Using HEC-RAS and HEC-GeoRAS for River Modeling
Adapted by E. Maurer, using an exercise by V. Merwade, Purdue Univ.

Introduction

This tutorial is designed to expose you to basic functions in HEC-GeoRAS for pre- and/or post-processing of GIS data and HEC-RAS results for flood inundation mapping using ArcGIS. It is expected that you are at least conceptually familiar with HEC-RAS, and that you have some experience with ArcGIS. For details of HEC-GeoRAS that are not covered in this tutorial please refer to the HEC-GeoRAS users manual.

Computer Requirements

You must have a computer with windows operating system, and the following programs installed:

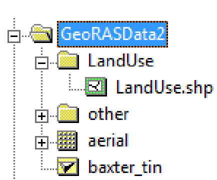
1. ArcGIS 10
2. HEC-GeoRAS
3. HEC-RAS

You can download HEC-RAS and HEC-GeoRAS for free from the US Army Corps of Engineers Hydrologic Engineering Center website http:/www.hec.usace.army.mil/software

Data Requirement

The only essential dataset required for HEC-GeoRAS is the terrain data (TIN or DEM). Additional datasets that may be useful are aerial photograph (s) and land use information. The dataset supplied with this tutorial includes a small portion of the fictional Baxter River available with HEC-GeoRAS users manual. The data (about 10 MB) required for this tutorial is available at the class website. **Download** the zip file on your local drive, and **unzip** its contents using 7-zip.

Open ArcCatalog and look at the directory where the zip file was extracted. The GeoRASData folder should contain two sub-folders, one TIN dataset, and one aerial image (as raster grid) as shown below:

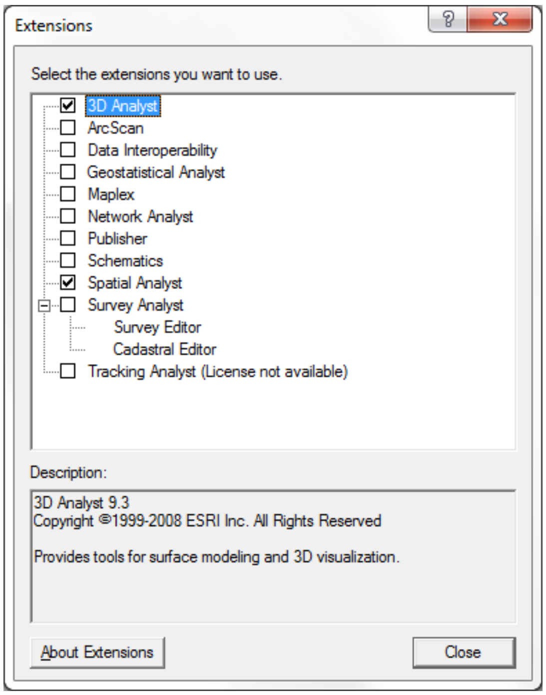


The *LandUse* folder contains a shapefile with land use data, *aerial* is the aerial image of the study area, and *baxter_tin* is the TIN dataset for the study area.

Getting Started

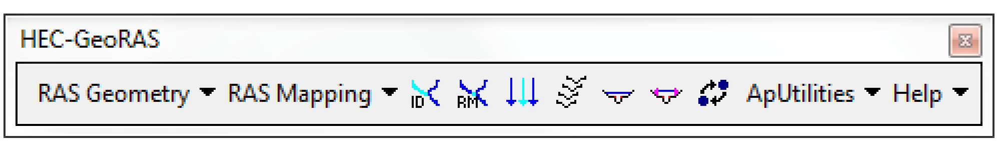
Start ArcMap by **clicking** *Start→Programs→ArcGIS→ArcMap*. Save the ArcMap document (by **clicking** *File→Save As..*) as *baxter_georas.mxd* in your working folder.

Since Hec-GeoRAS uses functions associated with ArcGIS *Spatial Analyst* and *3D Analyst* extensions, make sure these extensions are available, and are enabled. You can check this by **clicking** on *Tool→Extensions..,* and **checking** the boxes (if they are unchecked) next to *3D Analyst* and *Spatial Analyst* as shown below:



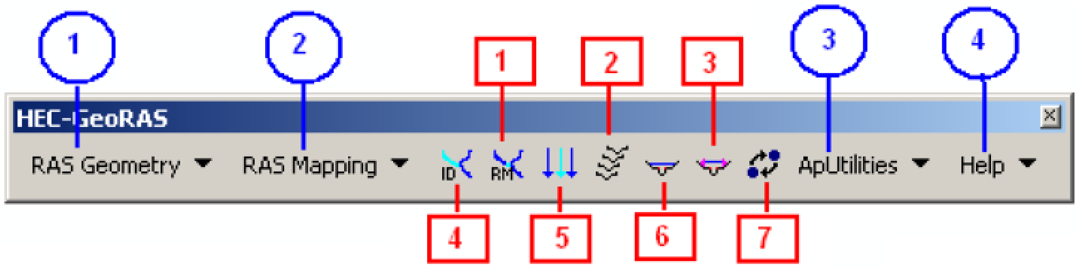
Close the Extensions window.

Now load the HEC-GeoRAS toolbar into ArcGIS by **clicking** on *View→Toolbars→HEC-GeoRAS* to see the toolbar as shown below:



You can either leave the HEC-GeoRAS toolbar on the map or dock it with other toolbars as desired.

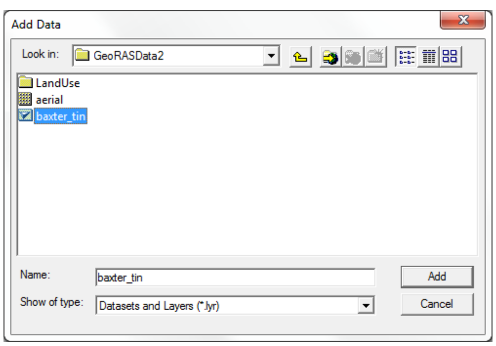
The HEC-GeoRAS toolbar has four menus (RAS Geometry, RAS Mapping, ApUtilities, Help) and seven tools/buttons (Assign RiverCode/ReachCode, Assign FromStation/ToStation, Assign LineType, Construct XS Cutlines, Plot Cross Section, and Assign Levee Elevation) as shown in circles and boxes, respectively in the figure below.



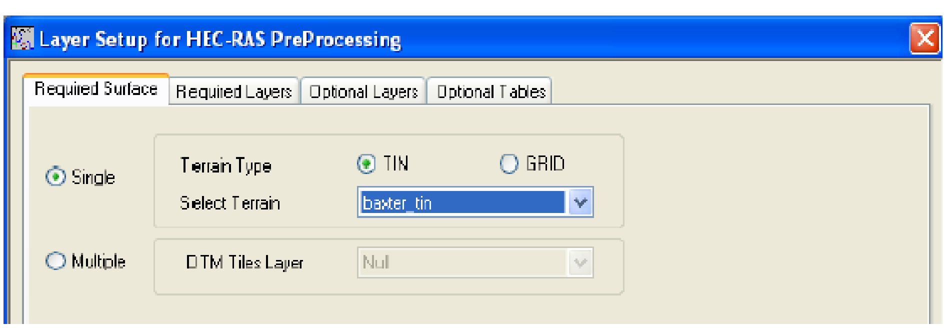
The *RAS Geometry* menu contains functions for pre-processing of GIS data for input to HEC-RAS. The *RAS Mapping* menu contains functions for post-processing of HEC-RAS results to produce flood inundation map. The *ApUtilites* menu contains functions mainly for data management. The *Help* menu is self-explanatory. You will learn about functions associated with these menus and buttons in the following sections.

Setting up Analysis Environment for HEC-GeoRAS

Using GIS for hydrologic/hydraulic modeling usually involves three steps: 1) pre-processing of data, 2) model execution, and 3) post-processing/visualization of results. To create a geometry file, you need terrain (elevation) data. **Click** on Add button  in ArcMap, and **browse** to *baxter_tin* to add the TIN to the map document.



You must have the same coordinate system for all the data and data frames used for this tutorial (or any GeoRAS project). Because *baxter_tin* already has a projected coordinate system, it is applied to the data frame. You can check this by right-clicking on the data frame and looking at its properties. Next, click on RAS Geometry→Layer Setup. Select baxter_tin as the single TIN in the Required Surface tab, and click OK.

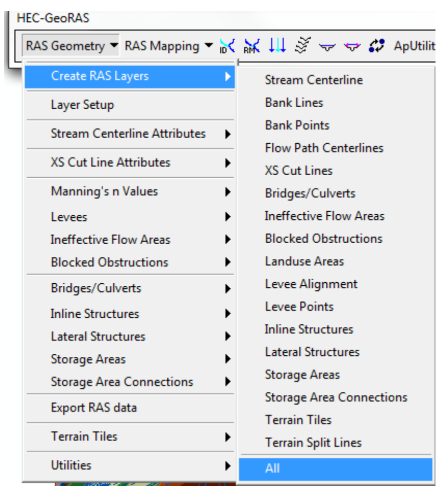


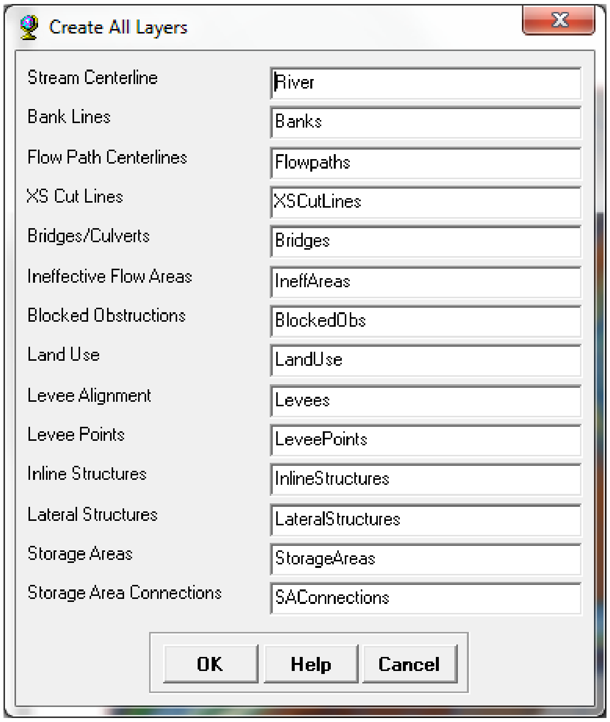
You must have the same coordinate system for all the data and data frames used for this tutorial (or any GeoRAS project). Because *baxter_tin* already has a projected coordinate system, it is applied to the BaxterGeometry data frame. You can check this by right-clicking on the data frame and looking at its properties.

Part 1: Creating RAS Layers

The geometry file for HEC-RAS contains information on cross-sections, hydraulic structures, river banks and other physical attributes of river channels. The pre-processing using HEC-GeoRAS involves creating these attributes in GIS, and then exporting them to the HEC-RAS geometry file. In HEC-GeoRAS, each attribute is stored in a separate feature class called as RAS Layer. So before creating river attributes in GIS, first create empty GIS layers using the RAS Geometry menu on the HEC-GeoRAS toolbar.

Click on *RAS Geometry→Create RAS Layers*. You will see a list of all the possible attributes that you can have in the HEC-RAS geometry file. If you wish, you can click on individual attribute to create a single layer at a time, or you can click on All to create all layers. For this tutorial, **click** on *ALL* to create all layers.





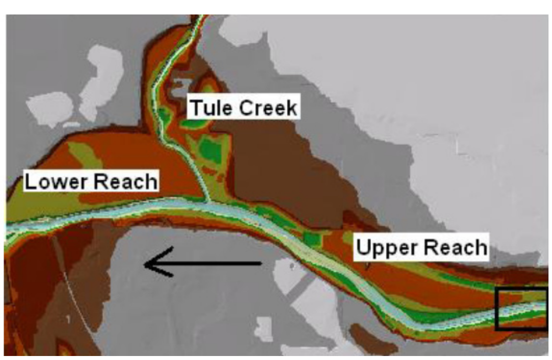
In the *Create All Layers* window, accept the default names, and **click** *OK*.

HEC-GeoRAS creates a geodatabase in the same folder where the map document is saved, gives the name of the map document (the .mxd file) to the geodatabase (Baxter_georas.mdb, in this case), and stores all the feature classes/RAS layers in this geodatabase.

After creating RAS layers, these are added to the map document with a pre-assigned symbology. Since these layers are empty, our task is to populate some or all of these layers depending on our project needs, and then create a HEC-RAS geometry file.

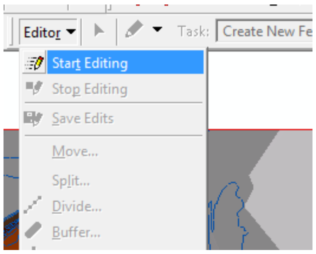
Creating River Centerline

The river centerline is used to establish the river reach network for HEC-RAS, and is a good starting reference point. The baxter_tin dataset has the Baxter River flowing from east to west with Tule Creek as a tributary. So there are three reaches: upper Baxter River, lower Baxter River and Tule Creek Tributary as shown below:

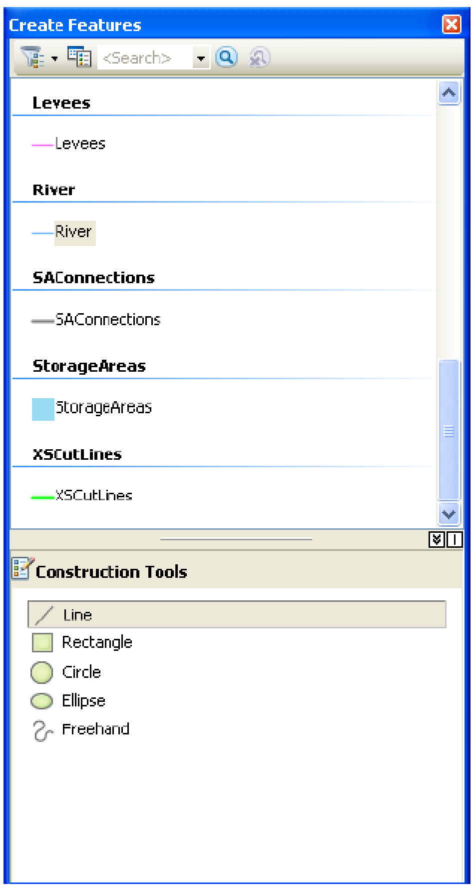


We will create/digitize one feature for each reach approximately following the center of the river, and aligned in the direction of flow. **Zoom-in** to the most upstream part of the upper Baxter reach to see the main channel (black outline shown in the above figure).

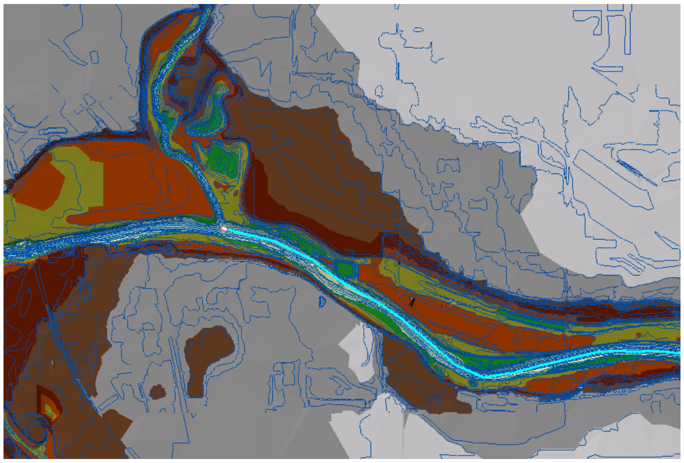
To create the river centerline (in River feature class), **start editing**.

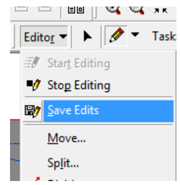


In the create features window, select River for features and Line for construction tools as shown in the figure below.

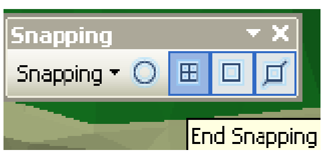


Using the straight segment tool on the Editor, **start digitizing** the river centerline from the upstream end of the upper Baxter River reach towards the downstream until you reach the intersection with the Tule Creek tributary. Each click creates a vertex on the line. When you reach the intersection, double click to complete the centerline for the upper reach of the Baxter River. If you need to pan (to slide the image), click the pan tool, pan through the map and then continue by clicking the sketch tool (do not double-click until you reach the junction). After finishing digitizing the upper Baxter Reach, **save the edits**.



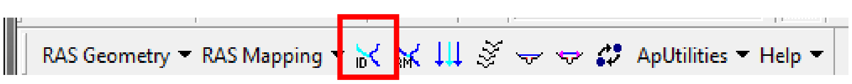


Before you start digitizing the Tule Creek tributary, modify some editing options. **Click** on *Editor*→*Snapping*→*Snapping Toolbar*, and select the check the *End* Snapping option as shown below.

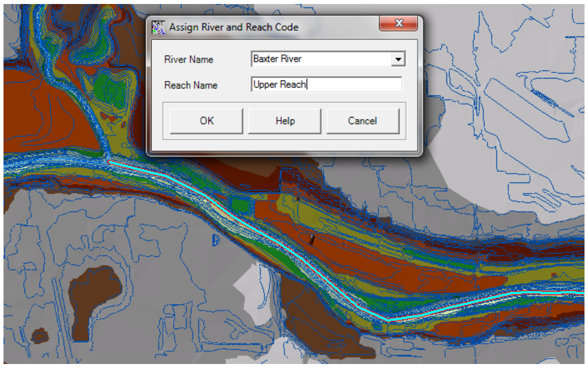


We are selecting this snapping option because when we digitize the Tule tributary we want its downstream end coincide with the downstream end of the upper Baxter Reach. Now **start digitizing** the Tule Tributary from its upstream end towards the junction with the Baxter River. When you come close to the junction, **zoom-in,** and you will notice that the tool will automatically try to snap to the downstream end of the upper Baxter Reach. **Double click** at this point to finish digitizing the Tule Tributary. **Save edits**. Finally, **digitize** the lower Baxter reach from junction with the Tule Tributary to the most downstream end of the Baxter River. Again make sure you snap the starting point with the common end points of Upper Baxter Reach and Tule Tributary. **Save edits**, and **stop editing**. Snapping of all the reaches at the junction is necessary for connectivity and creating a junction that HEC-RAS will use to define the intersection of these lines.

After the reaches are digitized, the next task is to name them. Each river in HEC-RAS must have a unique river name, and each reach within a river must have a unique reach name. We can treat the main stem of the Baxter River as one river and the Tributary as the second river. To assign names to reaches, **click** on *Assign RiverCode/ReachCode* button to activate it as shown below:



With the button active, **click** on the upper Baxter River reach. You will see the reach will become selected, invoking the following window:



Assign the River and Reach name as *Baxter River* and *Upper Reach*, respectively, and **click** *OK*. **Click** on the tributary reach, and use *Tule Creek* and *Tributary* for River and Reach name, respectively. For lower Baxter River, use *Baxter River* and *Lower Reach* for River and Reach name, respectively.

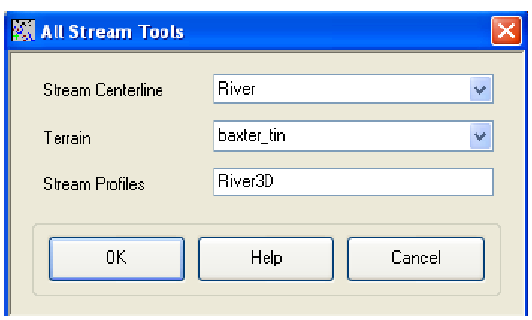
Now open the attribute table of River featureclass, and you will see that the information you just provided on river and reach names is entered as feature attributes as shown below:

Attributes of River

Shape *	OBJECTID *	Shape_Length	HydroID	River	Reach	FromNode	ToNode	ArcLength	FromSta	ToSta
Polyline	1	7435.439388	1	Baxter River	Upper Reach	<Null>	<Null>	<Null>	<Null>	<Null>
Polyline	2	4531.744298	2	Tule Creek	Tributary	<Null>	<Null>	<Null>	<Null>	<Null>
Polyline	3	7119.555121	3	Baxter River	Lower Reach	<Null>	<Null>	<Null>	<Null>	<Null>

Also note that there are still some unpopulated attributes in the River feature class (FromNode, ToNode, etc.). You can close the attribute table.

Before we move forward let us make sure that the reaches we just created are connected, and populate the remaining attributes of the River feature class. **Click** on *RAS Geometry→Stream Centerline Attributes→All*



Confirm *River* for *Stream Centerline* and *baxter_tin* for *Terrian TIN*, *River3D* for *Stream Profiles*, and **click** *OK*. This function will populate the FromNode and ToNode attribute of the River feature class. This will populate all attributes in River, and will also create 3D version (new feature class) of River centerline called River3D. Now open the attribute table for River, and understand the meaning of each attribute.

Attributes of River

Shape *	OBJECTID *	Shape_Length	HydroID	River	Reach	FromNode	ToNode	ArcLength	FromSta	ToSta
Polyline	1	7435.439388	1	Baxter River	Upper Reach	1	2	7435.44	7119.5552	14554.9
Polyline	2	4531.744298	2	Tule Creek	Tributary	3	2	4531.744	0	4531.74
Polyline	3	7119.555121	3	Baxter River	Lower Reach	2	4	7119.555	0	7119.55

Record: 1 Show: All Selected Records (0 out of 3 Selected) Options

HydroID is a unique number for a given feature in a geodatabase. The *River* and *Reach* attributes contain unique names for rivers and reaches, respectively. The *FromNode* and *ToNode* attributes define the connectivity between reaches. *ArcLength* is the actual length of the reach in map units, and is equal to Shape_Length. In HEC-RAS, distances are represented using station numbers measured from downstream to upstream. For example, each river has a station number of zero at the downstream end, and is equal to the length of the river at the upstream end. Since we have only one reach for Tule Creek tributary the *FromSta* attribute is zero and the *ToSta* attribute is equal to the *ArcLength*. Since the Baxter River has two reaches, the *FromSta* attribute for Upper Reach = *ToSta* attribute of lower reach, and the *ToSta* attribute for upper reach is the sum of *ArcLengths* for the upper and lower reach. **Close** the attribute table, and **save** the map document.

Creating River Banks

Bank lines are used to distinguish the main channel from the overbank floodplain areas. Information related to bank locations is used to assign different properties for cross-sections. For example, compared to the main channel, overbank areas are assigned higher values of Manning's n to account for more roughness caused by vegetation. Creating bank lines is similar to creating the channel centerline, but there are no specific guideless with regard to line orientation and connectivity - they can be digitized either along the flow direction or against the flow direction, or may be continuous or broken.

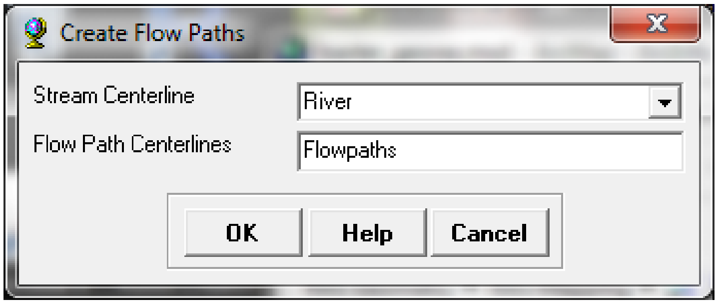
To create the channel centerline (in Banks feature class), follow the same digitization procedure by selecting bank feature and straight segment tool. Although there are no specific guidelines for digitizing banks, to be consistent, follow these guidelines: 1) start from the upstream end; 2) looking downstream, digitize the left bank first and then the right bank. When digitizing the left bank, you do not have to stop at the intersection, you can have a single bank for the whole reach. One the right hand side, however, you cannot cross the creek so you will need two separate lines for the right bank of the Baxter River.

Digitize banks for all three reaches and **save** the edits, stop the editor, and save the map document.

Creating Flowpaths

The flowpath layer contains three types of lines: centerline, left overbank, and right overbank. The flowpath lines are used to determine the downstream reach lengths between cross-sections in the main channel and over bank areas. If the river centerline that we created earlier lie approximately in the center of the main channel (which it does), it can be used as the flow path centerline. **Click** on *RAS Geometry→Create RAS Layers→Flow Path Centerlines.*

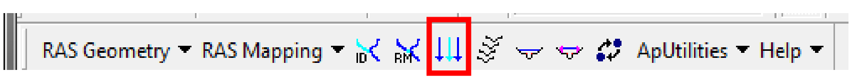
Click *Yes* on the message box that asks if you want to use the stream centerline to create the flow path centerline. Confirm *River* for *Stream Centerline* and *Flowpaths* for *Flow Path Centerlines*, and **click** *OK*.

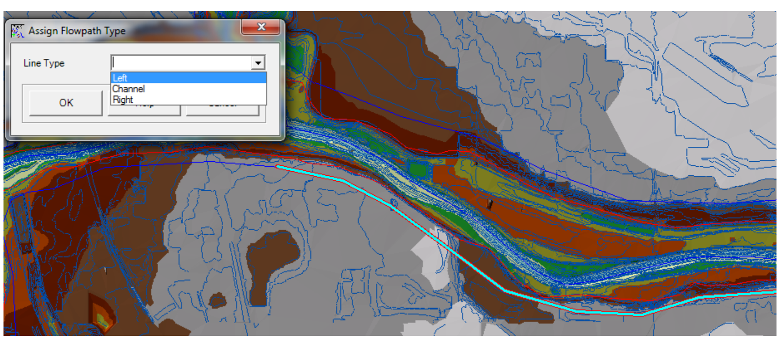


To create the left and right flow paths (in Flowpaths feature class), **start editing** again.

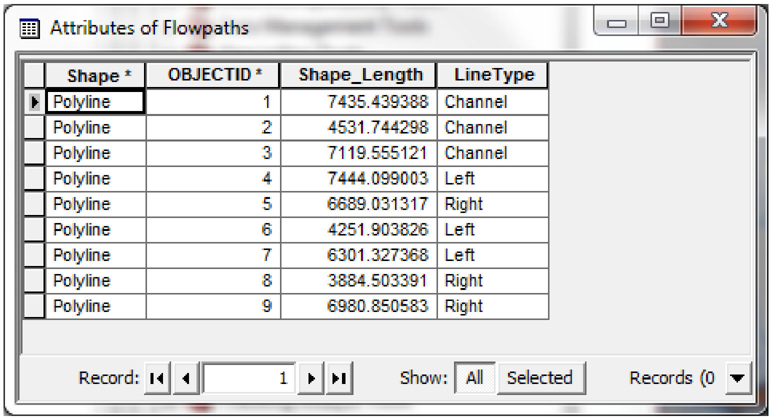
To create the left and right flow paths (in Flowpaths feature class) and **start editing**. Digitize flowpaths using the same digitization procedure as before. The left and right flowpaths must be digitized within the floodplain. These lines are used to compute distances between cross-sections in the over bank areas. Again, to be consistent, looking downstream first digitize the left flowpath followed by the right flowpath for each reach. After digitizing, **save the edits** and **stop editing**.

Now label the flowpaths by using the Assign LineType button . **Click** on the button (notice the change in cursor), and then **click** on one of the flow paths (left or right, looking downstream), and name the flow path accordingly as shown below:





Label all flow paths, and confirm this by opening the attribute table of the Flowpaths feature class. The LineType field must have data for each row if all flowpaths are labeled.



Shape *	OBJECTID *	Shape_Length	LineType
Polyline	1	7435.439388	Channel
Polyline	2	4531.744298	Channel
Polyline	3	7119.555121	Channel
Polyline	4	7444.099003	Left
Polyline	5	6689.031317	Right
Polyline	6	4251.903826	Left
Polyline	7	6301.327368	Left
Polyline	8	3884.503391	Right
Polyline	9	6980.850583	Right

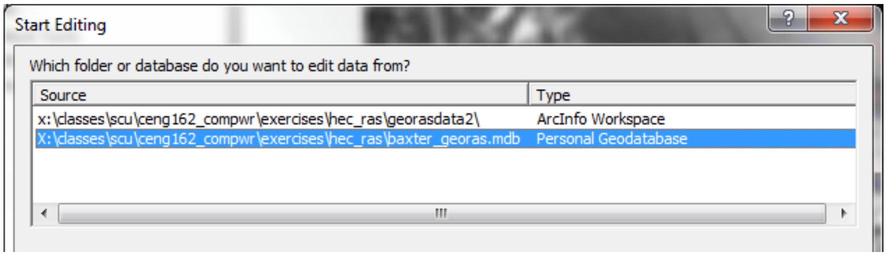
Creating Cross-sections

Cross-sections are one of the key inputs to HEC-RAS. Cross-section cutlines are used to extract the elevation data from the terrain to create a ground profile across channel flow. The intersection of cutlines with other RAS layers such as centerline and flow path lines are used to compute HEC-RAS attributes such as bank stations (locations that separate main channel from the floodplain), downstream reach lengths (distance between cross-sections) and Mannings n. Therefore, creating adequate number of cross-sections to produce a good representation of channel bed and floodplain is critical. Certain guidelines must be followed in creating cross-section cutlines:

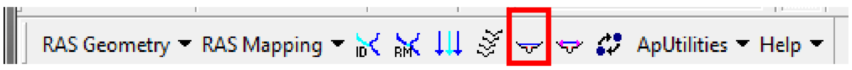
(1) they are digitized perpendicular to the direction of flow;
(2) must span over the entire flood extent to be modeled; and
(3) always digitized from left to right (looking downstream).

Even though it is not required, it is a good practice to maintain a consistent spacing between cross-sections. In addition, if you come across a structure (eg. bridge/culvert) along the channel, make sure you define one cross-section each on the upstream and downstream of this structure. Structures can be identified by using the aerial photograph provided with the tutorial dataset. Click the Add Data button to add the aerial photo layer (named "aerial") to the map. For example, we will use one bridge location in this exercise just downstream of the junction with tributary as shown below (bridge location is shown in red):

To create cross-section cutlines (in XSCutlines feature class), **start editing**. If you receive a window like that below, select the geodatabase that contains the feature XSCutlines.



Digitize cross-section cutlines using the same digitization procedure as before, but follow the guidelines outlined in the previous paragraph. While digitizing, make sure that each cross-section is wide enough to cover the floodplain. This can be done using the cross-sections profile tool.



Click on the profile tool, and then **click** on the cross-section to view the profile. For example, if you get a cross-section profile shown in Figure A below, then there is no need to edit the cross-section, but if you get a cross-section as shown in Figure B below, then the cross-section needs editing. (Note: This tool stops the edit session so you will have to start the edit session every time after viewing the cross-section profile).

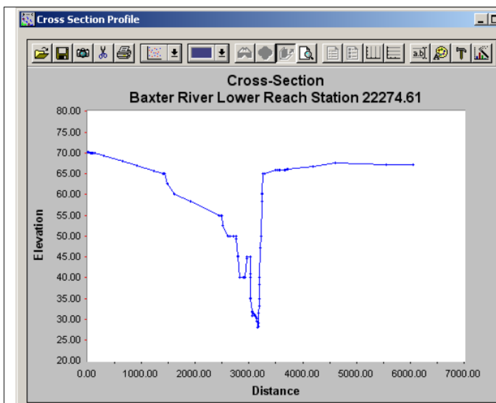


Figure A: adequate definition of cross-section

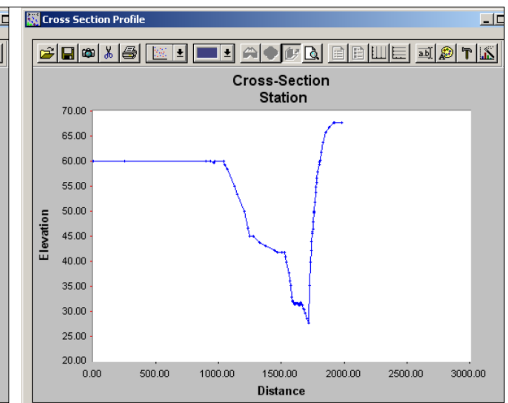
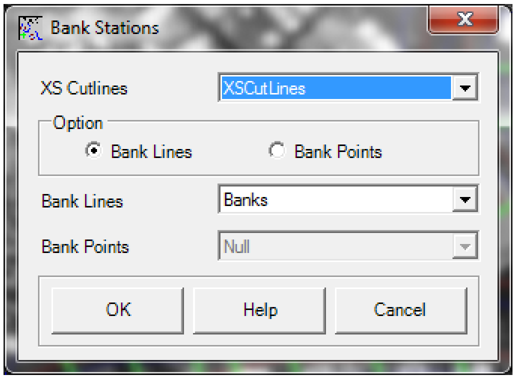


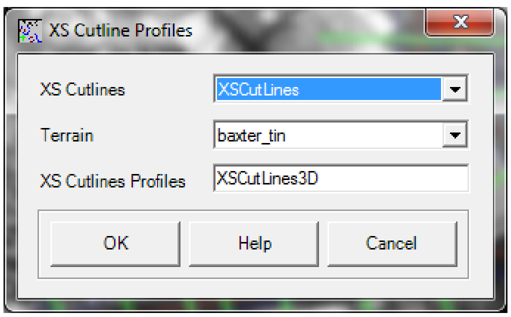
Figure b: inadequate definition of coss-section

After digitizing the cross-sections, **save** the edits and **stop** editing. The next step is to add HEC-RAS attributes to these cutlines. We will add Reach/River name, station number along the centerline, bank stations and downstream reach lengths. Since all these attributes are based on the intersection of cross-sections with other layers, make sure each cross-section intersects with the centerline and overbank flow paths to avoid error messages.

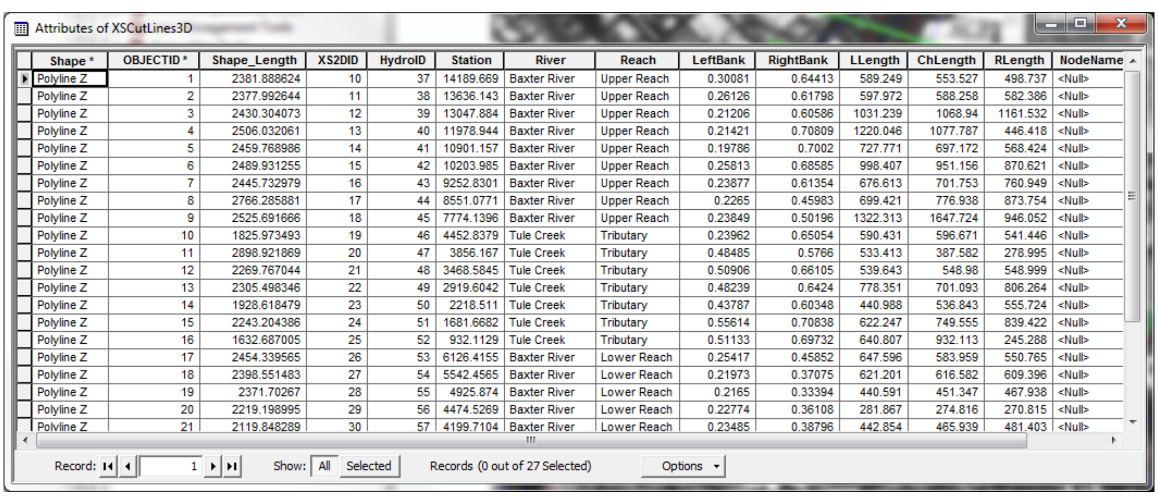
Click on *RAS Geometry→XS Cut Line Attributes→River/Reach Names*. This tool uses the River and Reach attributes of the centerline, and copy them to the XS Cutlines. Next, **click** on *RAS Geometry→XS Cut Line Attributes→Stationing*. This tool will assign station number (distance from each cross-section to the downstream end of the river) to each cross-section cutline. Next, **click** on *RAS Geometry→XS Cut Line Attributes→Bank Stations*. Confirm XSCutlines for XS Cut Lines, and Banks for Bank Lines, and **click** *OK*.



This tool assigns bank stations (distance from the starting point on the XS Cutline to the left and right bank, looking downstream) to each cross-section cutline. Finally, **click** on *RAS Geometry→XS Cut Line Attributes→Downstream Reach Lengths*. This tool assigns distances to the next downstream cross-section based on flow paths.



The cross-section cutlines are 2D lines with no elevation information associated with them (Polyline). When you used the profile tool earlier to view the cross-section profile, the program used the underlying terrain to extract the elevations along the cutline. You can convert 2D cutlines into 3D by **clicking** *RAS Geometry→XS Cut Line Attributes→Elevation*. Confirm *XSCutlines* for *XS Cut Lines*, and *baxter_tin* for *Terrian TIN*. The new 3D lines (XS Cut Lines Profiles) will be stored in the XSCutLines3D feature class. **Click** *OK*.


Attributes of XSCutLines3D

Shape *	OBJECTID *	Shape_Length	XS2DID	HydroID	Station	River	Reach	LeftBank	RightBank	LLength	ChLength	RLength	NodeName
Polyline Z	1	2381.888624	10	37	14189.669	Baxter River	Upper Reach	0.30081	0.64413	589.249	553.527	498.737	<Null>
Polyline Z	2	2377.992644	11	38	13636.143	Baxter River	Upper Reach	0.26126	0.61798	597.972	588.258	582.386	<Null>
Polyline Z	3	2430.304073	12	39	13047.884	Baxter River	Upper Reach	0.21206	0.60586	1031.239	1068.94	1161.532	<Null>
Polyline Z	4	2506.032061	13	40	11978.944	Baxter River	Upper Reach	0.21421	0.70809	1220.046	1077.787	446.418	<Null>
Polyline Z	5	2459.768986	14	41	10901.157	Baxter River	Upper Reach	0.19786	0.7002	727.771	697.172	568.424	<Null>
Polyline Z	6	2489.931255	15	42	10203.985	Baxter River	Upper Reach	0.25813	0.68585	998.407	951.156	870.621	<Null>
Polyline Z	7	2445.732979	16	43	9252.8301	Baxter River	Upper Reach	0.23877	0.61354	676.613	701.753	760.949	<Null>
Polyline Z	8	2766.285881	17	44	8551.0771	Baxter River	Upper Reach	0.2265	0.45983	699.421	776.938	873.754	<Null>
Polyline Z	9	2525.691666	18	45	7774.1396	Baxter River	Upper Reach	0.23849	0.50196	1322.313	1647.724	946.052	<Null>
Polyline Z	10	1825.973493	19	46	4452.8379	Tule Creek	Tributary	0.23962	0.65054	590.431	596.671	541.446	<Null>
Polyline Z	11	2898.921869	20	47	3856.167	Tule Creek	Tributary	0.48485	0.5766	533.413	387.582	278.995	<Null>
Polyline Z	12	2269.767044	21	48	3468.5845	Tule Creek	Tributary	0.50906	0.66105	539.643	548.98	548.999	<Null>
Polyline Z	13	2305.498346	22	49	2919.6042	Tule Creek	Tributary	0.48239	0.6424	778.351	701.093	806.264	<Null>
Polyline Z	14	1928.618479	23	50	2218.511	Tule Creek	Tributary	0.43787	0.60348	440.988	536.843	555.724	<Null>
Polyline Z	15	2243.204386	24	51	1681.6682	Tule Creek	Tributary	0.55614	0.70838	622.247	749.555	839.422	<Null>
Polyline Z	16	1632.687005	25	52	932.1129	Tule Creek	Tributary	0.51133	0.69732	640.807	932.113	245.288	<Null>
Polyline Z	17	2454.339565	26	53	6126.4155	Baxter River	Lower Reach	0.25417	0.45852	647.596	583.959	550.765	<Null>
Polyline Z	18	2398.551483	27	54	5542.4565	Baxter River	Lower Reach	0.21973	0.37075	621.201	616.582	609.396	<Null>
Polyline Z	19	2371.70267	28	55	4925.874	Baxter River	Lower Reach	0.2165	0.33394	440.591	451.347	467.938	<Null>
Polyline Z	20	2219.198995	29	56	4474.5269	Baxter River	Lower Reach	0.22774	0.36108	281.867	274.816	270.815	<Null>
Polyline Z	21	2119.848289	30	57	4199.7104	Baxter River	Lower Reach	0.23485	0.38796	442.854	465.939	481.403	<Null>

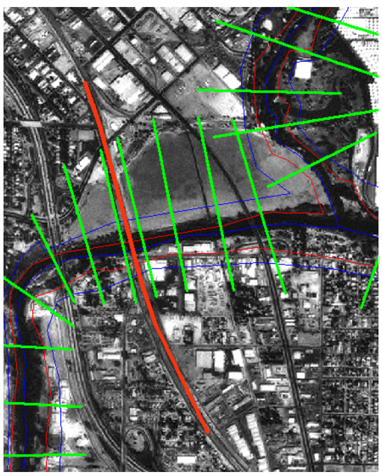
Record: 1 Show: All Selected Records (0 out of 27 Selected) Options

After this process is finished, open the attribute table of XSCutLines3D feature class and see that the shape of this feature class is now PolylineZ.

Creating Bridges and Culverts

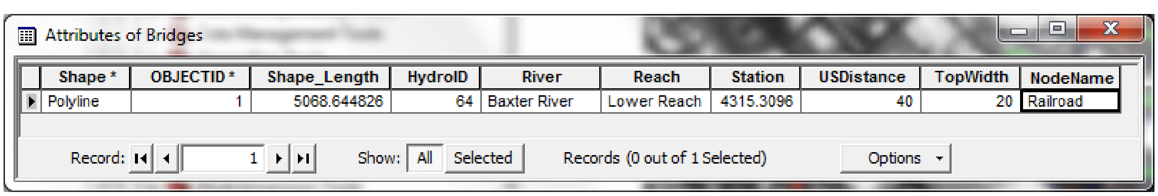
After creating cross-sections, the next step is to define bridges, culverts and other structure along the river. Since we used aerial photograph while defining the cross-sections, our job of locating the bridge is done. To create bridges/culverts (in Bridges feature class), **start editing.** A bridge or culvert is treated similar to a cross-section so the same criteria used for creating cross-sections must be used for bridge/culverts.

Using Bridge as the edit feature and the straight segment tool on the editor toolbar, **digitize** bridge location just downstream of the tributary junction. While digitizing the bridge, make use of the terrain model to make sure the bridge/road centerline fall on the high ground. **Save** your edits and **stop** editing.



After digitizing bridges/culverts, you need to assign attributes such as River/Reach name and station number to these features. **Click** on *RAS Geometry→Bridge/Culverts→River/Reach Names* to assign river/reach names. Next **click** on *RAS Geometry→Bridge/Culverts→Stationing* to assign station numbers.

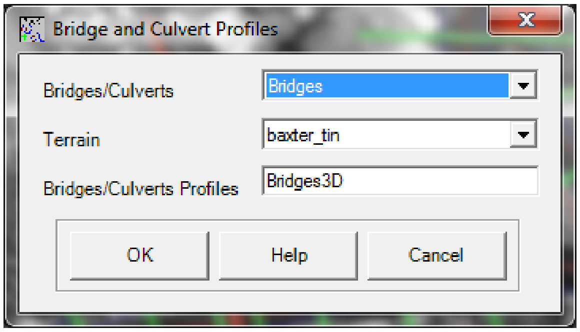
Besides these attributes, you must enter additional information about the bridge(s) such as the name and width in its attribute table as shown below. Start editing, and add the required information.



Shape *	OBJECTID *	Shape_Length	HydroID	River	Reach	Station	USDistance	TopWidth	NodeName
Polyline	1	5068.644826	64	Baxter River	Lower Reach	4315.3096	40	20	Railroad

Close the attribute table, **save** edits and **stop** editing.

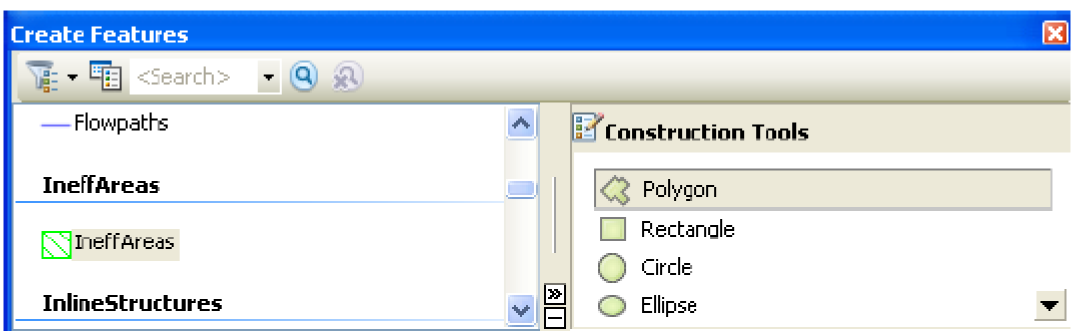
Similar to cross-sections, the Bridges feature class stores 2D polylines, you can make them 3D by **clicking** *RAS Geometry*→*Bridge/Culverts*→*Elevations* to create a new 3DBridges feature class. Confirm *Bridges* for *Bridges/Culverts*, *baxter_tin* for *Terrain*, *Bridges3D* for *Bridges/Culverts Profiles*, and **Click** *OK*.



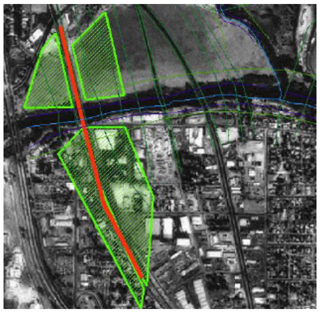
A new feature class (Bridges3D) will be created. You can check it is PolylineZ by opening its attribute table.

Creating ineffective flow areas

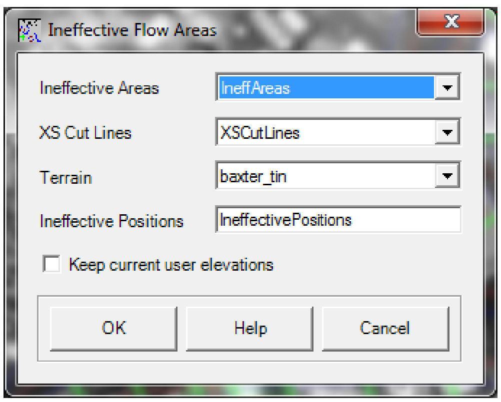
Ineffective flow areas are used to identify non-conveyance areas (areas with water but no flow/zero velocity) of the floodplain. For example, areas behind bridge abutments representing contraction and expansion zones can be considered as ineffective flow areas. To define ineffective areas (in IneffAreas feature class), **start editing**, and choose IneffAreas as the edit feature, and Polygon as the construction tool.



Use the sketch tool to define ineffective areas. The figure below shows an example of ineffective area for the bridge downstream of the tributary junction (Note: this is a polygon feature class).



HEC-RAS does not store all information about ineffective areas. Instead only the information where the ineffective area may interfere with cross-sections/flow is stored. To extract the position and elevation at points where these ineffective areas intersect with cross-sections, **click** on *RAS Geometry→Ineffective Flow Areas→Position*. Leave the default feature classes for *IneffectiveAreas*, *XS Cut Lines*, and *Terrain* unchanged. The position of ineffective areas will be stored in a new table named *IneffectivePositions*. Leave current user elevations **unchecked**, and **Click** *OK*.



Open the *IneffectivePositions* table (shown below; it may need to be added to your project, and if so you'll find it in the baxter_georas geodatabase) to understand how this information is stored.

Attributes of IneffectivePositions

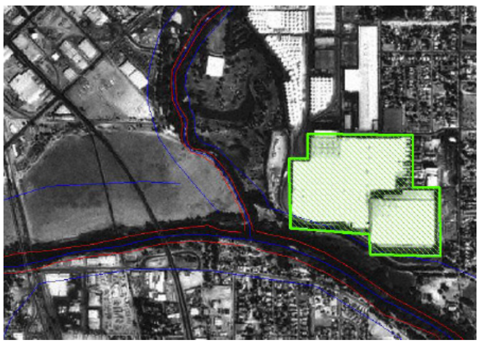
OBJECTID *	XS2DID	IA2DID	BegFrac	EndFrac	BegElev	EndElev	UserElev
13	28	66	0.3991	1	59.535	89.84	<Null>
14	29	66	0.78811	0.99948	63.84	87.84	<Null>
15	30	67	0.46365	0.50337	55.84	55.84	<Null>
16	31	67	0.47859	0.63238	54.611	55.84	<Null>
17	29	68	0	0.24297	72.489	66.022	<Null>
18	30	68	0	0.27142	72.963	54.12	<Null>

Record: 1 Show: All Selected Records (0 out of 6 Selected) Options

IA2DID is the HydroID of the ineffective flow area, *XS2DID* is the HydroID of the intersecting cross-section, *BeginFrac* and *EndFrac* are the relative positions of the first and last intersecting points (looking downstream) of the ineffective area with the cross-section. *BegElev* and *EndElev* are the elevations of the first and last intersecting points of the ineffective area with the cross-section. Since you left the *UserElev* box unchecked there are no values in this field.

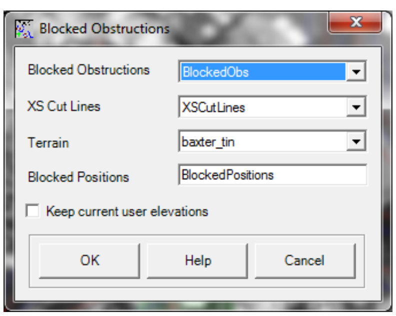
Creating obstructions

Obstructions represent blocked flow areas (areas with no water and no flow). For example, buildings in the floodplain and levees are considered obstructions. We can add blocked obstructions to our study by using building locations in the aerial photograph. In the upper reach of the Baxter River just before Tule Creek, there are two building in the floodplain that can be considered as blocked obstructions.



To define blocked obstructions (in BlockedObs feature class), **start editing**, and choose BlockedObs as the edit feature and Polygon as the construction tool.

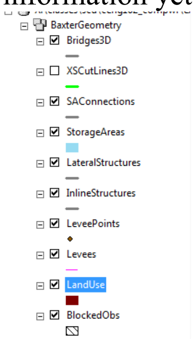
Using the straight segment tool, digitize the blocked obstruction polygon, **save edits** and **stop editing**. Similar to Ineffective flow areas, the positions and elevations of the intersection of this obstruction with cross-sections needs to be stored in a table. Click on *RAS Geometry→Blocked Obstructions→Positions*.



Leave the default values in the *Blocked Obstructions* window, and **click** *OK*. You will notice that a new table (*BlockedPositions*) will be added to the map document, and its content are identical to *IneffectivePositions* table.

Assigning Manning's n to cross-sections

The final task before exporting the GIS data to HEC-RAS geometry file is assigning Manning's n value to individual cross-sections. In HEC-GeoRAS, this is accomplished by using a land use feature class with Manning's n stored for different land use types. Ideally you will store this information in the LandUse feature class added to the map document. Since we created empty feature classes at the beginning of the tutorial, we do not have this information yet. If you open this you will find an empty table:



Remove the empty LandUse feature class from the map, and add LandUse shapefile (stored in LandUse folder) provided with the tutorial dataset. (Note: you can also replace the LandUse feature class in Baxter_georas.mdb with the shapefile in ArcCatalog). the attribute table for the LandUse shapefile looks like this:

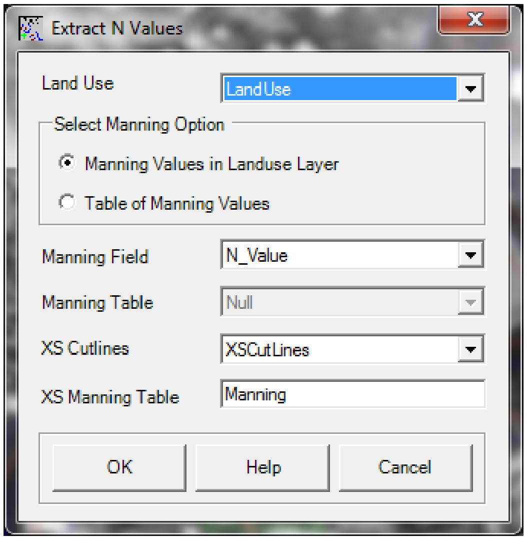
Attributes of LandUse

FID	Shape *	LUCode	N_Value	Shape_Leng	Shape_Area
0	Polygon	Nearstream	0.035	74238.039962	47884074.2645
1	Polygon	Nearstream	0.035	94595.836838	77091525.3525
2	Polygon	Nearstream	0.035	19499.698779	3890744.91706
3	Polygon	Urban	0.05	25155.899476	37360769.6827
4	Polygon	Farming	0.06	57470.779847	132969468.043
5	Polygon	Urban	0.05	43729.482249	80210914.7352
6	Polygon	Urban	0.05	49495.579306	131147302.076
7	Polygon	Farming	0.06	98679.544048	569365177.1
8	Polygon	Farming	0.06	139480.817374	582945502.797
9	Polygon	Urban	0.05	77954.853646	194373174.064
10	Polygon	Farming	0.06	74511.026072	88342217.8171
11	Polygon	Farming	0.06	88235.010374	388565989.4
12	Polygon	Urban	0.05	30673.048295	44559471.0035
13	Polygon	Urban	0.05	72745.701764	105386332.035
14	Polygon	Nearstream	0.035	34810.970843	18744922.7034
15	Polygon	Farming	0.06	57903.275167	128069785.89
16	Polygon	Urban	0.05	13425.045625	3597365.82156
17	Polygon	HDResidential	0.08	23345.712486	21645098.4585
18	Polygon	HDResidential	0.08	13269.064964	6767179.53546
19	Polygon	Industrial	0.1	6281.239796	1721705.16172
20	Polygon	Orchards	0.055	51184.291415	50989001.7624
21	Polygon	Orchards	0.055	24349.432937	16911348.2337
22	Polygon	OpenSpace	0.04	6223.68837	2388434.68023
23	Polygon	Crop/Pasture	0.05	34706.032189	41444364.6289
24	Polygon	Crop/Pasture	0.05	11101.650671	7681981.48856
25	Polygon	Commercial	0.12	9041.893299	4079362.04699
26	Polygon	HDResidential	0.08	2904.766995	328375.183161
27	Polygon	School	0.055	2983.818755	542738.171626
28	Polygon	OpenSpace	0.04	3777.952701	739209.523499
29	Polygon	Industrial	0.1	41986.054538	54852278.3281
30	Polygon	Floodplain	0.04	5442.322348	1470514.88886
31	Polygon	Industrial	0.1	14341.78707	7408749.11325
32	Polygon	HDResidential	0.08	17746.993539	11911370.749

Record: 1 Show: All Selected Records (0 out of 90 Selected) Options

The land use table must have a descriptive field identifying landuse type, which is *LUCode* in this case, and a field for corresponding Manning's n values. In addition, HEC-GeoRAS requires the land use polygons to be non multi-part features (a multipart feature has multiple geometries in the same feature). The issue of non multi-part features is taken care of for the tutorial dataset.

To assign Manning's n to cross-sections, **click** on *RAS Geometry*→*Manning's n Values*→*Extract n Values*. Confirm *LandUse* for *Land Use*, choose *N_Value* for *Manning Field*, *XSCutLines* for *XS Cut Lines*, leave the default name *Manning* for *XS Manning Table*, and **click** *OK*. (Note: Summary Manning Table is not required if n values already exist in the LandUse table.).

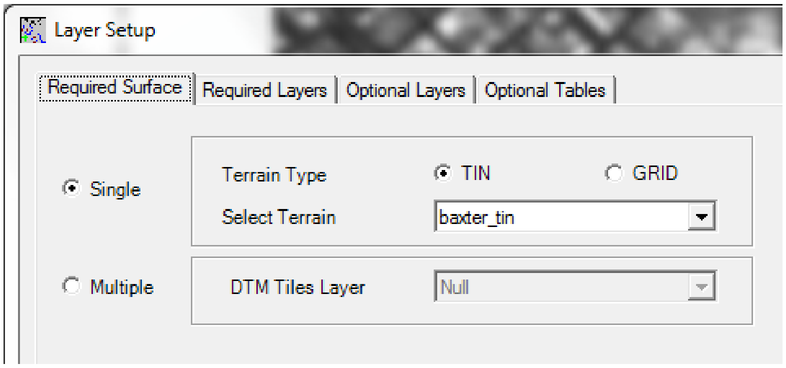


Add the table to the ArcMap document, if it isn't already added (it should be there). Depending on the intersection of cross-sections with landuse polygons, Manning's n are extracted for each cross-section, and reported in the XS Manning Table (*Manning*). **Open** the Manning table, and see how the values are stored. Similar to previous tables, the data are organized as the feature identifier (*XS2DID*), its relative station number and the corresponding n value as shown below:

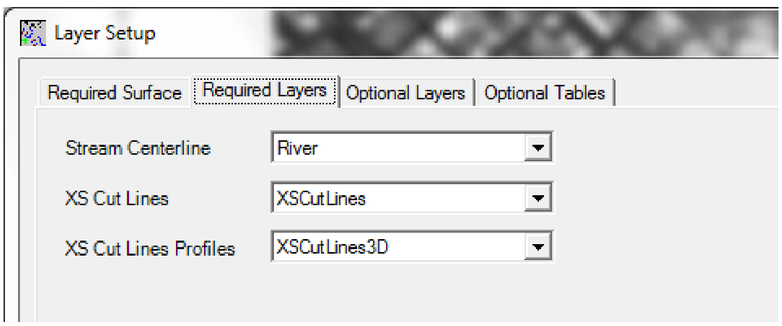
Attributes of Manning

OBJECTID *	XS2DID	Fraction	N_Value
1	10	0	0.06
2	10	0.28586	0.035
3	11	0	0.06
4	11	0.22391	0.035
5	12	0	0.06
6	12	0.18684	0.035
7	13	0	0.06
8	13	0.20129	0.035
9	13	0.68474	0.05
10	14	0	0.06
11	14	0.23858	0.035
12	14	0.66345	0.05
13	15	0	0.06
14	15	0.27991	0.035
15	15	0.5818	0.05
16	16	0	0.06
17	16	0.25322	0.035

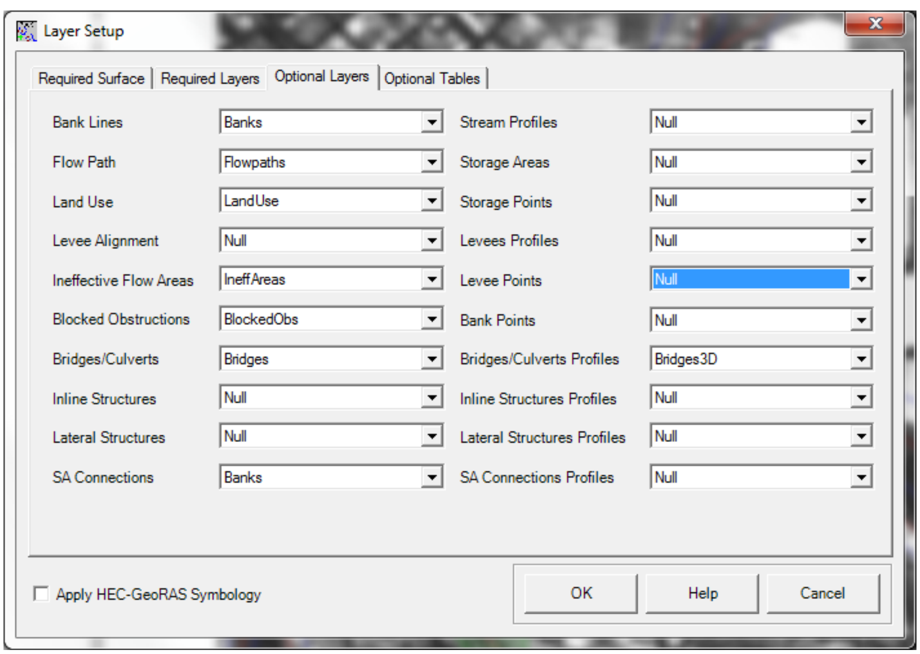
Close the table. We are almost done with GeoRAS pre-processing. The last step is to create a GIS import file for HEC-RAS so that it can import the GIS data to create the geometry file. Before creating an import file, make sure we are exporting the right layers. **Click** on *RAS Geometry→Layer Setup*, and verify the layers in each tab. The required surface tab should have *baxter_tin* for *single Terrain* option.



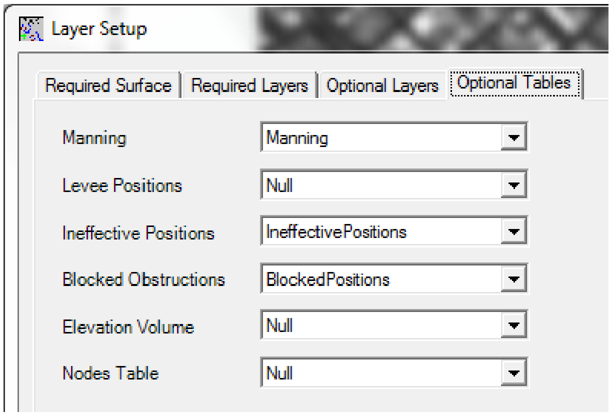
The *Required Layers* tab should have *River*, *XSCutLines* and *XSCutLines3D* for *Stream Centerline*, *XSCutmLines* and *XSCut Lines Profiles*, respectively.



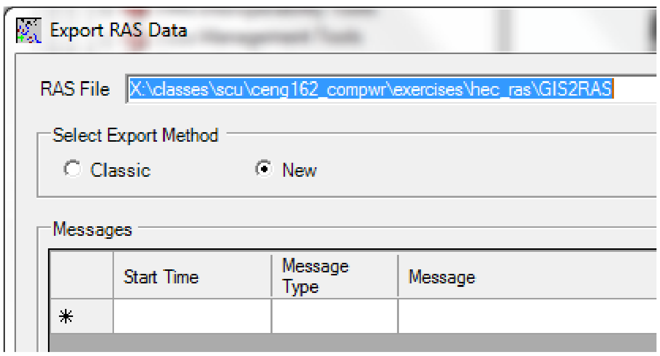
In the *Optional Layers* tab, make sure the layers that are empty are set to *Null*.



Finally, verify the tables and **Click** *OK*.



After verifying all layers and tables, **click** on *RAS Geometry→Export RAS Data*.



Confirm the location and the name of the export file (*GIS2RAS* in this case), and **click** *OK*. This process will create two files: *GIS2RAS.xml* and *GIS2RAS.RASImport.sdf*. **Click** *OK* on the series of messages about computing times. You are done exporting the GIS data! The next step is to import these data into a HEC-RAS model.

Save the map document. You can either close the ArcMap session or leave it running.

<u>To Turn In:</u>

Create an ArcMap layout of your created HEC-RAS model. This should show (at least):

- cross sections
- river banks
- flow paths
- bridge
- ineffective flow areas
- obstructions

A legend should describe these, and other appropriate annotation should be shown on the map. Do not include extraneous information. A north arrow and scale bar are necessary for any map.

Provide a diagram of a cross-section, and identify clearly where it is on the map.