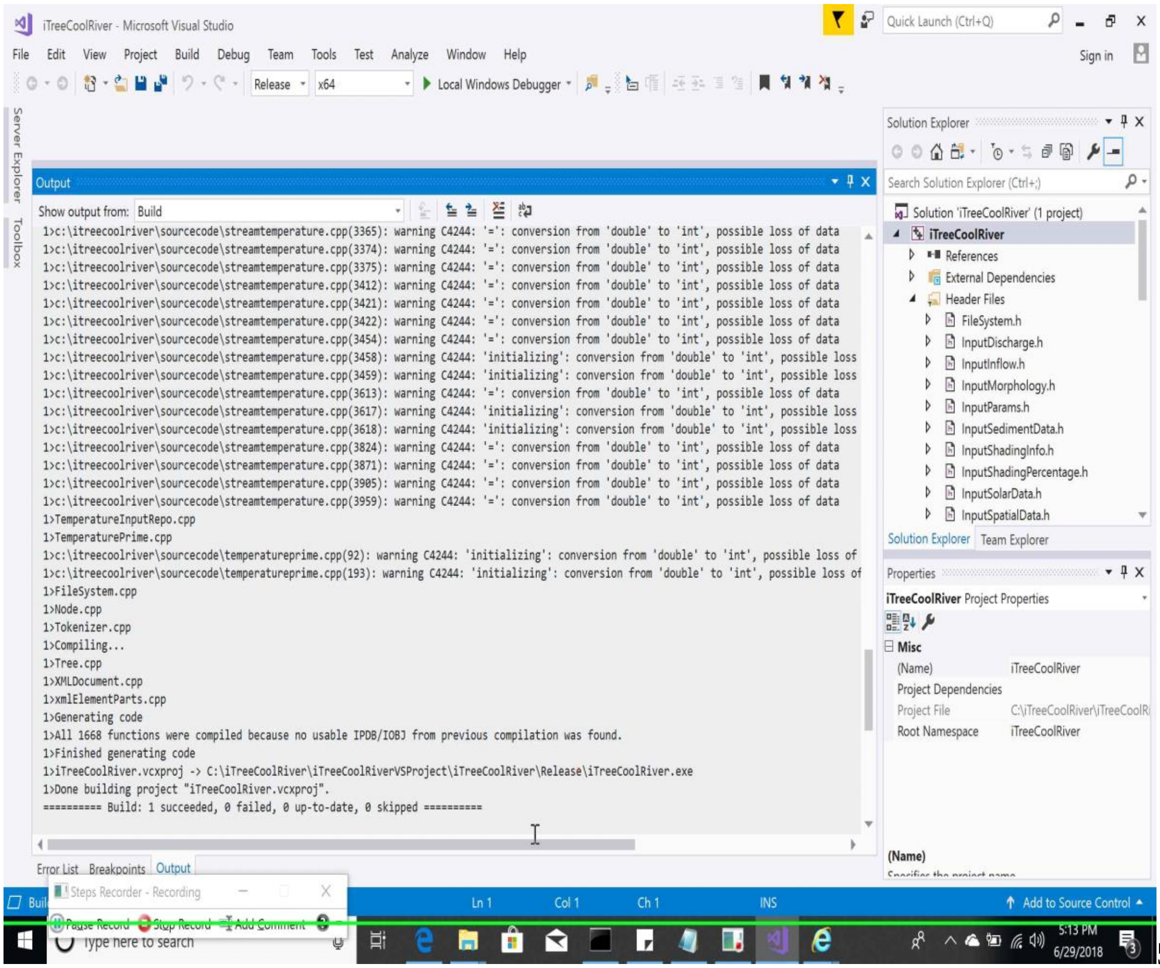
- Change the Code Generation Runtime Library project setting to Multi-threaded (/MT) for All Configuration All Platforms.



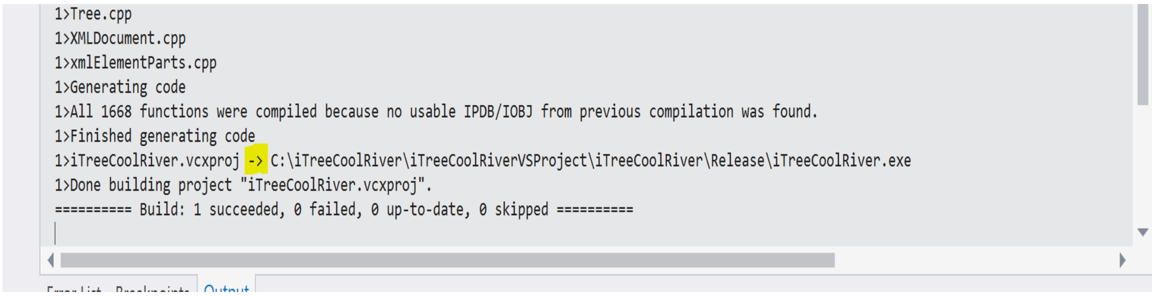

iTreeCoolRiver - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger
Quick Launch (Ctrl+Q)
Sign in
Server Explorer Toolbox
iTreeCoolRiver Property Pages
Configuration: All Configurations Platform: All Platforms Configuration Manager...
Configuration Properties
General
Debugging
VC++ Directories
C/C++
General
Optimization
Preprocessor
Code Generation
Language
Precompiled Headers
Output Files
Browse Information
Advanced
All Options
Command Line
Linker
Manifest Tool
XML Document Generator
Browse Information
Build Events
Custom Build Step
Code Analysis
Enable String Pooling
Enable Minimal Rebuild No (/Gm-)
Enable C++ Exceptions Yes (/EHsc)
Smaller Type Check No
Basic Runtime Checks <different options>
Runtime Library Multi-threaded (/MT)
Struct Member Alignment Default
Security Check Enable Security Check (/GS)
Control Flow Guard
Enable Function-Level Linking <different options>
Enable Parallel Code Generation
Enable Enhanced Instruction Set Not Set
Floating Point Model Precise (/fp:precise)
Enable Floating Point Exceptions
Create Hotpatchable Image
Spectre Mitigation Disabled
Runtime Library
Specify runtime library for linking. (/MT, /MTd, /MD, /MDd)
OK Cancel Apply
Solution Explorer
Output
Show output from:
Steps Recorder - Recording
Pause Record Stop Record Add Comment
Error List Breakpoints Output
Ready
Add to Source Control
Type here to search
5:10 PM
6/29/2018
4.

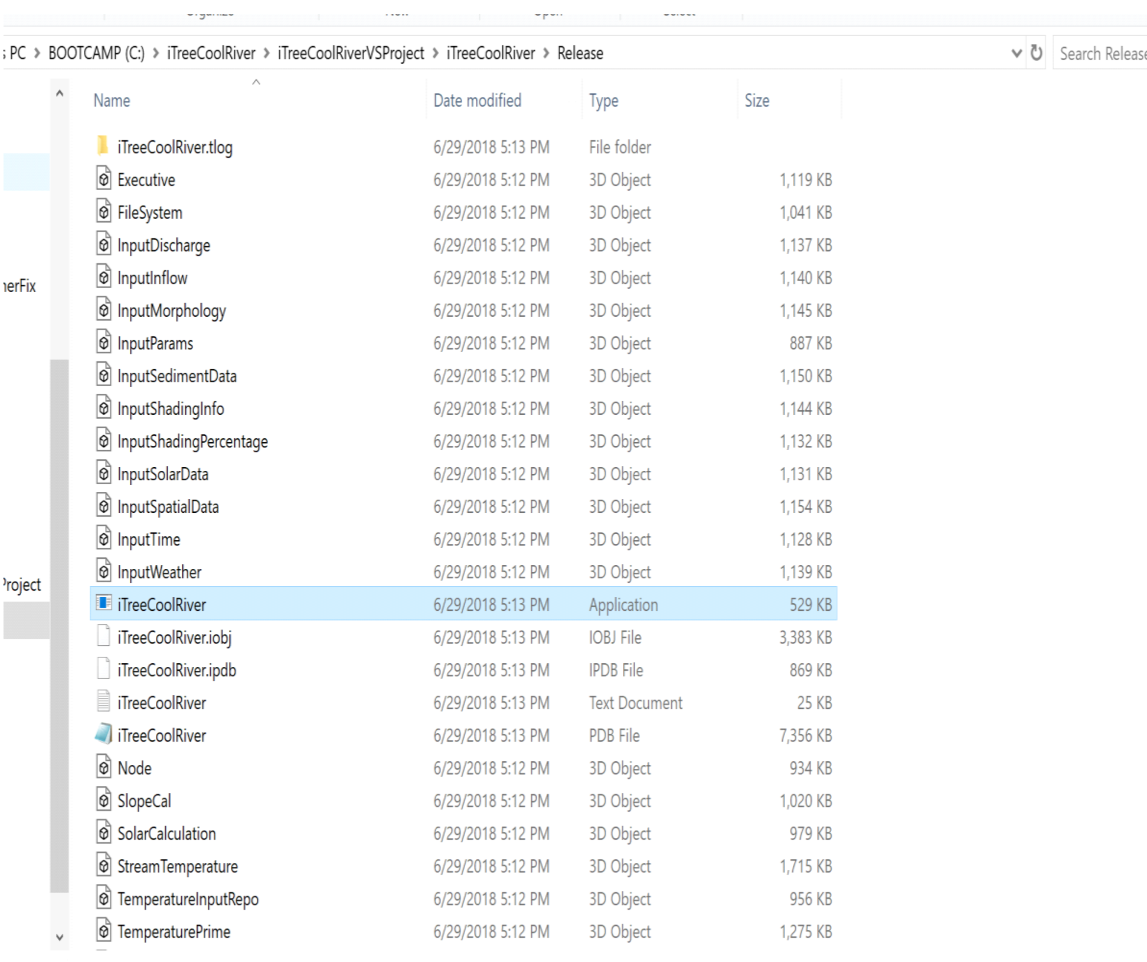
- Build the project




iTreeCoolRiver - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Sign in
Release x64 Local Windows Debugger
Server Explorer Toolbox
Output
Show output from: Build
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3365): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3374): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3375): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3412): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3421): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3422): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3454): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3458): warning C4244: 'initializing': conversion from 'double' to 'int', possible loss
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3459): warning C4244: 'initializing': conversion from 'double' to 'int', possible loss
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3613): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3617): warning C4244: 'initializing': conversion from 'double' to 'int', possible loss
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3618): warning C4244: 'initializing': conversion from 'double' to 'int', possible loss
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3824): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3871): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3905): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>c:\itreecoolriver\sourcecode\streamtemperature.cpp(3959): warning C4244: '=': conversion from 'double' to 'int', possible loss of data
1>TemperatureInputRepo.cpp
1>TemperaturePrime.cpp
1>c:\itreecoolriver\sourcecode\temperatureprime.cpp(92): warning C4244: 'initializing': conversion from 'double' to 'int', possible loss of
1>c:\itreecoolriver\sourcecode\temperatureprime.cpp(193): warning C4244: 'initializing': conversion from 'double' to 'int', possible loss of
1>FileSystem.cpp
1>Node.cpp
1>Tokenizer.cpp
1>Compiling...
1>Tree.cpp
1>XMLDocument.cpp
1>xmlElementParts.cpp
1>Generating code
1>All 1668 functions were compiled because no usable IPDB/IOBJ from previous compilation was found.
1>Finished generating code
1>iTreeCoolRiver.vcxproj -> C:\iTreeCoolRiver\iTreeCoolRiverVSProject\iTreeCoolRiver\Release\iTreeCoolRiver.exe
1>Done building project "iTreeCoolRiver.vcxproj".
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
Error List Breakpoints Output
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'iTreeCoolRiver' (1 project)
iTreeCoolRiver
References
External Dependencies
Header Files
FileSystem.h
InputDischarge.h
InputInflow.h
InputMorphology.h
InputParams.h
InputSedimentData.h
InputShadingInfo.h
InputShadingPercentage.h
InputSolarData.h
InputSpatialData.h
Solution Explorer Team Explorer
Properties
iTreeCoolRiver Project Properties
Misc
(Name) iTreeCoolRiver
Project Dependencies
Project File C:\iTreeCoolRiver\iTreeCoolRi
Root Namespace iTreeCoolRiver
(Name)
Steps Recorder - Recording
Pause Record Stop Record Add Comment
Ln 1 Col 1 Ch 1 INS
Add to Source Control
Type here to search
5:13 PM
6/29/2018
5.

- Run your build.





- Double click on application file.
- Enter path of config file (i-TreeCoolRiver.xml) and hit enter.