HydroModeler Documentation

Beta Release

Anthony Castronova, Shirani Fuller, and Jonathan Goodall

University of South Carolina

July 19, 2010

Table of Contents

Introduction

HydroModeler is a HydroDesktop plugin that extends its capabilities of HydroDesktop to include component-based modeling. HydroModeler is built on an open-source model coupling environment provided by the OpenMI Association Technical Committee (OATC). Therefore, the approach for developing model components is identical to that outlined in the OpenMI document series (http://www.openmi.org). HydroModeler's integration within the HydroDesktop environment provides the ability to retrieve, visualize, and edit data from remote sources. HydroModeler adds the ability to build and execute a model within HydroDesktop using a component-based (i.e., plug-and-play) model development strategy. Basic functionality included in the HydroModeler plug-in includes the ability design, save, and run model compositions, and the ability to input and output time series between HydroDesktop's internal database and model components. Additional functionality will be added as the plug-in is further developed over the coming months.

We do our best to keep this documentation up to data and as complete as possible. However, there are many details of HydroModeler that are not covered in this documentation. If you have additional questions after reading this documentation, please feel free to email Jon Goodall at goodall@sc.edu.

Scope of this Document

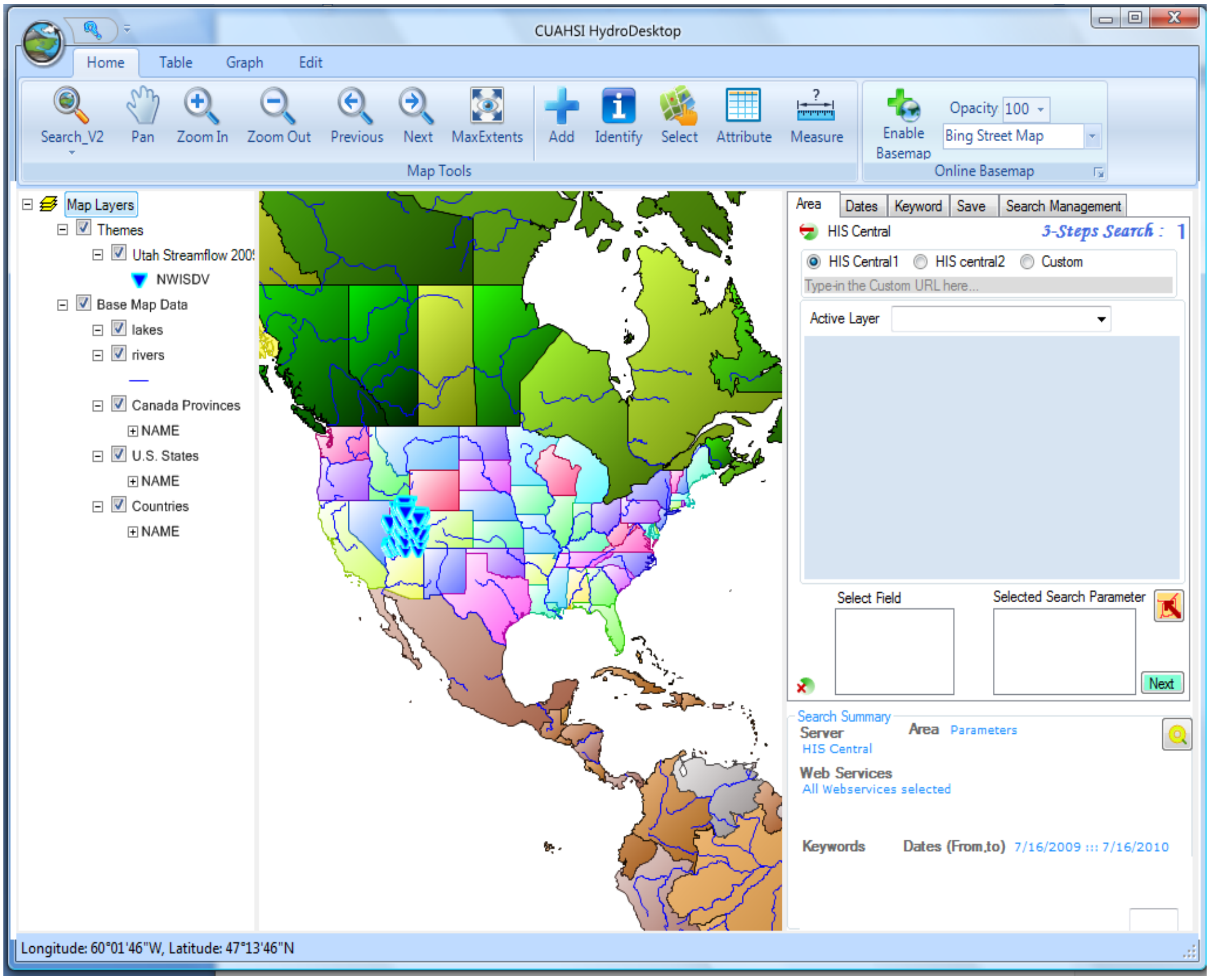
This document is divided into three main sections. Part 1 explains the basic concepts of HydroModeler by walking you through a sample model configuration. Part 2 explains how to take an existing model configuration and apply it to a specific location by altering model input files. Finally, Part 3 provides a description of how to create a model component from scratch by walking through the creation of a sample model component. All source codes are available for viewing and downloading from our repository (http://hydrodesktop.codeplex.com/SourceControl/list/changesets).

Part 1 - Getting started with the HydroModeler Plugin

1. Starting HydroDesktop

1. Download the latest version of HydroDesktop (available at http://www.hydrodesktop.com/), and follow the installation wizard.

HydroDesktop is based on an open source GIS software system called MapWindow, so the primary interface in the software is the map found on the "Home" tab. By default, the map contains simple base data (countries of the world, U.S states, major rivers and lakes, watersheds). HydroDesktop also contains an array of plugins which are visible as individual ribbon tabs. This is illustrated by the Table and Graph plugins shown in the figure below. One of the primary purposes of HydroDesktop is to act as a client application for the CUAHSI Hydrologic Information System web data services. This functionality is described in other documentation. Here we focus on the HydroModeler extension functionality.

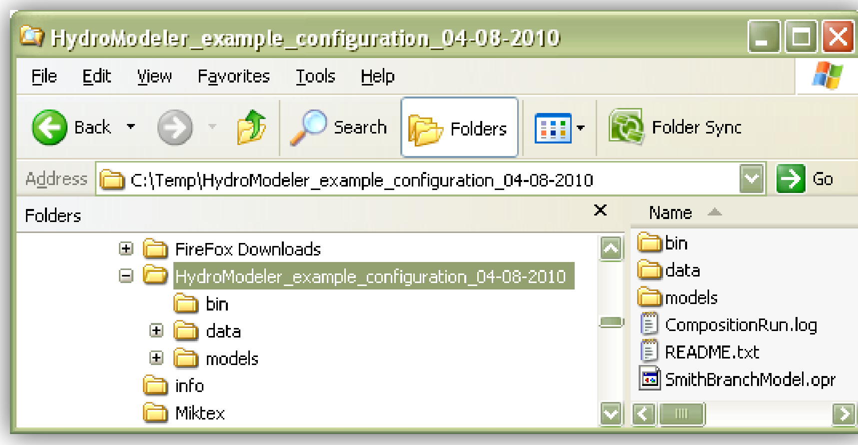


2. Open an Existing Model

A simple model configuration can be created and executed using freely available model components, provided by the HydroDesktop community. This section describes how to utilize pre-developed model components to recreate a model simulation.

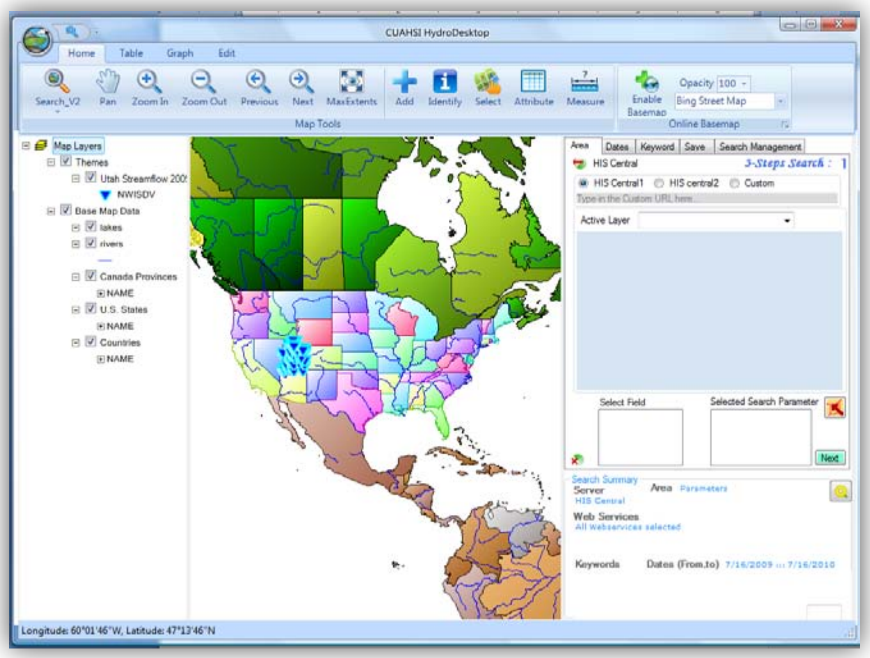
2. Navigate to the Documentation tab on the HydroDesktop website (http://www.hydrodesktop.com). Here you will find tutorials on various features of HydroDesktop, including “HydroModeler Example Configuration”. Download the latest version.

3. Extract this folder into c:/Temp, so that the HydroModeler folder is located at *C:/Temp/HydroModeler example configuration…*

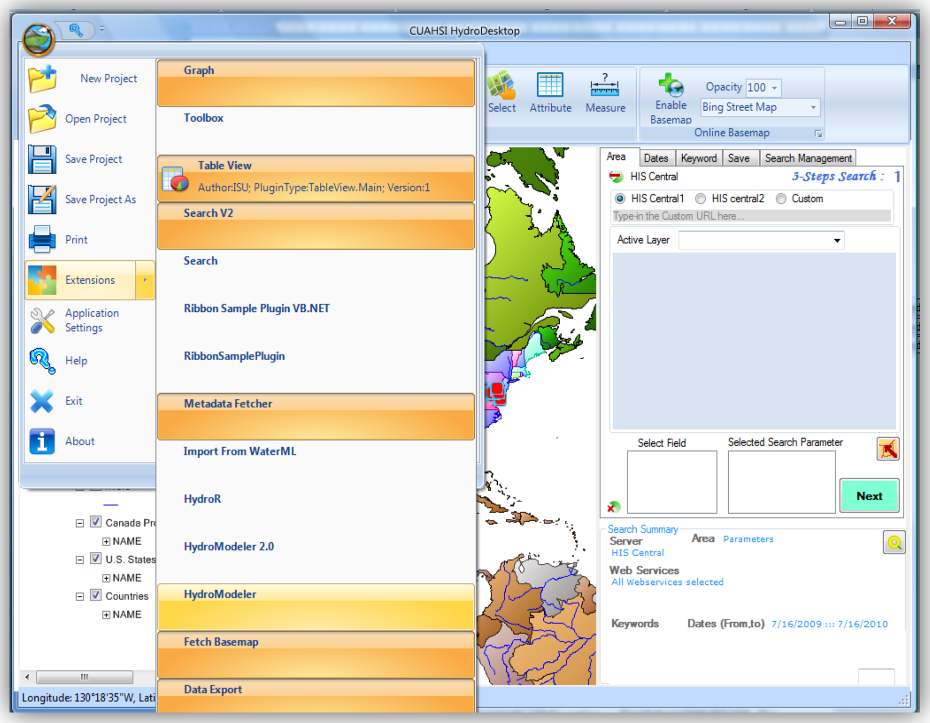


In the bin directory you will find all of the libraries needed to run the model. The "data" folder contains all of the necessary input files for each model component, and the "models" folder contains files pertaining to each individual model component.

4. Open HydroDesktop (*Start → Program Files → HydroDesktop)*

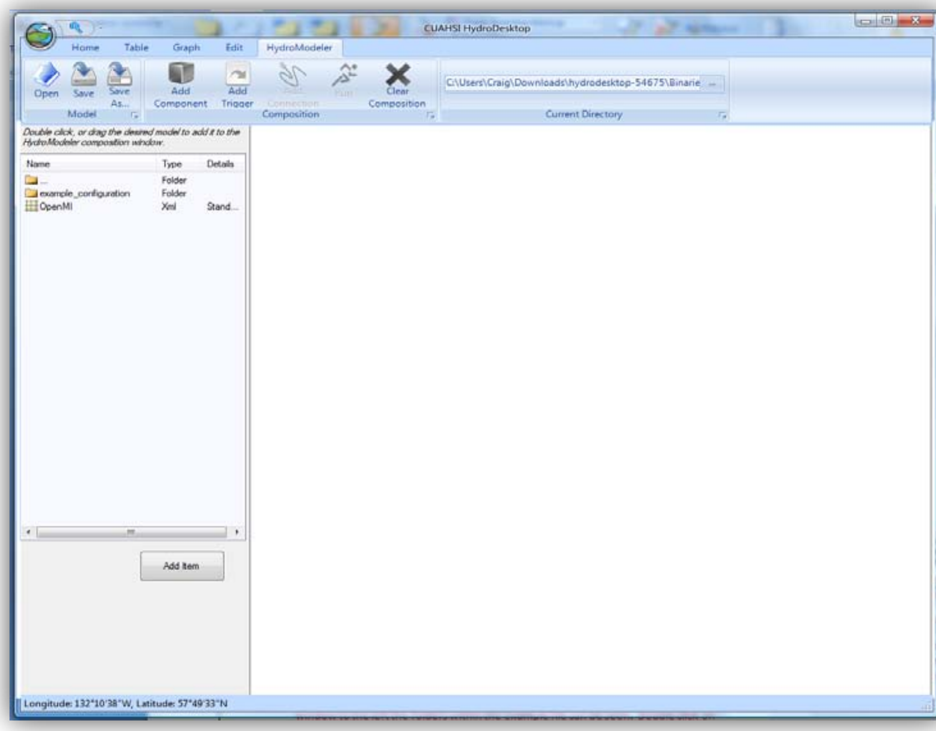


5. Load the HydroModeler plugin by selecting the orb in the upper left corner of the screen. A drop down menu will appear, select →*Extensions* → *HydroModeler*.



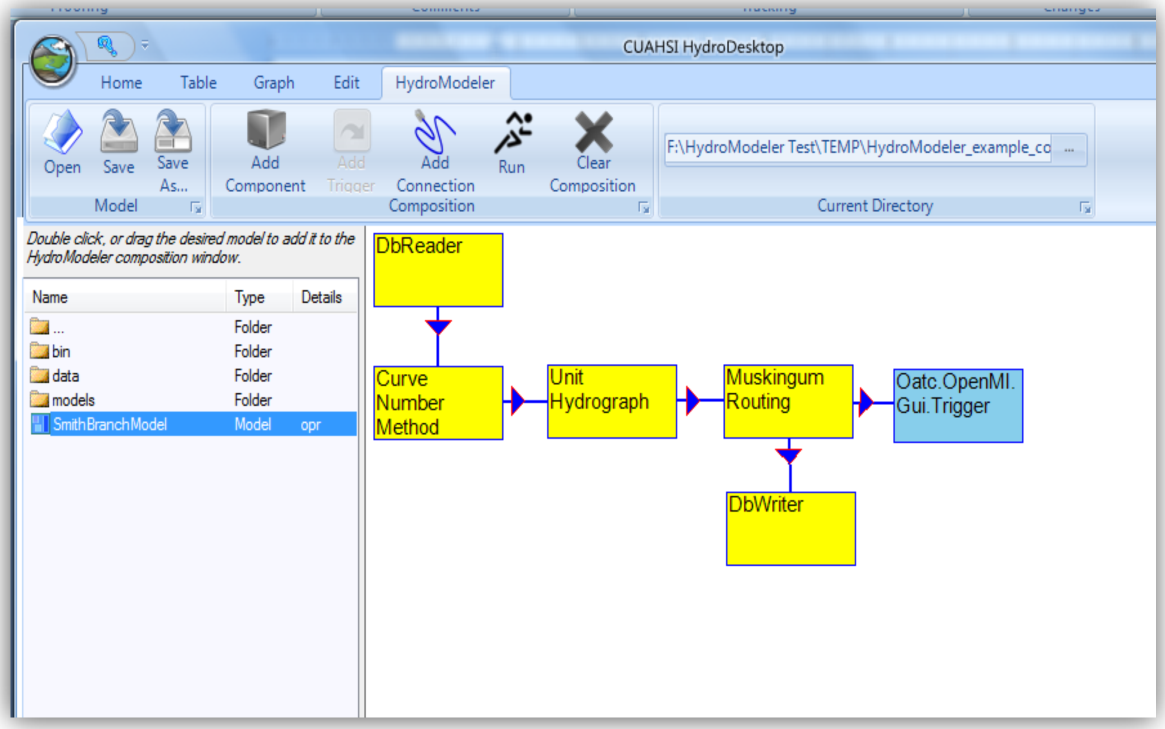
Note: Your list of available extensions may differ from this list.

6. The HydroModeler tab will appear:



3. Running an Existing Model

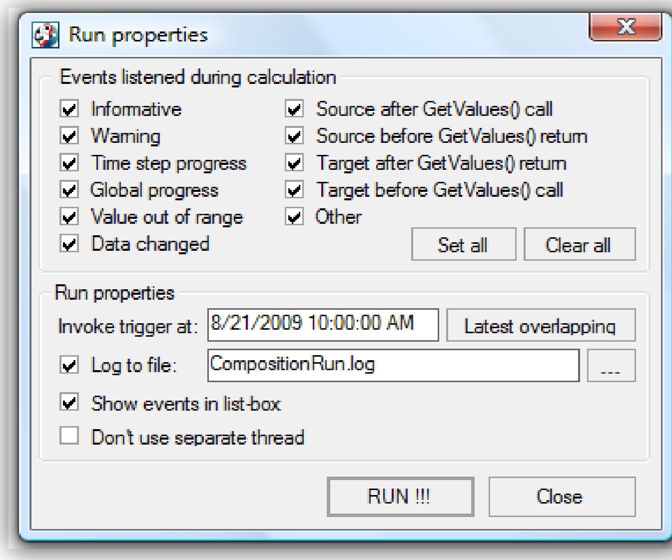
7. At the top of the page there is an area labeled *Current Directory*. Contained within is a textbox that when changed will update the file browser shown on the left panel of the screen. Alternatively, the toggle button located next to the navigation textbox can be used to browse for a specific directory. Navigate to the location of the HydroModeler example that was downloaded in step 2 (*c:/temp/HydroModeler_example_configuration_ …*). In the window to the left you should see the folders within the HydroModeler example. Double click on SmithBranchModel.opr. The HydroModeler window should now look like this:



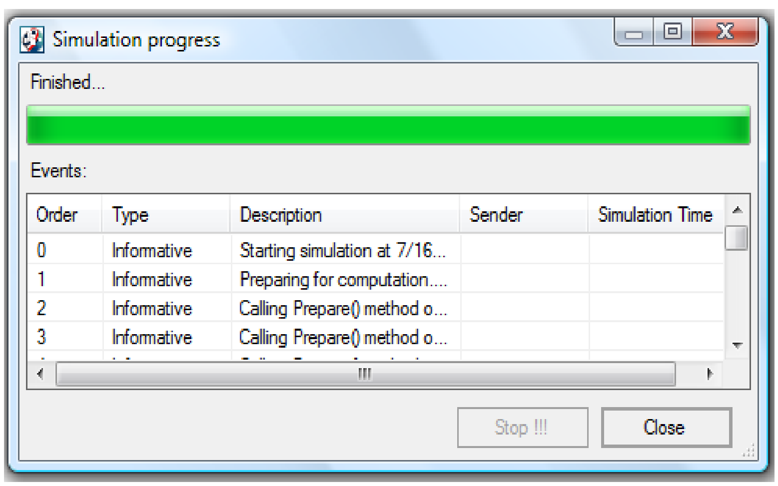
Right clicking on the arrows provides information about the data being passed across each link. Everything has been pre-defined, so there is no need to edit any of the links.

8. Execute the model by selecting the"Run" button from the top of the page.

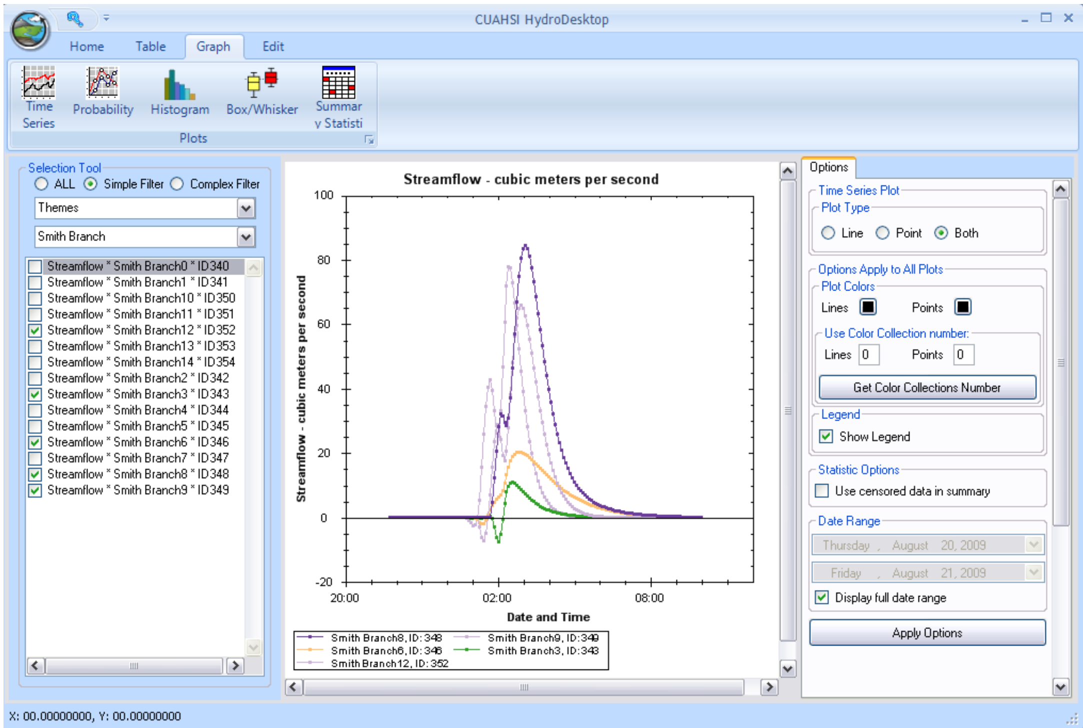
9. The Run Properties window will appear. This is where simulation properties are specified, such as notifications, log file path, etc... Everything has been pre-set so all that is needed is to click **RUN !!!**



10. During model simulation, a progress window will show you events that are occurring. When simulation is complete, the last notification will read "Simulation Finished Successfully". Click "Close". At the next dialog window, click "Yes" to reload the model components.



11. To view the model's output, select the Graph View tab in HydroDesktop. The *Simple Filter* allows us to summarize the data within the database by theme and criteria. Choose *Themes* under *criterion* and *Smith Branch* under *filter option*.



You have successfully executed your first HydroModeler simulation. The next section will describe how to generate the input files for this model to simulate a watershed other than the default watershed.

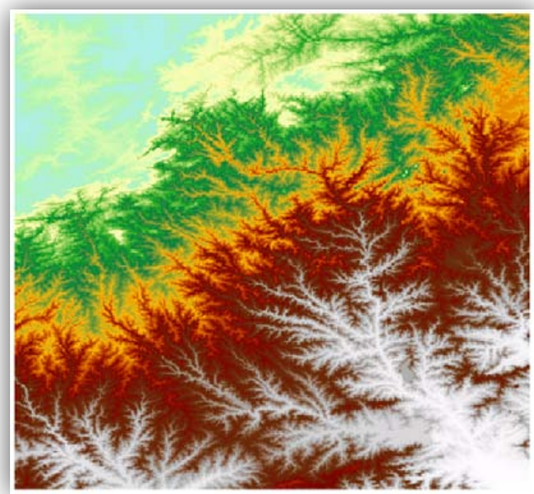
Part 2 - Adjusting Model Input

This section walks you through the entire model development process, from creating input files to linking model components. All of the necessary data can be retrieved from online sources and manipulated using standard ArcInfo tools. The necessary data for this section is listed below.

Name	Source	Description
DEM	http://seamless.usgs.gov/	Digital Elevation Model
NHD+	http://www.horizon-systems.com/nhdplus/	National Hydrography Dataset
Soils	http://soils.usda.gov/survey/geography/ssurgo/	Soil Survey Geographic Database
LandCover	http://seamless.usgs.gov/	Land Use and Land Cover 2001

1. Defining a New Watershed

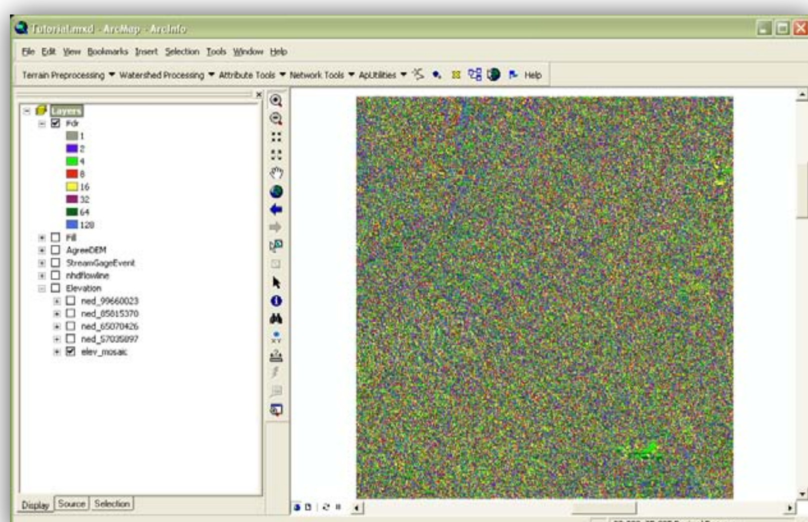
1. Open ArcMap and add the elevation DEM. The map below shows a typical elevation map.



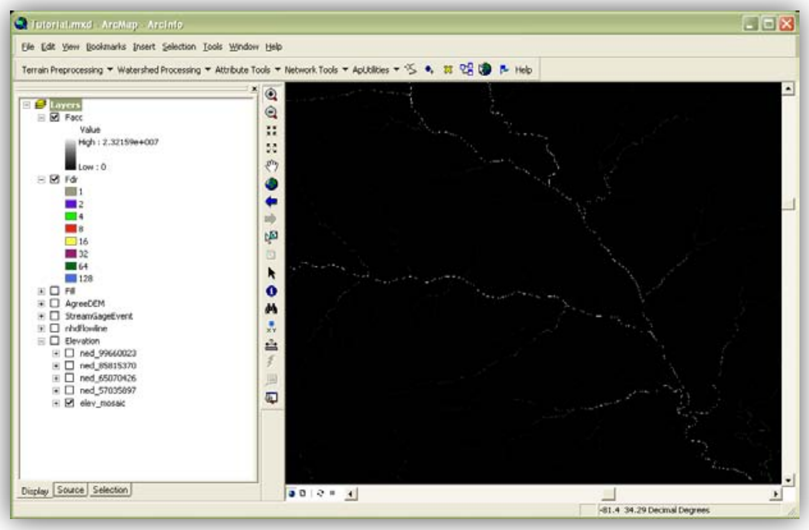
2. Convert this **DEM** into a projected coordinate system rather than its default geographic coordinate system. This is done using the *Project Raster* tool which can be found within the ArcToolbox.

3. Next, all "sinks" within the **DEM** must be filled to ensure that water won't accumulate at any cell within the elevation grid. Use the *Fill* tool within the ArcToolbox. The output raster will only have minor changes. Name this file **Fill**

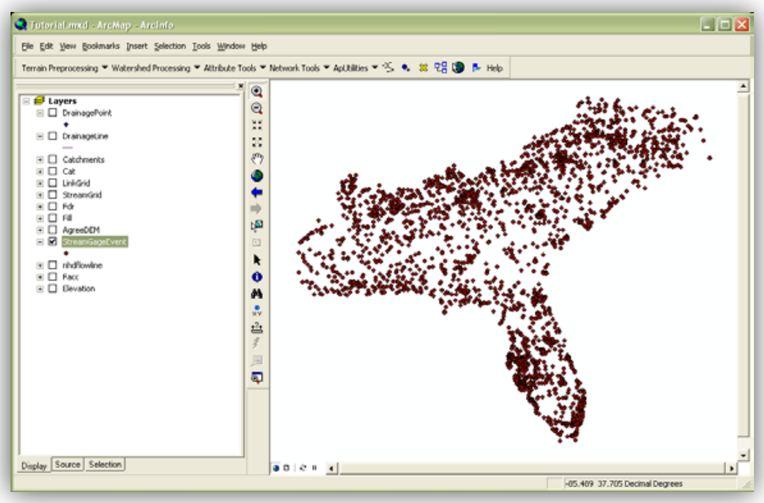
4. From the ArcToolbox, select *Spatial Analyst Tools → Hydrology → Flow Direction.* This tool determines the direction water will flow from each raster cell. Supply **Fill** as the "Input surface raster", and save the resulting flow direction raster as **Fdr.** The output will look similar to map below, where values range from 1 to 128.



5. From the ArcToolbox, use the *Spatial Analyst Tools* → *Hydrology* → *Flow Accumulation* tool to calculate how many contributing pixels each element in the raster has. This is useful in determining the where water will accumulate after a rain event. Use **Fdr,** created in the last step, as this "Input flow direction raster", and save the output accumulation raster as **Facc**.

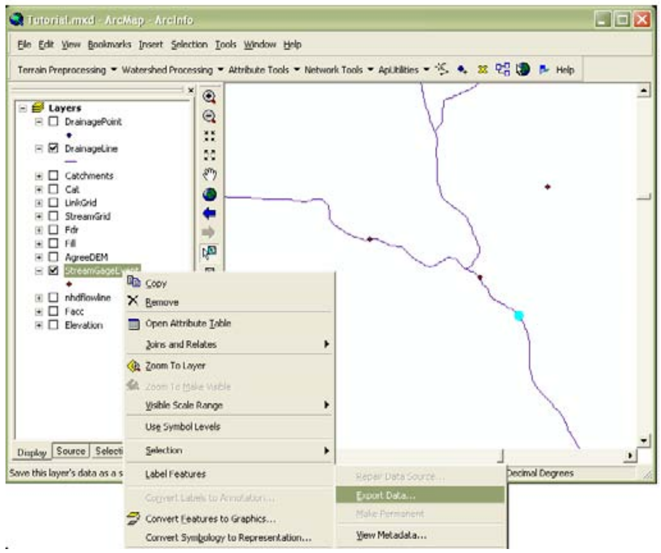


6. Next, we can use the USGS gage stations supplied within the **NHD+** data to select an appropriate outlet point for our watershed. Load **StreamGageEvent.shp** into the project (this can be found within the NHD+ download).

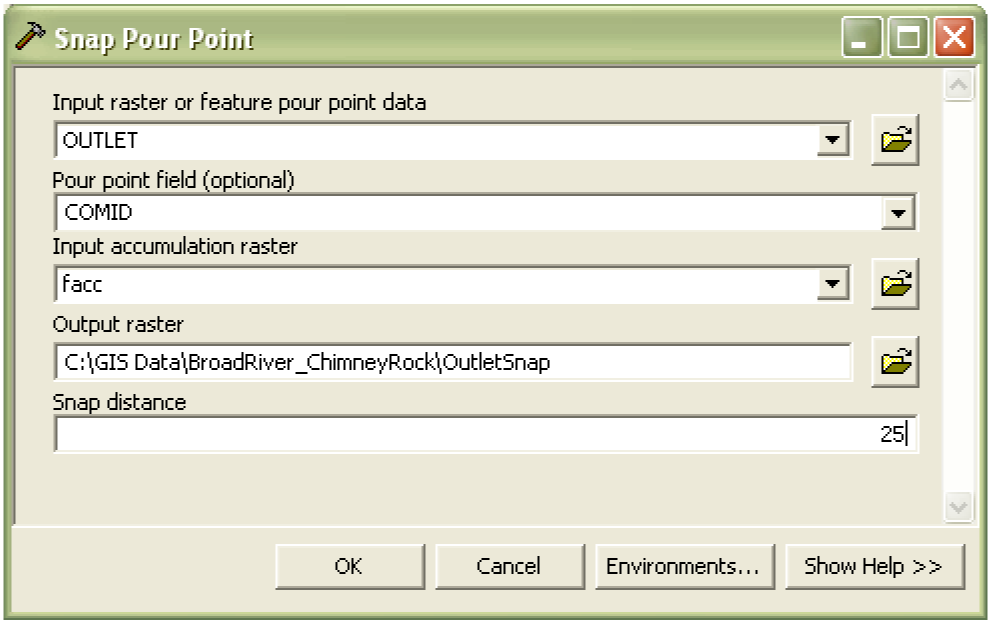


7. Once you have determined a suitable outlet point, select the point using the "Select Features Tool" . Next, right click on **StreamEventLayer.shp** and select *Data* → *Export*

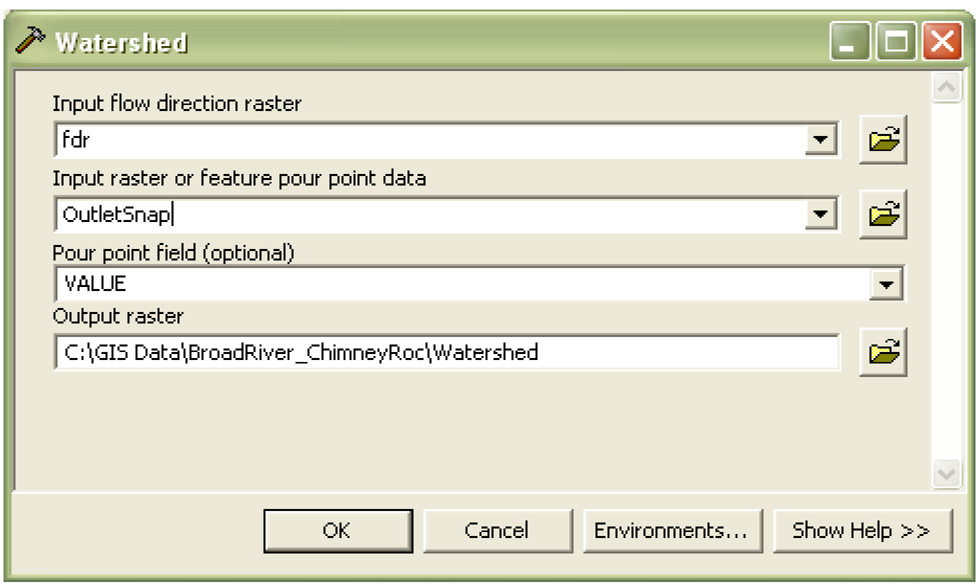
Data. Be sure to export only the selected features. Name this point **Outlet.shp**. After creating outlet.shp converted it to a projected coordinate system rather than its default geographic coordinate system

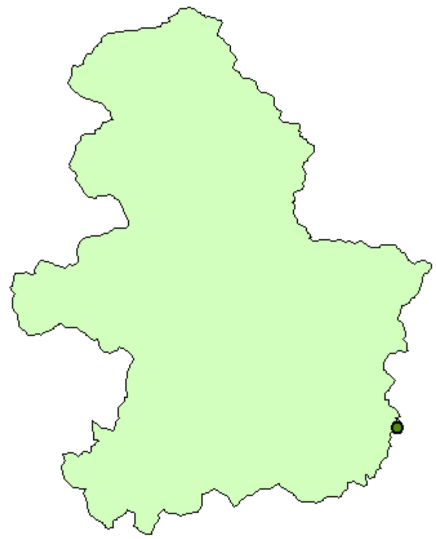


8. Within the ArcToolbox, use *Spatial Analyst Tools* → *Hydrology* → *Snap Pour Point* to make sure that this outlet point is positioned on a location of high accumulation. Specify **Outlet.shp** as the “Input feature pour point data”, COMID as the “pour point field” and **Facc** as the “Input accumulation raster”. Be sure to select as reasonable “Snap Distance” (maximum distance the point will be allowed to move from its original position). To determine a good snap distance use the ruler tool and measure from the outlet to the flowline. Save the output as **OutletSnap**. Once the outlet snap is created verify that the snap is on the flow path of the Facc and is near the desired outlet point.



9. Define the watershed boundary using *Spatial Analyst Tools → Hydrology → Watershed.* Supply **Fdr** and **OutletSnap** as inputs, save the output as **Watershed.**

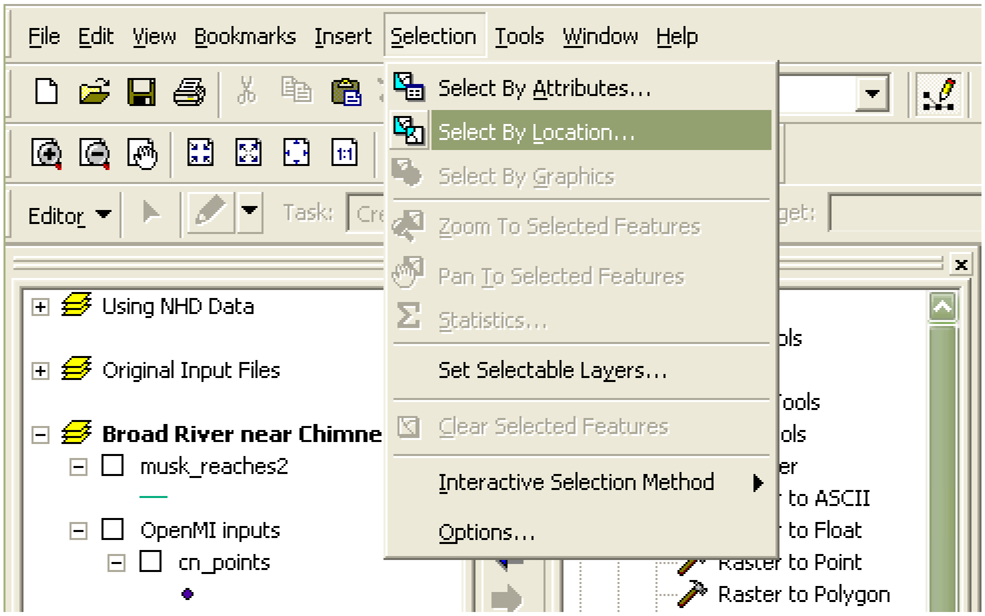




10. From the ArcToolbox use *Conversion Tools → From Raster → Raster To Polygon t*o conve**rt** the w**atershed r**aster into a feature class. Na**me the output f**ile **Watershed.shp**.

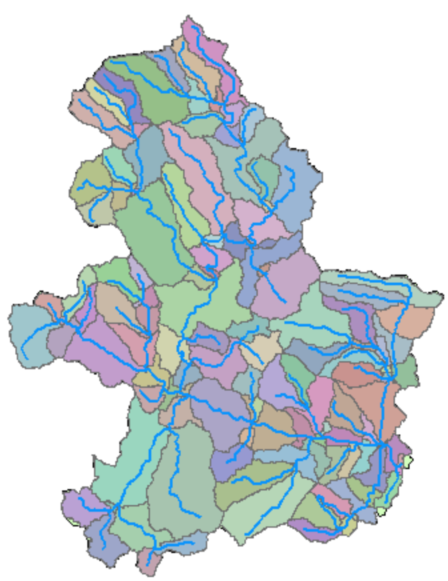
11. Next, the subbasin features must be delineated. There are several methods for doing this, but for simplicity we'll use the catchments that have already been delineated within the NHD Plus data set. Load the **catchments.shp** file located within the NHD+ download. Reminder: The catchment file is large; stay zoomed into the area of interest and for faster loading.

12. Select by location, all features from **catchments.shp** that intersect with **Watershed.shp**. Right click **on catchments.shp** and select *Data→Export Data* (export selected features). Save this output as **Subbasins.shp**.

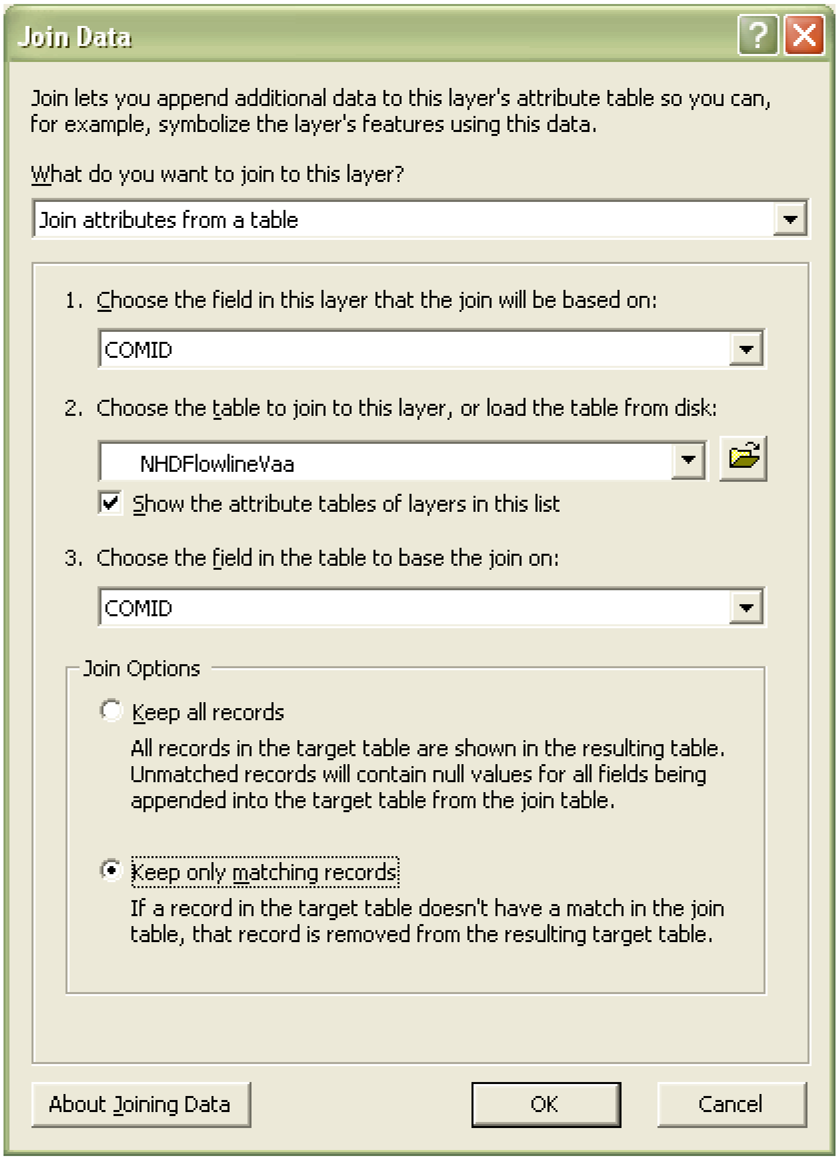


13. From the NHD+ dataset, load **NhdFlowline.shp**.

14. Join **Subbasins.shp** with **NhdFlowline.shp** based on ComID (keep only matching records). This will select only the streams corresponding to the desired subbasins. Right click on **NhdFlowline.shp** and select *Data* → *Export Data* (only selected features). Save this as **Reaches.shp**. NOTE: Make sure that there are an equal number of subbasins and stream elements in **Subbasins.shp** and **Reaches.shp**, respectively.



15. Add the **NHDFlowlineVaa.dbf** (part of the NHD+ dataset) table to the project. This table contains information about how the reaches are related. Right click on **Reaches.shp** and select *Joins and Relates →Join.* Specify COMID as the field to be joined on and **NHDFlowlineVaa.dbf** as the table to join. Make sure to select "*keep only matching records".*

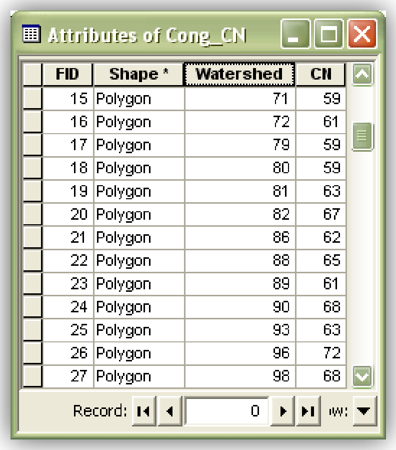


16. Add **FlowlineAttributesFlow.dbf** (part of the NHD+ dataset) table to the project. This table gives us some flow characteristics of each reach. As was done in the previous step, *Join* this table, based on COMID, to **Reaches.shp**. Make sure to select "*keep only matching records".*

2. Creating Component Inputs

The SCS Curve Number Component

This component requires one shapefile with the following attribute table. The shapefile can contain polygon or point features, each of which represents a subbasin within the element set. Each row must contain a unique Watershed ID, and a weighted Curve Number.

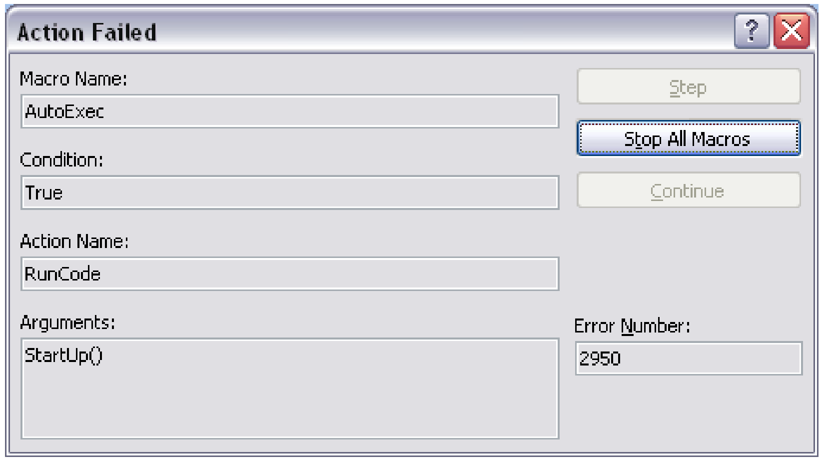


FID	Shape *	Watershed	CN
15	Polygon	71	59
16	Polygon	72	61
17	Polygon	79	59
18	Polygon	80	59
19	Polygon	81	63
20	Polygon	82	67
21	Polygon	86	62
22	Polygon	88	65
23	Polygon	89	61
24	Polygon	90	68
25	Polygon	93	63
26	Polygon	96	72
27	Polygon	98	68

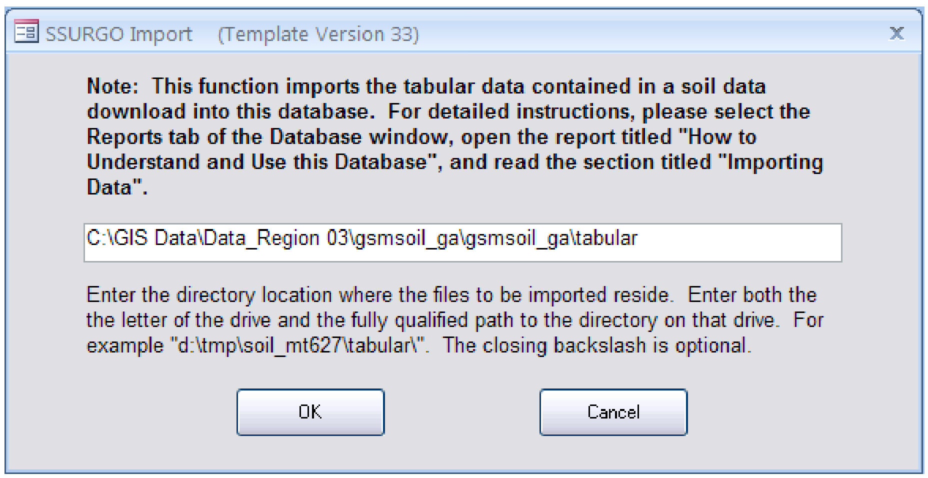
The following steps explain how to create this input file.

1. Load **SoilData.shp** and **LandCoverData.tif** into the project. The soil map can be found under the spatial directory of the Soils download, and the land cover data is specified within its download directory.

2. Next, we need to prepare the muggatt.dbf data before we add it to our project.
   a. Extract the zipped file for the soil data that you downloaded from the Soil Survey Geographic Database and you should a file called soildb_US_2002.zip.
   b. Extract this file. You should now have soildb_US_2002.mdb.
   c. Open soildb_US_2002.mdb with Microsoft Access.
   d. If dialog box pops up asking you to "Stop All Macros", click it.
      Then you should see a bar below the Microsoft ribbon toolbar that says "Security Warning ...". Click on "Options..." and select "Enable this content" then click OK.

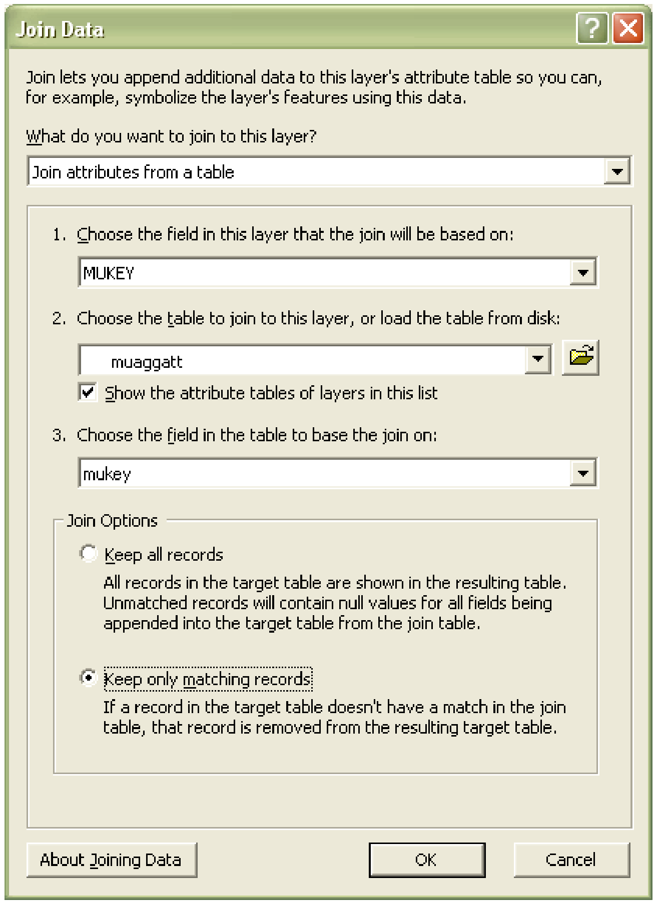


e. Enter the path where the tabular files reside, for example c:/temp/gsmsoil_sc/tabular. This will put the data from the database into the shapefiles.



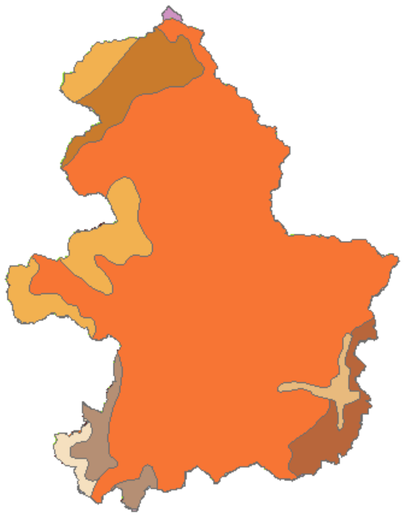
3. Once you click “OK”, Microsoft Access will begin populating the database. The next dialog box will ask you if you would like to generate a soil reports, click “Exit”.

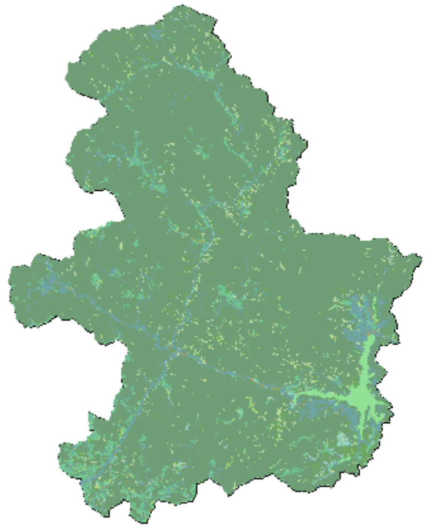
4. Back in ArcMap, *Join* muaggatt.dbf to **Soil.shp**, based on MUKEY. Muaggatt.dbf can be found in Soildb_US_2002.mdb, included in the download. Be sure to select “keep only matching records.



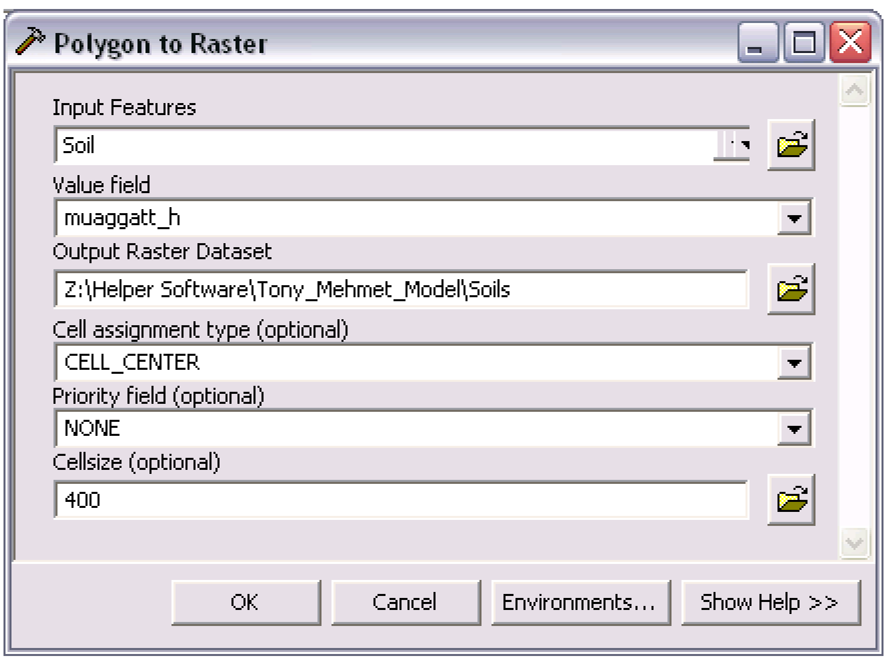
5. Use the *Analysis Tools → Extract →Clip* to extract the soil data that overlaps the watershed. The input features are **Soils.shp** and the clip features are **Watershed.shp**. Save the output as **Soil.shp**

6. LandCoverData is a raster and needs to be extracted by mask. Use *Spatial Analyst Tools →Extractions →Extract by Mask*. The input raster is LandCoverData.tif and the feature mask is Watershed.shp. Save the output as **LandCov**.



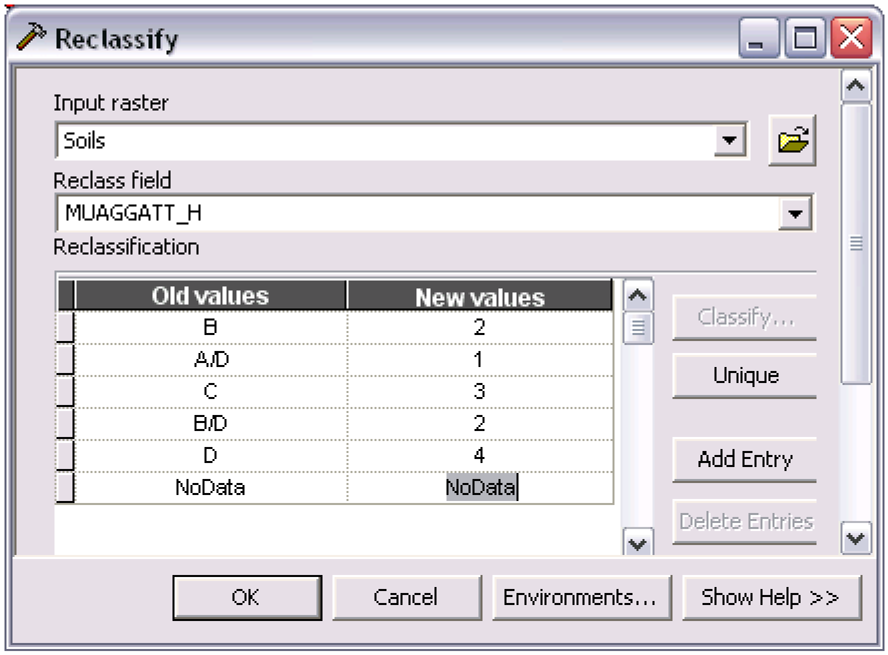


7. Convert **Soil.shp** into a raster using the *Conversion Tools → To Raster → Polygon to Raster*, specify SoilGrp as the value field. The result is a raster soil map representing the soil group numerically. Save the output grid as **Soils**.

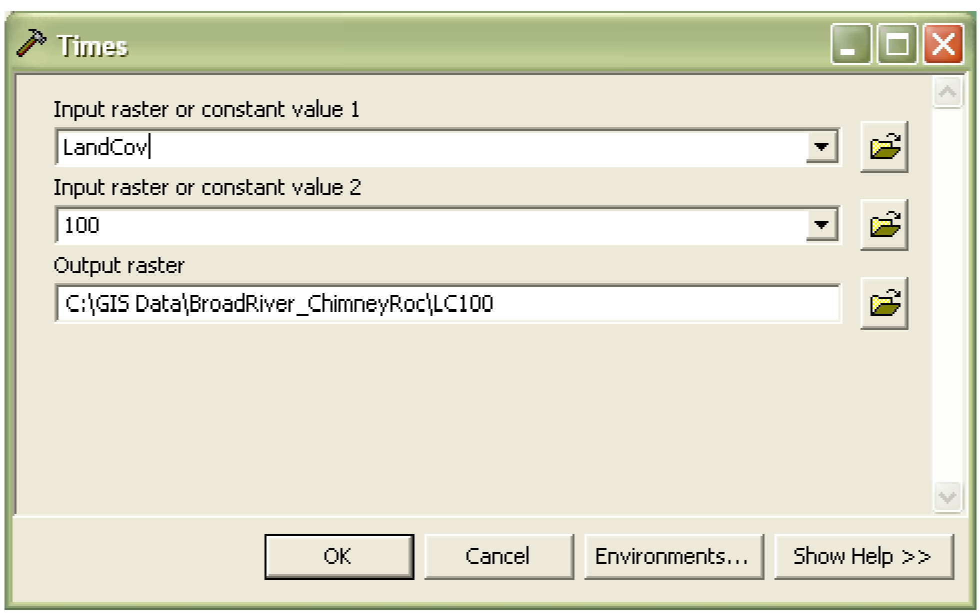


8. Create a new field in the soil attribute table named SoilGrp of type Short Integer, then use the field calculator to convert the values of *A,B,C,D to* 1,2,3,4. The easiest way to do this is using the Reclassify tool: *3d Analyst Tools → Raster Reclass → Reclassify.* Specify MUAGGATT_H as the reclass field, and then enter in numeric values for soil types, A, B, C, D,

B/D, etc… Note: It will be helpful to use the soil classification defined in the TR-55 manual, i.e. A =1, B=2,C=3,D=4. Save the Output as SoilsGrp.



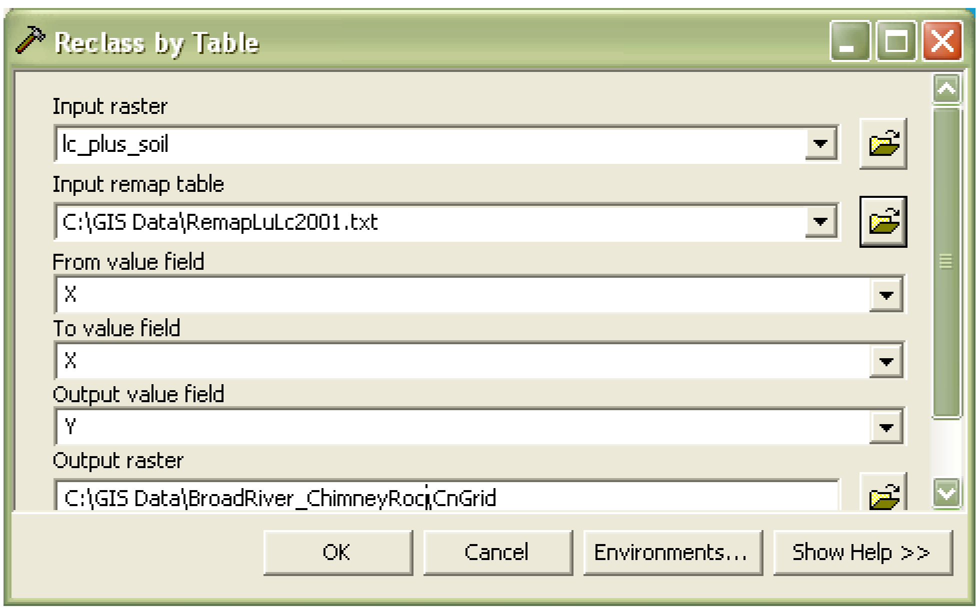
9. Multiply the **LandCov** by 100, using *3d Analyst* → *Raster Math* → *Times* from the ArcTool box. Save the output raster as **LC100**.



10. Using *3d Analyst* → *Raster Math* → *Plus,* add **LC100** and **Soils** together to form a composite land use / soil raster. Name this output **LC_plus_Soil.**

11. Next, remap these unique land uses and soil IDs to their corresponding Curve Numbers. Curve numbers must be determined by referencing TR-55 and NLCD 2001. Once Curve Numbers have been chosen there are two options for reclassifying.
    Option 1 : Using *3d Analyst → Raster Reclass → Reclassify by Table.* The input table should look similar to the one shown below, where the X column contains unique land cover/soil identifiers, and Y are their corresponding curve numbers. Be sure to check the box "change missing values to NoData". (This is just an example, the actual list should have all land cover / soil id's) Name this output **CnGrid**.

X	*Y*
2301	*77*
2302	*85*
2303	*90*
2304	*92*

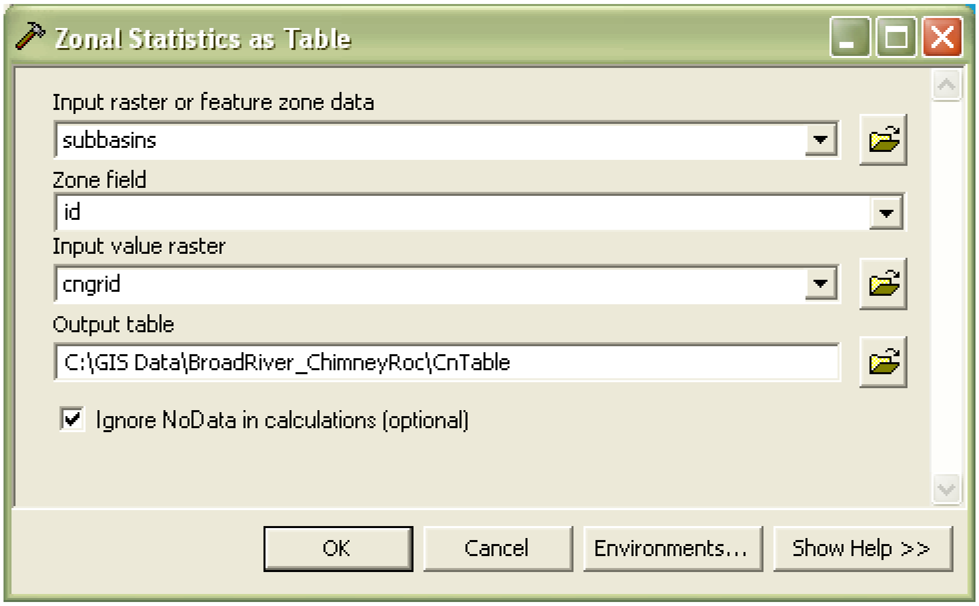


    Option 2: Use the Reclasify (3d) tool and enter the predetermined curve number as a new value. This method is good for one use only. If curve numbers are changed all values will have to been re-entered.

12. Create a new field in the **Subbasins.shp** attribute table, named ID. Using the field calculator, set its value equal to FID.

13. Next, we need to summarize this data based on the **Subbasins.shp**. To do this use *Spatial Analyst Tools → Zonal → Zonal Statistics As Table.* The zone data will be **Subbasins**, zone field is ID, and the value raster is **CnGrid**. Make sure that "ignore NoData in calculations" is

checked. This produces a table containing weight averaged curve numbers, for each subbasin. Save the output table as **CnTable.dbf**.

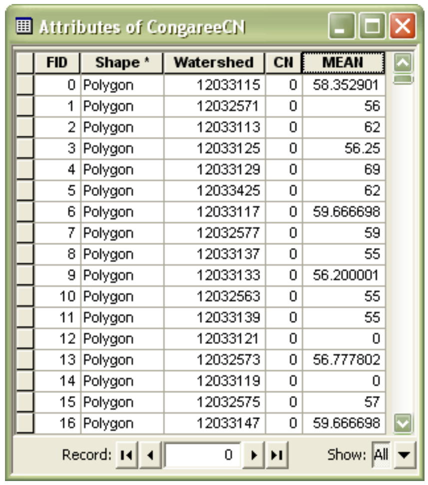


14. Right click on **Subbasins.shp** and select *Joins and Relates→Join.* Join this layer based on ID, to **CnTable.dbf** based on VALUE. Be sure to check the box "Keep only matching records".

15. Next, right click on the **Subbasins.shp** again and select *Data → Export.* Save this output layer as **CnSubasins.shp**.

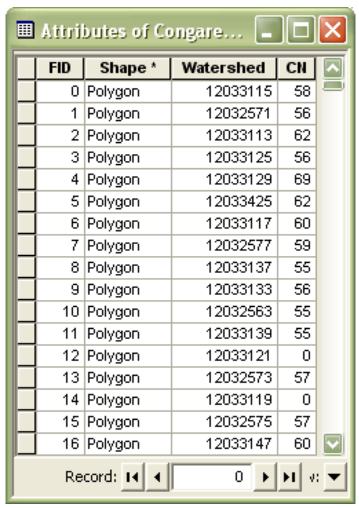
16. Add two new fields to **CnSubasins.shp** : CN (type of short integer) and Watershed (type of long integer).

17. Right click on the *Watershed* column and select *Field Calculator*. Enter [COMID] in the lower box, then click *OK*. After turning off unnecessary fields, the attribute table should look like this.



FID	Shape *	Watershed	CN	MEAN
0	Polygon	12033115	0	58.352901
1	Polygon	12032571	0	56
2	Polygon	12033113	0	62
3	Polygon	12033125	0	56.25
4	Polygon	12033129	0	69
5	Polygon	12033425	0	62
6	Polygon	12033117	0	59.666698
7	Polygon	12032577	0	59
8	Polygon	12033137	0	55
9	Polygon	12033133	0	56.200001
10	Polygon	12032563	0	55
11	Polygon	12033139	0	55
12	Polygon	12033121	0	0
13	Polygon	12032573	0	56.777802
14	Polygon	12033119	0	0
15	Polygon	12032575	0	57
16	Polygon	12033147	0	59.666698

18. Using the field calculator, set the CN column equal to MEAN. This will automatically round the MEAN values to integers. After removing the MEAN column, the attribute table should now look like this.



FID	Shape *	Watershed	CN
0	Polygon	12033115	58
1	Polygon	12032571	56
2	Polygon	12033113	62
3	Polygon	12033125	56
4	Polygon	12033129	69
5	Polygon	12033425	62
6	Polygon	12033117	60
7	Polygon	12032577	59
8	Polygon	12033137	55
9	Polygon	12033133	56
10	Polygon	12032563	55
11	Polygon	12033139	55
12	Polygon	12033121	0
13	Polygon	12032573	57
14	Polygon	12033119	0
15	Polygon	12032575	57
16	Polygon	12033147	60

19. Lastly, use *Data Management Tools → Features → Feature to* Point to convert **CnSubasins.shp** polygons into a point shapefile. Name the output file **Cn_Points.shp.** This will be the input shapefile for the curve number component.

SCS Unit Hydrograph Component

This component requires one shapefile containing point or polygon features. The attribute table (shown below) outlines the necessary fields. Area is in units of mi^2 and Tc is in units of minutes.

1. Right click on **Subbasins.shp** and *Export* the layer as **Uh_Subbasins.shp**.

2. Create a new column named Watershed (type of Double) and set it equal to COMID using the field calculator.

3. Create a new column named Area (type of Double). (Note that if a column named area already exist in the table, new column was named area_1). Right click on this column and select *Calculate Geometry.* Select *Area* in *Square Miles*. If you are not able to select square miles for the area make sure the map is in a projected coordinate system and not the default geographic coordinate system.
4. Add a new field to hold Time of concentration, Tc, [minutes] (type of double). These values were calculated outside of ArcMap and then manually entered. The resulting attribute table is shown below.

Attributes of UH_Subasins

FID	Shape *	Watershed	Area	Tc
312	Polygon	12036453	1.1970	96.332636
313	Polygon	12034331	0.6261	83.783784
314	Polygon	12036455	0.7594	75.702801
315	Polygon	12034309	0.6883	52.128367
316	Polygon	12034379	3.8085	21.303816
317	Polygon	12036391	0.5392	33.421025
318	Polygon	12036461	0.0531	33.138733
319	Polygon	12035763	0.3278	60.777602
320	Polygon	12036465	0.6884	52.485018
321	Polygon	12036385	0.0316	28.490701
322	Polygon	12034301	0.3122	42.089863
323	Polygon	12034319	4.4676	71.443744
324	Polygon	12036503	0.6699	87.927999
325	Polygon	12036393	0.4144	59.837537
326	Polygon	12036505	1.0378	35.022011
327	Polygon	12037809	0.0410	91.729331
328	Polygon	12034459	9.9607	49.825144
329	Polygon	12037807	0.0132	45.884385

Record: 0 Show: All Selected Records

Muskingum Component

This component requires 1 input shapefile, containing polyline features. Its attribute table consists of a GridID, TO_NODE, FROM_NODE, K, and X [0<=X<=0.5]. The following steps illustrate how to create this file.

1. Right click on **Reaches.shp** and export all of the features as **Musk_Reaches.shp**.

2. Create a new field called GridID (Type of Long Integer). Using the field calculator, set it equal to COMID. After removing unnecessary fields, the attribute table should look something like this.

Attributes of MUSK_STREAMS

FID	Shape *	FROMNODE	TONODE	MAVELU	GridID
0	Polyline ZM	8432704280	8432701700	3.3682	9678308
1	Polyline ZM	8432701700	8432702432	2.63349	9684210
2	Polyline ZM	8432702432	8432702435	2.45058	9684218
3	Polyline ZM	8433900087	8433900205	0.78487	9690638
4	Polyline ZM	8433900002	8433900205	0.77752	9690640
5	Polyline ZM	8433900003	8433900206	0.64371	9690642
6	Polyline ZM	8433900004	8433900207	0.44538	9690644
7	Polyline ZM	8433900005	8433900208	0.44983	9690646
8	Polyline ZM	8433900006	8433900209	0.6848	9690648

Record: 0 Show: All Selected ℓecords

3. Add a new column for the Muskingum weighting factor, X (type of float). These values should be chosen to satisfy equation stability, and range from 0 <= X <= 0.5, where 0 represents reservoir-type storage and 0.5 represents full wedge storage. For this exercise these values were entered manually. The attribute table should now look similar to the one below.

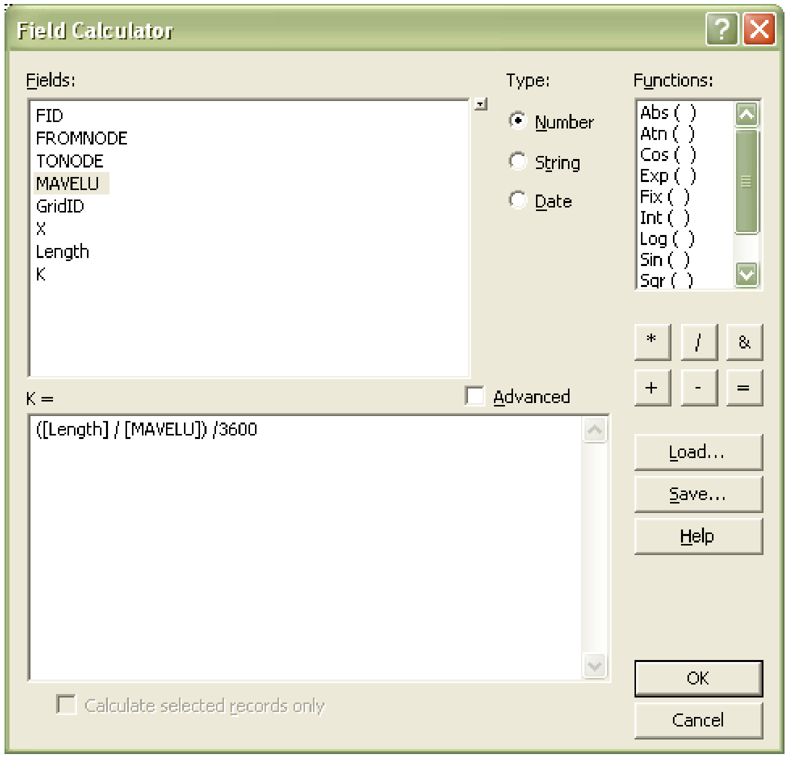
Attributes of MUSK_STREAMS

FID	Shape *	FROMNODE	TONODE	MAVELU	GridID	X
21	Polyline ZM	8433900257	8433900218	0.90869	9691080	0.2
22	Polyline ZM	8433900223	8433900219	1.10535	9691084	0.2
23	Polyline ZM	8433900219	8433900225	1.171	9691086	0.2
24	Polyline ZM	8433900225	8433900253	1.34366	9691088	0.2
25	Polyline ZM	8433900031	8433900220	0.85426	9691090	0.2
26	Polyline ZM	8433900253	8433900220	1.4477	9691092	0.5
27	Polyline ZM	8433900220	8433900230	1.79393	9691094	0.2
28	Polyline ZM	8433900036	8433900222	0.98317	9691096	0.5
29	Polyline ZM	8433900256	8433900221	1.06268	9691098	0.2
30	Polyline ZM	8433900221	8433900222	1.32585	9691100	0.1

Record: 10 Show: All Selected Records (0

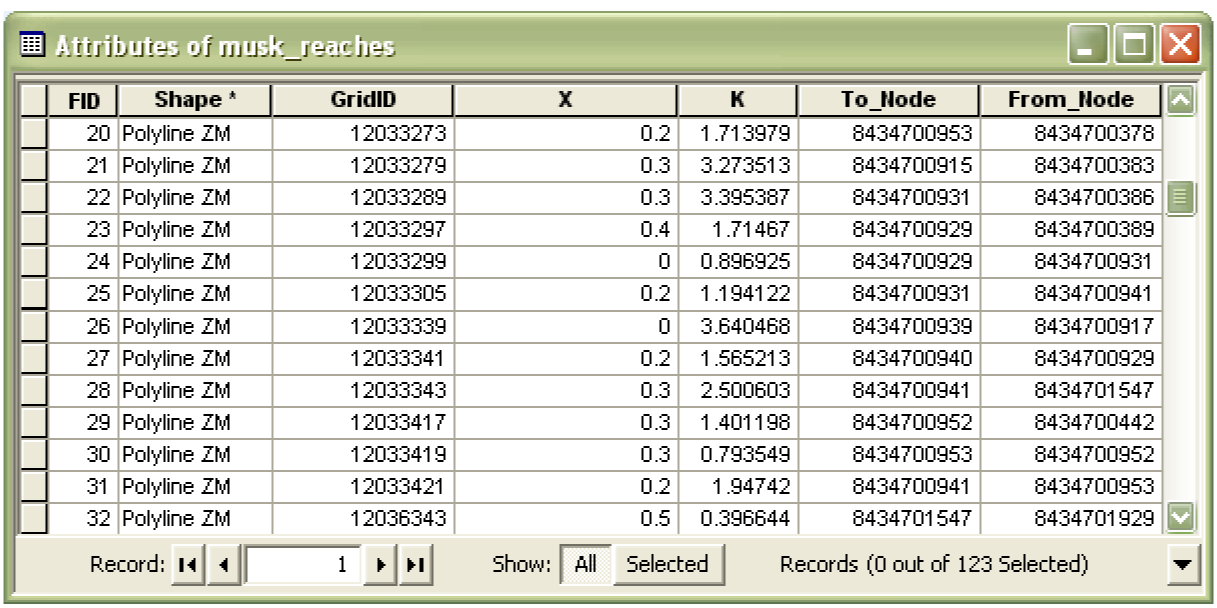
4. Create a new field to hold the features length, called LENGTH (type of double). Set this column equal to the feature's length, *Options* → *Calculate Geometry* → *Length [ft]*.

5. Create a new field for proportionality coefficient, K (type of double). This can be estimated as the time of travel of a flood wave through the reach. To estimate this parameter we can divide the feature's length (LENGTH) by the mean annual velocity (MAVELU) and then divide by 3600 to convert into hours. This will give a rough estimate for reach travel time.



6. Next, create two new columns: To_Node (type of Long) and From_Node (type of Long). Using the field calculator, set To_Node equal to TONODE and From_Node equal to FROMNODE. Note that if you are unable to set the new nodes equal while the type is long, try changing the type to double instead.

7. After turning off unnecessary fields, the attribute table should look similar to this:



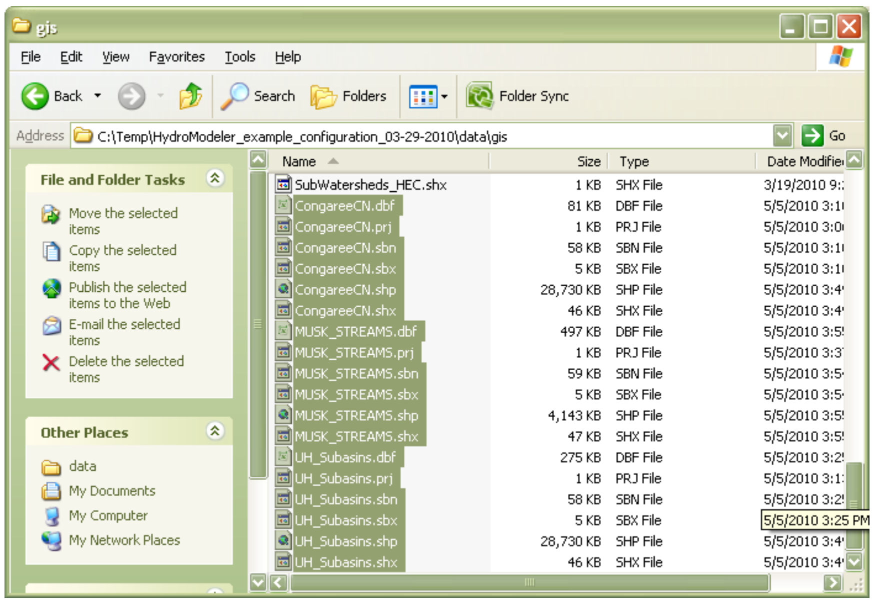
Attributes of musk_reaches

FID	Shape *	GridID	X	K	To_Node	From_Node
20	Polyline ZM	12033273	0.2	1.713979	8434700953	8434700378
21	Polyline ZM	12033279	0.3	3.273513	8434700915	8434700383
22	Polyline ZM	12033289	0.3	3.395387	8434700931	8434700386
23	Polyline ZM	12033297	0.4	1.71467	8434700929	8434700389
24	Polyline ZM	12033299	0	0.896925	8434700929	8434700931
25	Polyline ZM	12033305	0.2	1.194122	8434700931	8434700941
26	Polyline ZM	12033339	0	3.640468	8434700939	8434700917
27	Polyline ZM	12033341	0.2	1.565213	8434700940	8434700929
28	Polyline ZM	12033343	0.3	2.500603	8434700941	8434701547
29	Polyline ZM	12033417	0.3	1.401198	8434700952	8434700442
30	Polyline ZM	12033419	0.3	0.793549	8434700953	8434700952
31	Polyline ZM	12033421	0.2	1.94742	8434700941	8434700953
32	Polyline ZM	12036343	0.5	0.396644	8434701547	8434701929

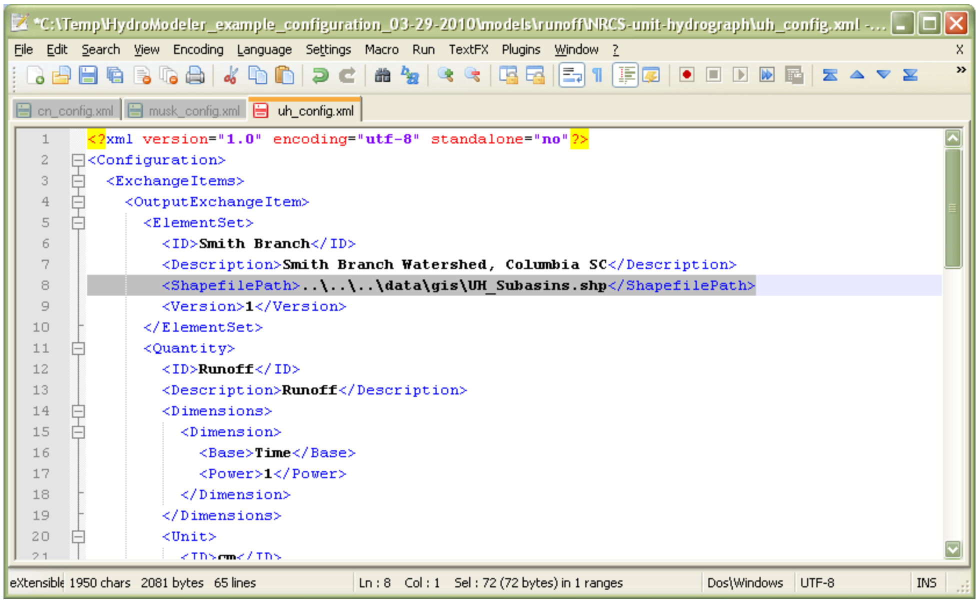
Record: 1 Show: All Selected Records (0 out of 123 Selected)

3. Creating a New Model Composition

1. Copy all files associated with the three shapefiles that were created in the previous sub-sections (cn_points, uh_subbasins, and musk_reaches) into the HydroModeler example directory. C:\Temp*HydroModeler_Example_Configuration* → *data* → *gis.*

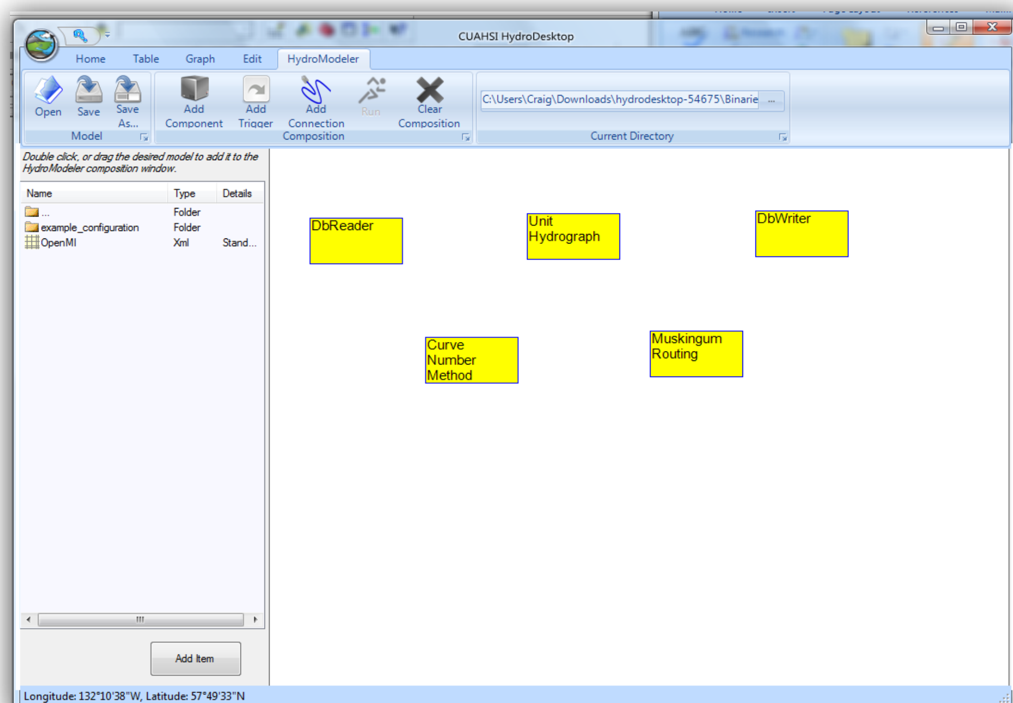


2. Navigate to C:\Temp*HydroModeler_Example_Configuration* → *models*, and open the configuration files for the Curve Number, Unit Hydrograph, and Muskingum components. Change the *ShapeFilePath* element with these files to point to the new shapefiles.



3. Open HydroDesktop and load the HydroModeler Plugin (as outlined in section 2).

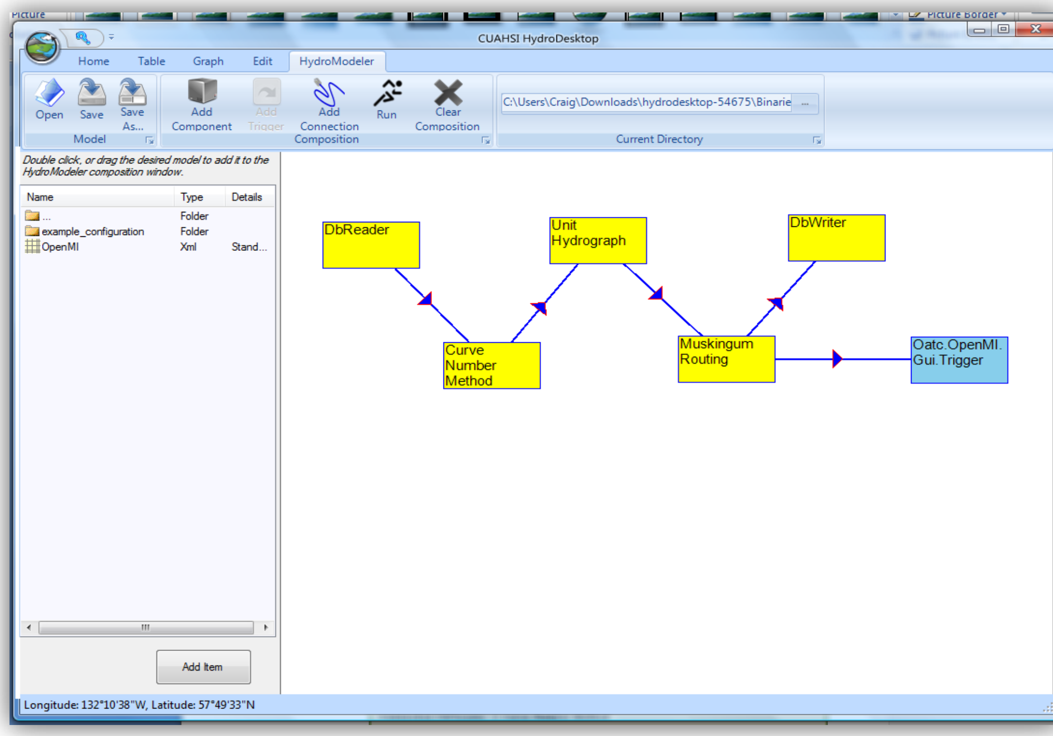
4. Right click on the HydroModeler workspace and select *Add Model*. Navigate to *C:\Temp\HydroModeler_Example_Configuration\models* and add all three models, as well as the DbReader and DbWriter components.



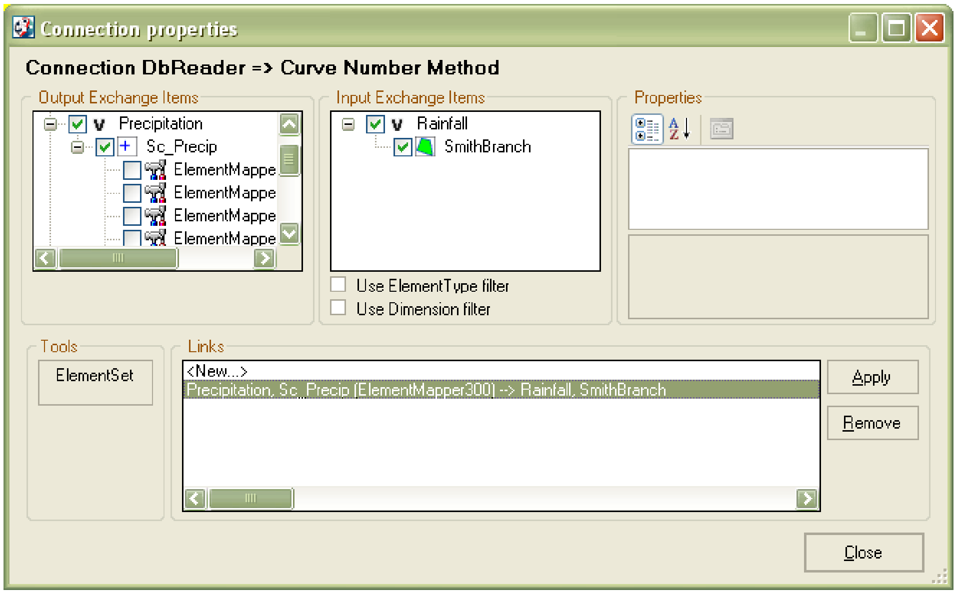
5. Right click in the HydroModeler workspace and select *Add Trigger*

6. Right click in the HydroModeler workspace and select *Add Connection.* Next, click on the DbReader to assign it as the source component and then click on the Curve Number Method to assign it as the target component. Repeat this to establish linkages between all components.

   NOTE: Retrieval of data for the DbReader component is NOT covered. The remainder of the tutorial assumes that proper rainfall data has been downloaded.



7. Click on each arrow to open a connection properties window. Define the output exchange item that will be supplied to an input exchange item. Repeat for all links.



8. Save the model by clicking *Save as…*

9. Right click in the HydroModeler workspace and select *Run*.

10. Select *Set all* within Events listened during calculation, click *Latest overlapping* to determine the simulation end time, and finally click *RUN!!*

4. Component Definitions

SCS CN Method

Inputs Files

*.OMI:

Argument Description	Key	Value
Configuration File	ConfigFile	Path the Configuration.xml
Output Directory	OutDir	Path to output file

Shapefile Input (each row = 1 element):

Column ID	Type	Description
Watershed	Short Integer	Watershed ID Number
CN	Short Integer	Curve Number

SCS Unit Hydrograph Method

Inputs Files

*.OMI:

Argument	Key	Value
Configuration File	Config	Path the Configuration.xml
Output Directory	OutDir	Path to output file

Shapefile Input (each row = 1 element):

Column ID	Type	Description
GridID	Short Integer	Subbasin ID Number
Area	Double	Subbasin Area [mi^2]
Tc	Double	Time of Concentration [min]

Muskingum Method

Input Files

*.OMI:

Argument	Key	Value
Configuration File	Config	Path the Configuration.xml
Output Directory	OutDir	Path to output file

Shapefile Input (each row = 1 element):

Column ID	Type	Description
GridID	Short Integer	Reach ID Number
FROM_NODE	Short Integer	ID of the Start Node
TO_NODE	Short Integer	ID of the Target Node
X	Double	Weighting Factor $0 \leq X \leq 0.5$
K	Double	Proportionality Coefficient

DbReader

*.OMI:

Argument	Key	Value
Database Path (optional)	DbPath	Path to desired HydroDesktop Database
Relaxation Factor	Relaxation	Value from 0 to 1

DbWriter

*.OMI:

Argument	Key	Value
Database Path (optional)	DbPath	Path to desired HydroDesktop Database

Green Amp Infiltration

*.OMI:

Argument	Key	Value
Configuration File	ConfigFile	Path to configuration file
Output Directory	OutDir	Path to output directory

Shapefile Input (each row = 1 element):

Column ID	Type	Description
Ks	Double	Saturated Hydraulic Conductivity [mm/hr]
Suction	Double	Average Suction Head [mm]
Porosity	Double	Porosity
FieldCapac	Double	Field Capacity
WiltingPt	Double	Wilting Point
DepStorage	Double	Depression Storage [mm]
ID	Short Int	Element ID

Part 3 – Creating a Component in Microsoft Visual Studio

Up to this point we have run an example simulation in HydroDesktop and created a new model composition in ArcMap. In this section we will walk through a sample component to gain an understanding of how creating a specialized component in Visual Studio C# can be useful. The sample component is a shell showing how code should be written to run in HydroDesktop. This component can be modified by the user to run more meaningful hydrological applications.

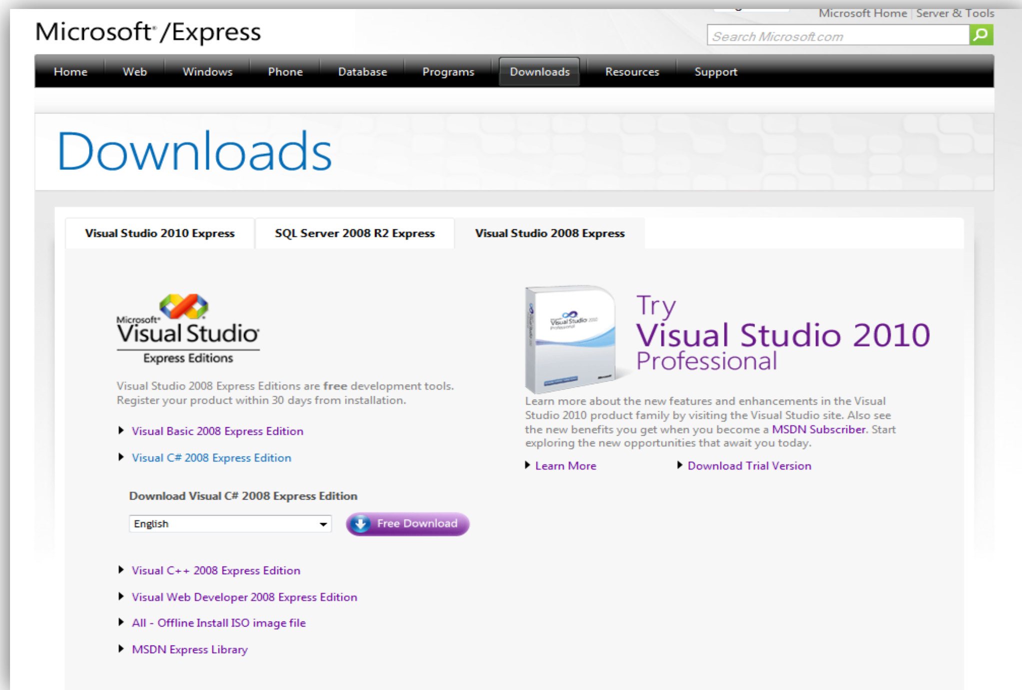
C# is an object oriented programming language designed for building a wide variety of applications that run on the .NET Framework. The .NET Framework is a Windows component that supports building and running of applications and Web services. The key components of the .NET Framework are the common language runtime and the class library. The class library is a collection of classes, interfaces, and value types that are included in the Microsoft .NET Framework. The class library is designed to be the foundation on which the .NET Framework applications, components, and controls are built.

1. Getting started with Visual C#

If you are unfamiliar with programming in C#, this section will provide you with basic assistance in obtaining a free development environment for writing, compiling, and debugging C# code.

1. If you do not have access to Microsoft Visual Studio you can download Microsoft Visual C# 2008 for free from http://www.microsoft.com/express/Downloads/

2. Choose the tab Visual Studio 2008 Express. Then select Visual C# 2008 Express Edition. Click Download and follow instructions to install.

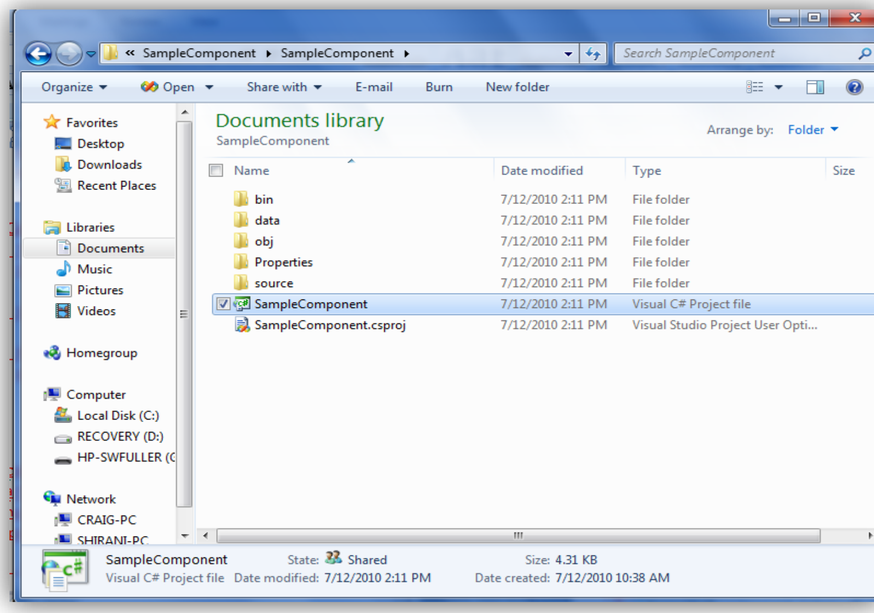


3. If you are learning C# or need a refresher there are many tutorials available on-line that can help. The address below will take you to Microsoft's Development Network were they have a beginners learning center. There are several links available here for learning Visual Studio C# and Visual Studio Basic: http://msdn.microsoft.com/en-us/beginner/bb308734.aspx

2. Sample Component

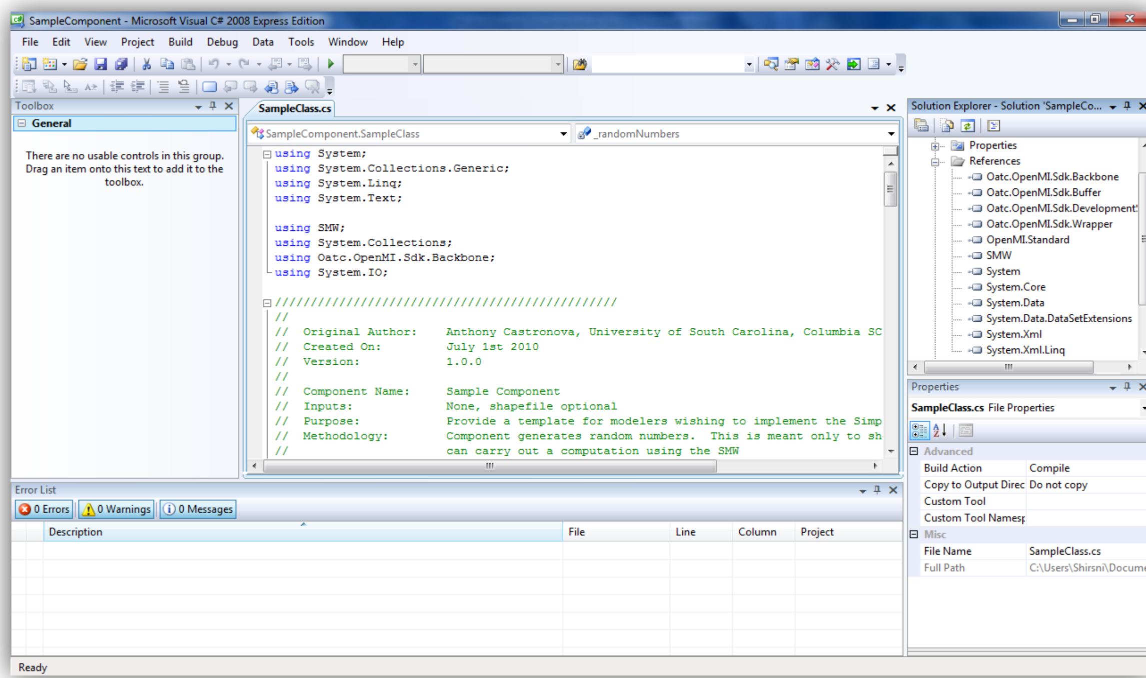
1. A sample component that contains comments on each part of the model component development process is available on the HydroDesktop website under the Documentation tab. Download this component and extract the files to a directory (e.g., c:\temp).

2. Navigate through the Sample Component folder to the Sample Component Visual C# project file. Double click this file to open in Visual C#.



Setting References

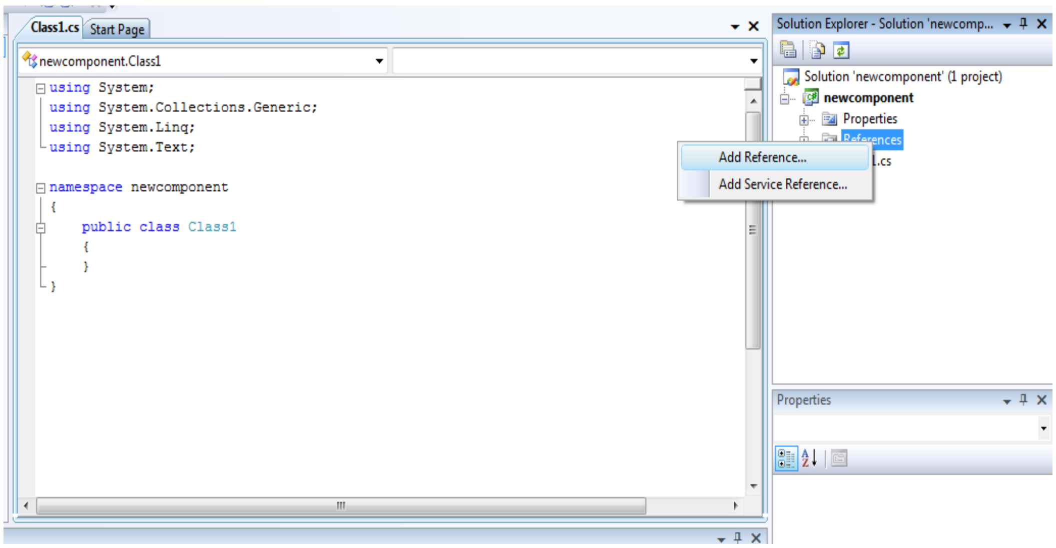
A reference is a file that contains external code required to run the code you are writting. Usually a reference identifies a dll (Dynamic Linking Library) file. The references can be viewed by expanding the *Reference* option listed in the Solution Explorer window located on the upper right side of the page.



When you expand the References you will see that all of the references you need to run the sample component are listed here already. The physical location of these files is Sample Component→ bin → debug

If you need to add a reference to your application:

1. Right click on *Reference* and select *Add Reference*.

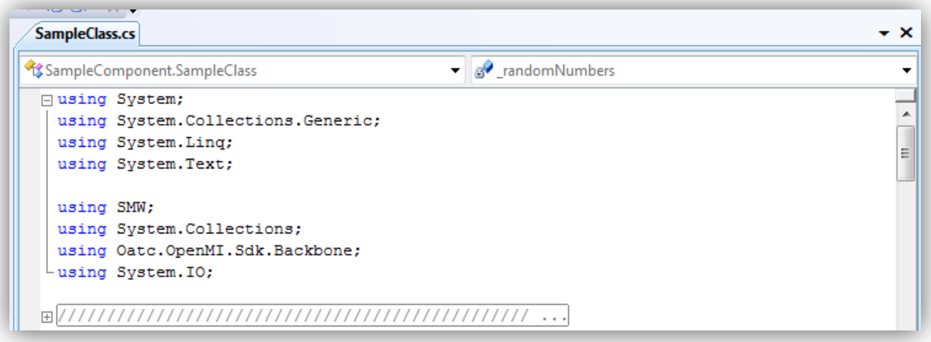


2. A window will pop up. Using the tabs, browse to the location of the dll files to be added.

You do not need to add references to the sample component project because we have already done this for you. If you were starting a new component from scratch, however, it would be necessary to add references to the project in this way.

Namespaces

Namespaces allow you a way to organize code within a software system. The "using" directive can be implemented as a way of accessing members of a namespace without having to type out their full name each time they are used. When a new project is created several common namespaces are inserted. In the sample component you can see these defaults listed at the top of the code. Directly below them you will see the namespaces that were added to the code with the "using" directive.



Methods

There are several major parts to this code:

1. Defining the global variables- In this section variables are defined using specific data types such as string, integer, double, and Boolean

2. The Finish method- This section of code tells the application to write output files based on data acquired during the simulation.

3. The Initialize method- Gives the application instructions on operations that need to be preformed prior to running the simulation. This section locates the configuration file and sets internal variables in OpenMI.

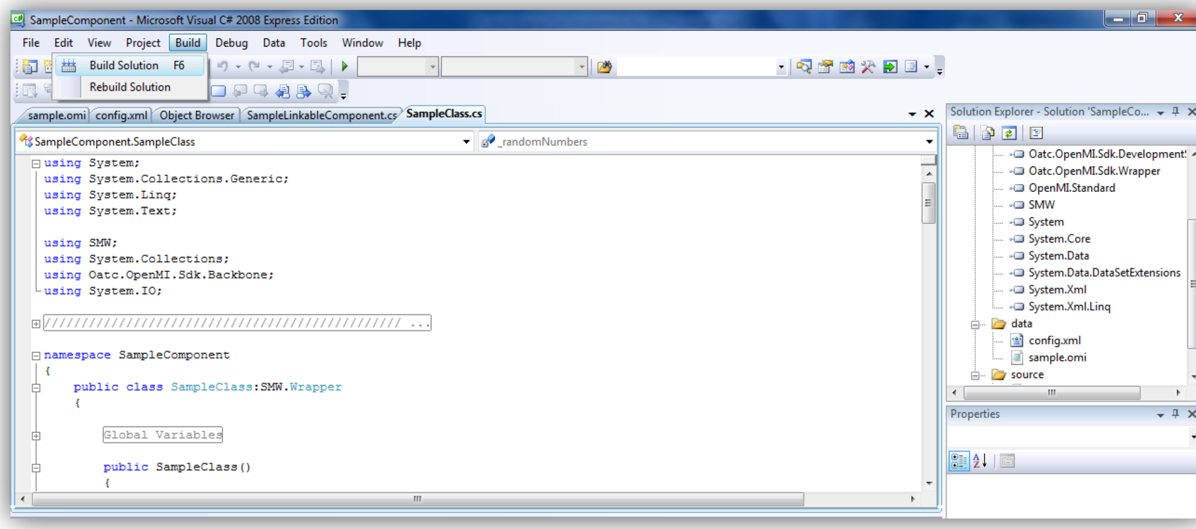
4. The Perform Time Step method- This method is used to perform calculations during simulation runtime. For the Sample Component the calculation section is written so that the application simply generates random numbers.

Within the Finish method there is code telling the application where to write the output file. This line should be changed now to specify where you would like the output file. If you choose not to change the code then by default the output text file created by HydroModeler will go up two directories from where you run HydroDesktop.

Compiling

The next steps involve compiling the application. Compiling is the process of converting written code into an executable file that the computer can run. If you have downloaded the sample component, you should be able to build this project without modification and complete the remaining portion of the tutorial.

In Visual Studio C# select *Build* → *Build Solution*

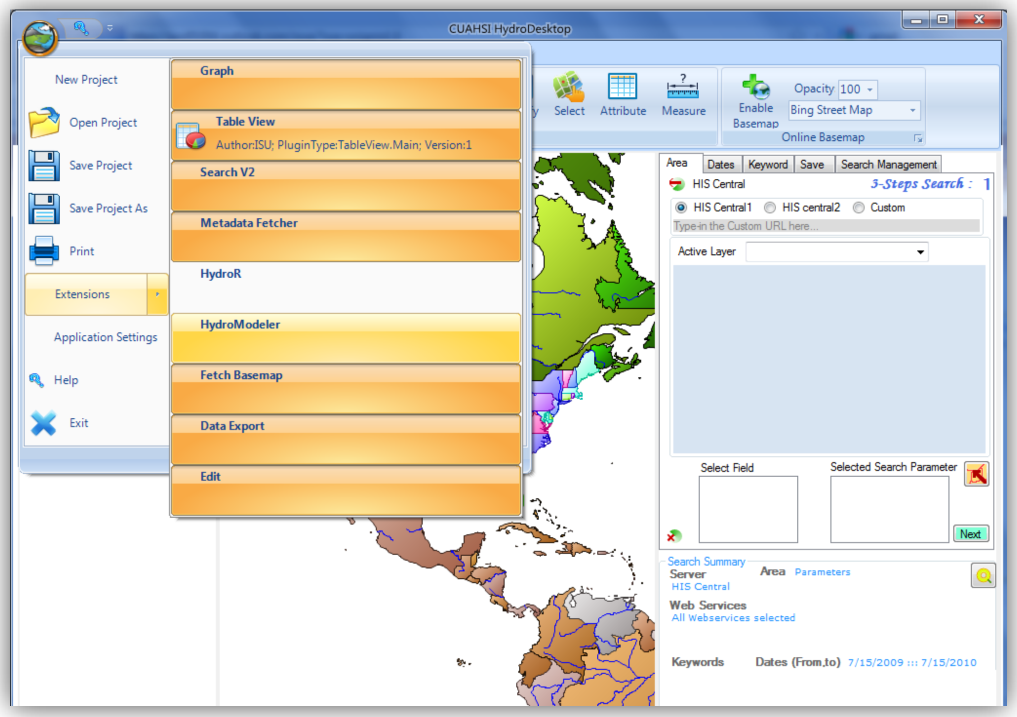


The Building function will check for any errors and alert you to them in the Error List window at the bottom of the screen.

HydroDesktop

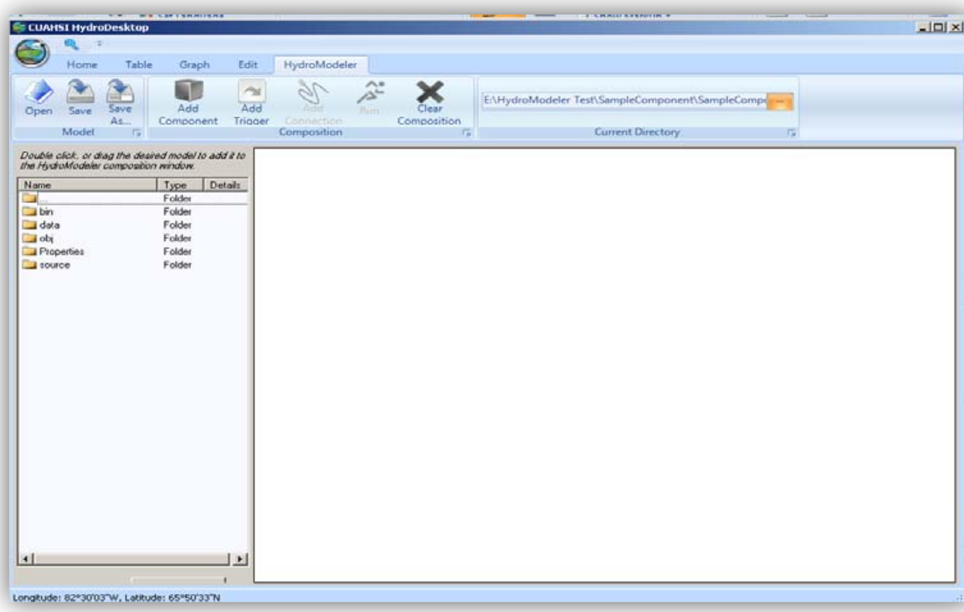
1. Start HydroDesktop

2. Click the icon in the upper left corner of the screen. Then select Extensions→HydroModeler

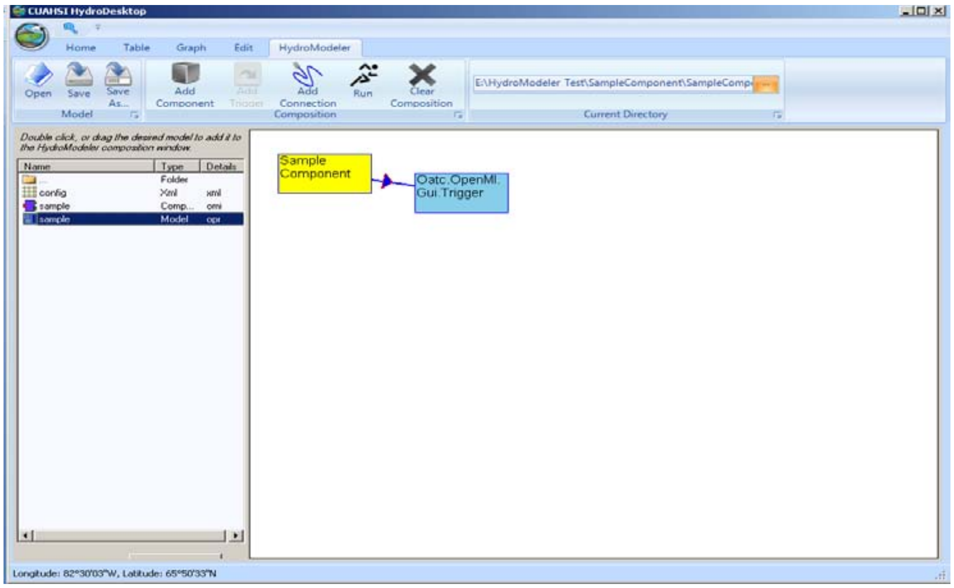


3. Using the drop down tab next to Current Directory, change the directory to the location of your Sample Component files

4. Once you have the directory pointing to the Sample Component a list of folders will appear to the left side of the screen

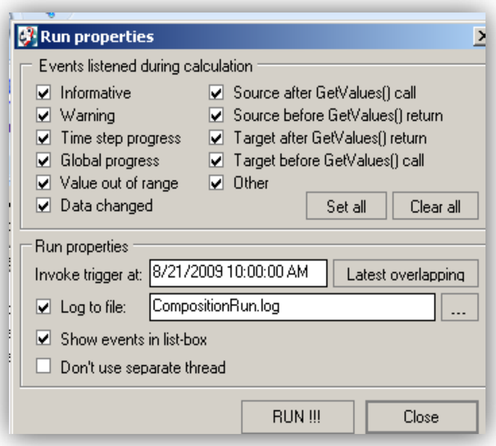


5. To add the Sample Component Model (opr file) double click on the *data* folder then on the *sample* (type = model).



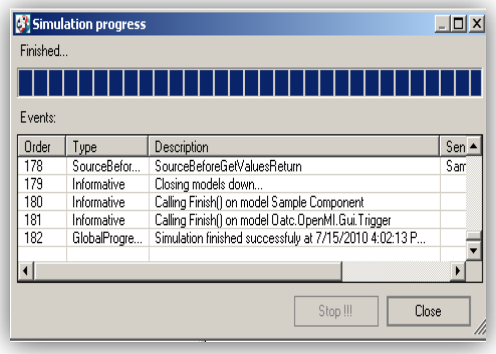
6. A yellow box will appear labeled Sample Component with a link to a trigger. Note that if you opened the omi file, then you opened just the sample model component and you will only see the yellow box.

7. Now, select the Run button at the top of the page or you can right click within the workspace and select run.



8. Leave all of the boxes checked and click RUN.

9. A progress window should appear. Once the program is finished scroll down and the last line should read simulation finished successfully.



10. You can select Close and Yes to the next window that appears.

You have just built (or compiled) your first model component. Going through the source code and getting familiar with C#, you can modify the sample component to perform a hydrologic calculation. There are a growing number of sample components available in the HydroDestkop source code that provide good working examples.

Source Code: http://hydrodesktop.codeplex.com/SourceControl/list/changesets
Click “Browse” to view the code online or “Download” to obtain a copy of the source code.
You will see sample HydroModeler components in the following folder:
\Source\Plugins\HydroModeler\Components