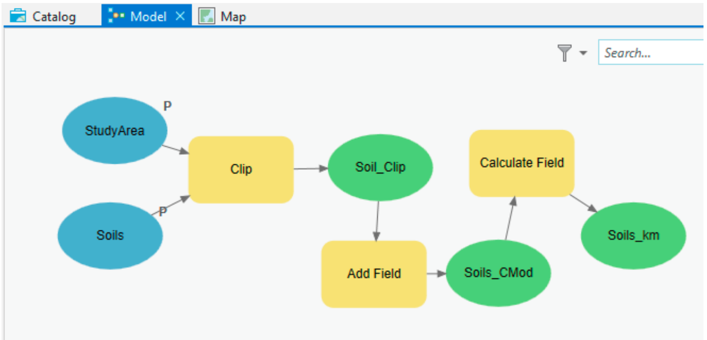
Model Builder – Automate Processing

Model Builder is an ArcGIS application that you can use to create processing workflows and tools. You can create something like a flowchart on a Model Builder canvas, with connected data and operations that result in some output. The figure at right shows an example model that clips soils data by a study area, and then adds and calculates a field.

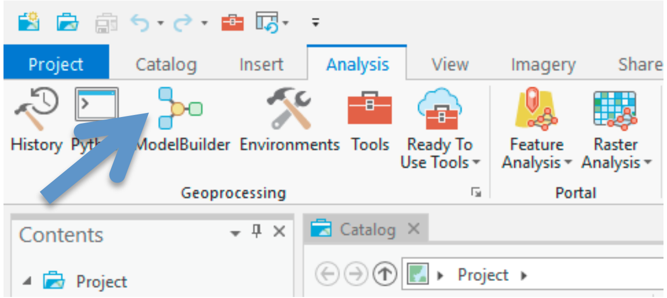


Model Builder can be used to automate spatial processing, create custom tools for oft-applied steps, and provide a record of the steps in an analysis. It can string together long-winded processes so that you may concentrate on other tasks while your spatial computations procede.

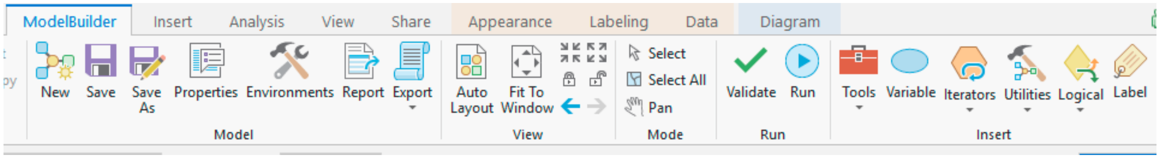
You typically first do an analysis manually, applying a set of spatial operations to your input data sets while puzzling out a sequence that yields your target information, often on a small piece of your study area. Large data sets may cause long long run times, slowing down your prototyping. You may manually subset your data to create smaller, more manageable data sets, and apply a series of preliminary analysis, debugging your workflow. You review intermediate and final results of this manual analysis for accuracy, and verify that your processing stream answers the target spatial question. If not, you alter the processes and steps until you have a correct analysis. You then replicate the sequence of steps in a spatial model in Model Builder.

This exercise introduces Model Builder, and "active" flowchart the applies geoprocessing workflows. Base data that I used for the example below are on the L: drive, in the main course directory, in the Data\ModelBuilder subdirectory. There is a geodatabase named ExampModBuildData.gdb with a study boundary and a soils data layer. You may load these into a new project, and follow along with the example below. But note you do not need to turn in this example model. As described at the end of this document, your assignment is to turn a model that creates watersheds, given a DEM and a set of pourpoints.

You start model builder through the Analysis tab > ModelBuilder icon (right figure):



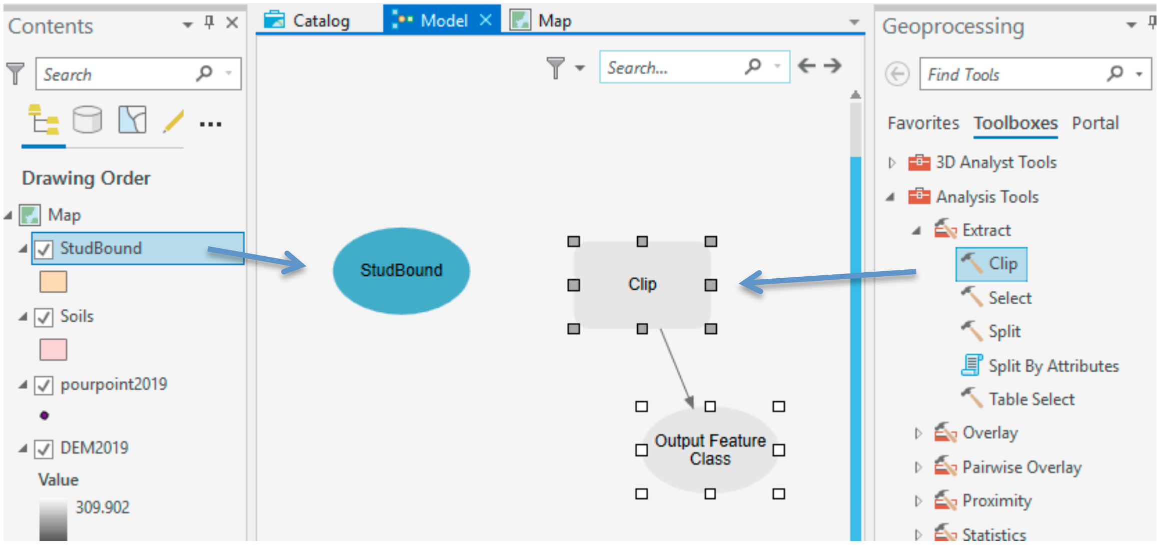
This opens a Model Builder tab, which displays a number of tools (figure below):



Below the tool ribbon is a new window pane for building models.

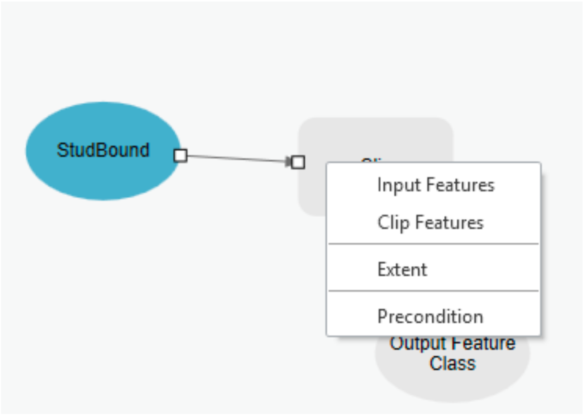
You typically build a model by dragging and dropping input data from the Table of Contents on the left of your ArcGIS Pro display, and geoprocessing tools from the right of the display. You connect data and tools by clicking and dragging arrows from "bubble to bubble" to indicate the processing flow. You may introduce additional variables, and set layers or scalars as parameters for your tool. You can label and rename layers or tools.

In the figure below, I click and held on the StudBound layer, and drug it over to the Model window. I did the same with the Clip tool, click-dragging from the right geoprocessing tab into the model window.



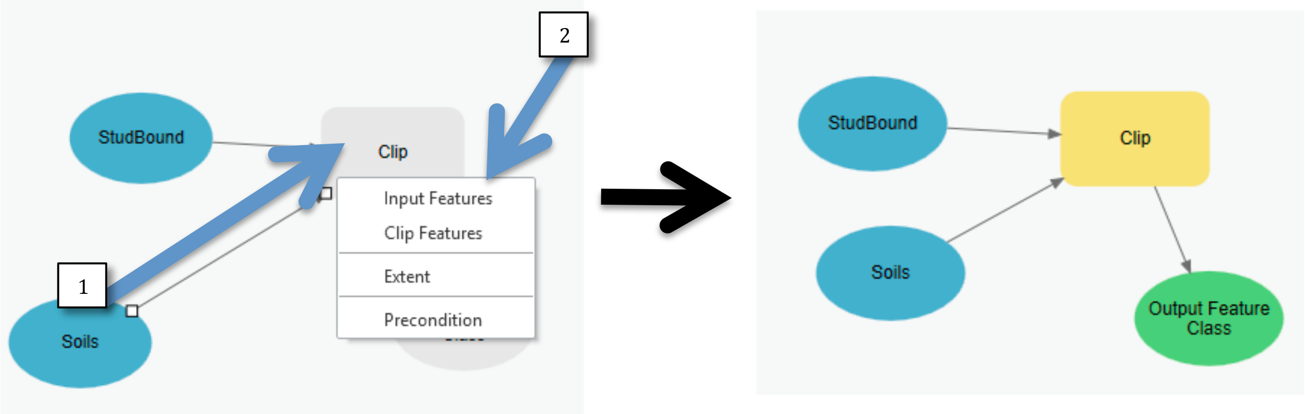
Note that the data oval labeled StudBound is colored, meaning its parameters are all set, and the Clip tool is not colored, as I haven't defined the parameters. A grayed bubble indicates something is needed.

Once I've added a spatial operation, I need to specify the inputs and parameters, here for the Project command. I do this by click-holding on the data layer, and dragging from the input data layer to the Clip bubble:



This should create an arrow connecting the data to the Clip tool, and open a table listing the available inputs. I select "Clip Features" in the popup table, as I want this to be my Clip Features layer, used to clip my target (Input) features.

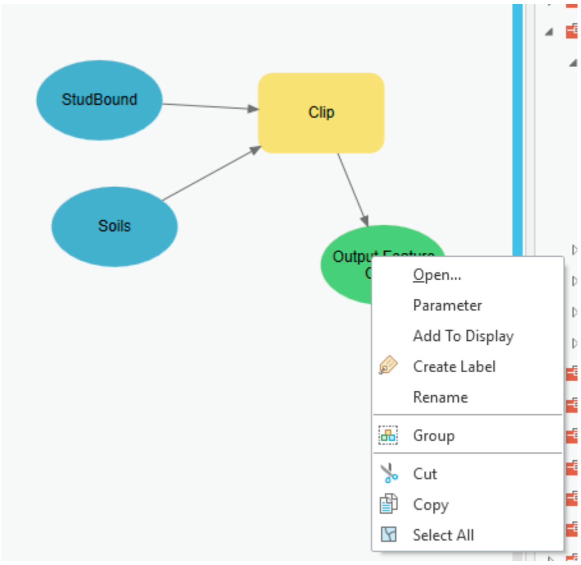
I'll set the input features by dragging a Soils data layer on to the canvas, and then click-dragging (1) a line from the Soils layer to the Clip tool, opening a table as shown in the figure, below left. If I specify the Soils layer as the Input features in the popup table by clicking on the Input Features label (2), the Clip and Output Feature Class bubbles should change color (below right):



We often wish to rename output data layers, create labels, or cut, copy, or paste it, although copy and paste are most often used for geoprocessing tools. Right clicking on a tool or data set allows you to set properties, e.g., for the output feature class at right

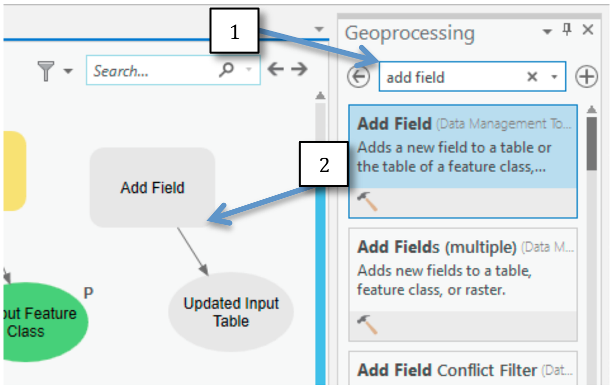
Clicking on Parameter in the popup table will make the data set a parameter of each model run. This means the user will have to input values. We often specify that input and output data sets as parameters, because we wish to use the tools on different data, or save the results to different files.

When we make something a parameter, a P will be displayed next to the bubble.



We can also drag a tool from a Geoprocessing search result into the model. Here I searched for the Add Field command (1, at right); by a click-hold-drag (2) I can pull the tool on to the model canvas, and then connect it to one of the processing streams.

You should drag the Add Field tool on to your canvas, and connect from the Output Feature Class from the Clip tool to the Add Field Tool.

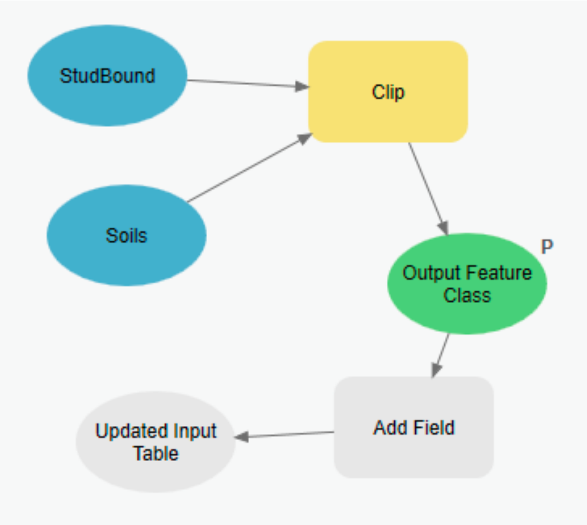


Add Field (Data Management To...
Adds a new field to a table or the table of a feature class,...

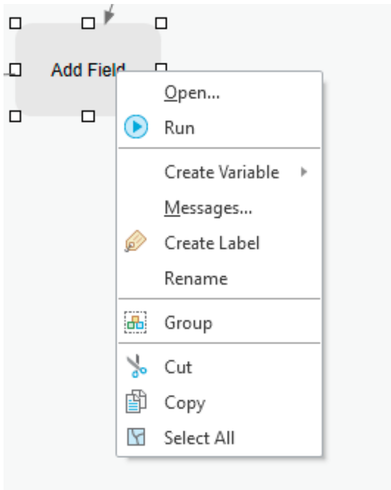
Add Fields (multiple) (Data M...
Adds new fields to a table, feature class, or raster.

Add Field Conflict Filter (Dat...

Note that it stays grey in color, unlike the previous times you connected tools to data. This indicates you need to specify additional inputs or parameters for the tool to function.



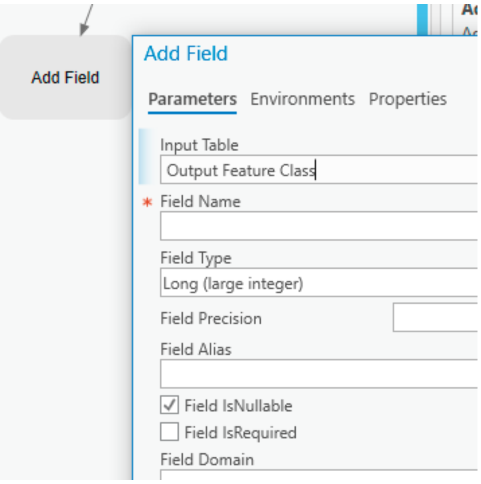
Right click on the Add Field bubble, to display a drop-down list of choices:



Select the ***Open…*** option at the top of the list, note it displays a set of parameters. The Field Name must be specified, indicated by the red asterisk next to the parameter.

Change the field name to AreaAcres (or something similar).

Set the type to float or double, the precision to 12, and the scale to 1. You can leave the other variables at their defaults.



Input Table: Output Feature Class

* Field Name

Field Type: Long (large integer)

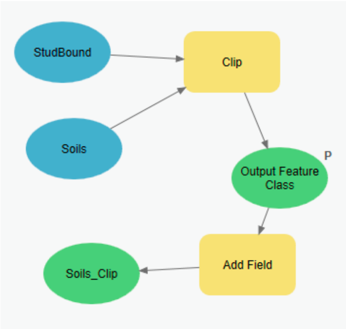
Field Precision

Field Alias

☑ Field IsNullable

□ Field IsRequired

Field Domain

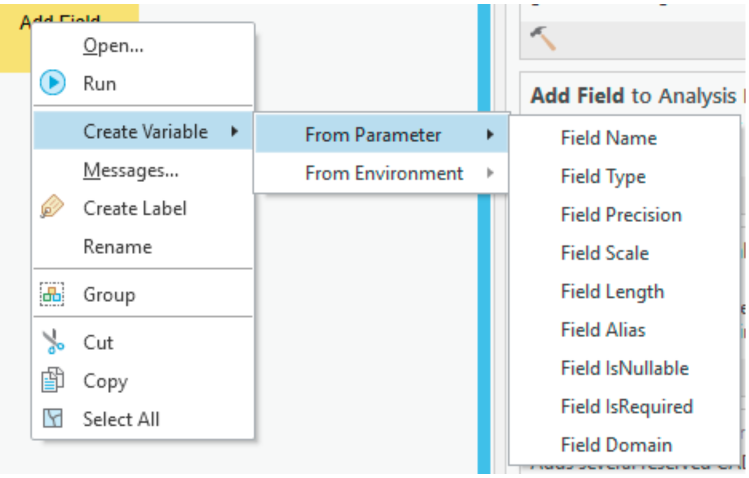


The bubbles for the tool and output data switch from gray to colored after making these changes (thumbnail view at left).

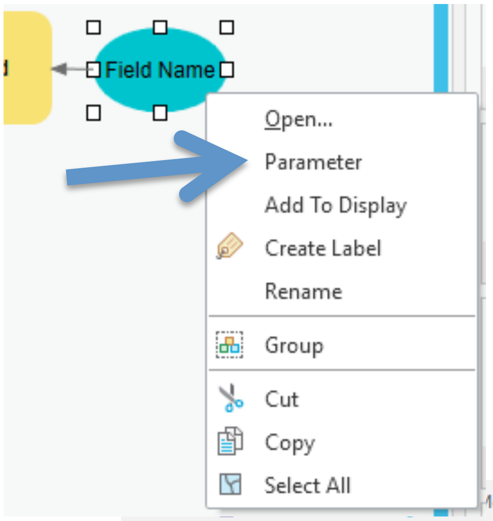
We can also make the parameters into variables that will be entered when running the tool, for example, to allow the user to input a field name.

Right click on the Add Field bubble, and select ***Create Variable*** from the dropdown list:

Note you can make all the entered parameters variables. Here just make the ***Field Name*** a variable.

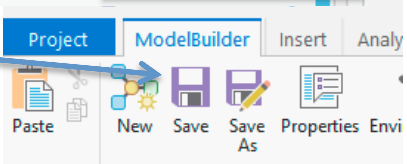


You should get a new bubble labeled field name, and color filled. Right click on it, and make it a ***Parameter***:

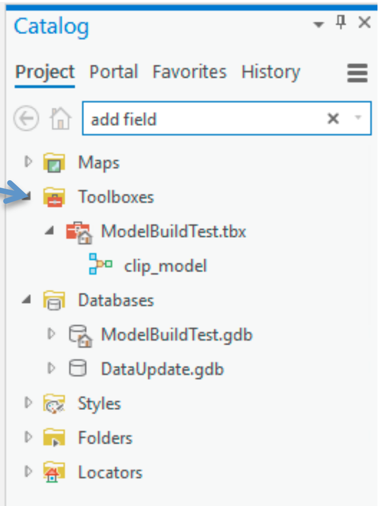


After you make the Field Name a parameter, display the dropdown table again and select ***Open*** to view the default value for the parameter. It should display a window with the name, e.g., AreaAcres.

Let's save our model. Click on the ***Save*** icon in the ModelBuilder ribbon near the top of the ArcGIS main window:



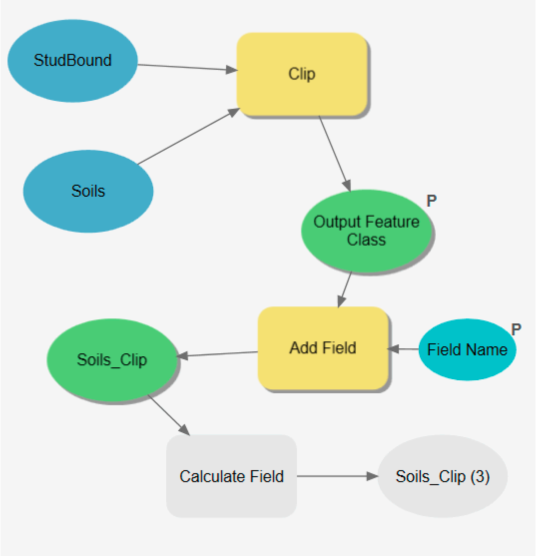
The default location is in the Toolboxes folder associated with the current project. You can also see the Toolboxes folder in the Catalog window. Create a new toolbox, here named ModelBuildTest, and save the model as a new tool, here clip_model.



The model should appear in the Catalog view for the project, under the Toolboxes list.

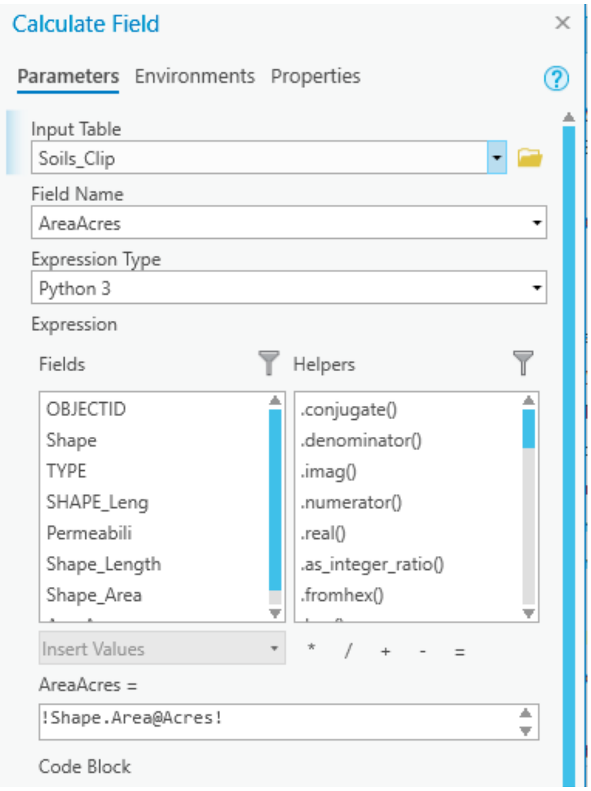
You can save to other areas besides your toolbox in a project, we'll cover that later.

Let's add one more operation to our model. Locate the ***Calculate Field*** command, drag it to the canvas, and set it to use as input the Soils_Clip data from our Add Field command. This should yield a canvas similar to the figure to the right.



Select and Open the Calculate Field bubble, and modify the inputs to match the figure at right. The input table should be filled correctly, but you should specify the:

- target field (here AreaAcres, the new field created earlier), and
- the Expression that defines the value to be assigned to AreaAcres, here the odd looking !Shape.Area@Acres! near the bottom of the figure. This is the syntax needed to access pre-defined ArcGIS code that calculates the area of polygons in a layer.



Calculate Field

Parameters Environments Properties

Input Table: Soils_Clip

Field Name: AreaAcres

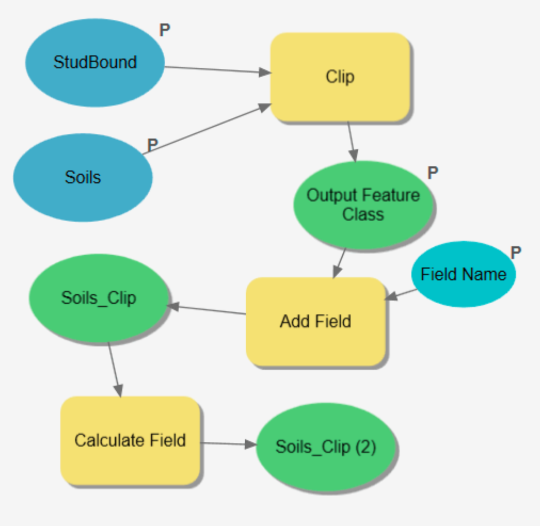
Expression Type: Python 3

Expression

AreaAcres =

```
!Shape.Area@Acres!
```

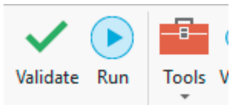
Code Block



The bubbles should all be colored, somewhat like the figure at left. Make the input StudBound and Soils layers parameters, so that P's appear next to them.

Save the model again.

Once you've built your complete model, the next step is typically to Validate the Model, via an icon found in the main ModelBuilder ribbon:



If the model is successfully validated, all the bubbles stay colored. If anything fails, the offending bubbles switch to gray. You typically open the first in the sequence of failed bubbles, review parameters and outputs, modify, and validate again. You work down the sequence of offending bubbles until they all remain colored during a validation.

The next step is to run the model, by clicking on the Run icon, next to the Validate icon, in the figure above.

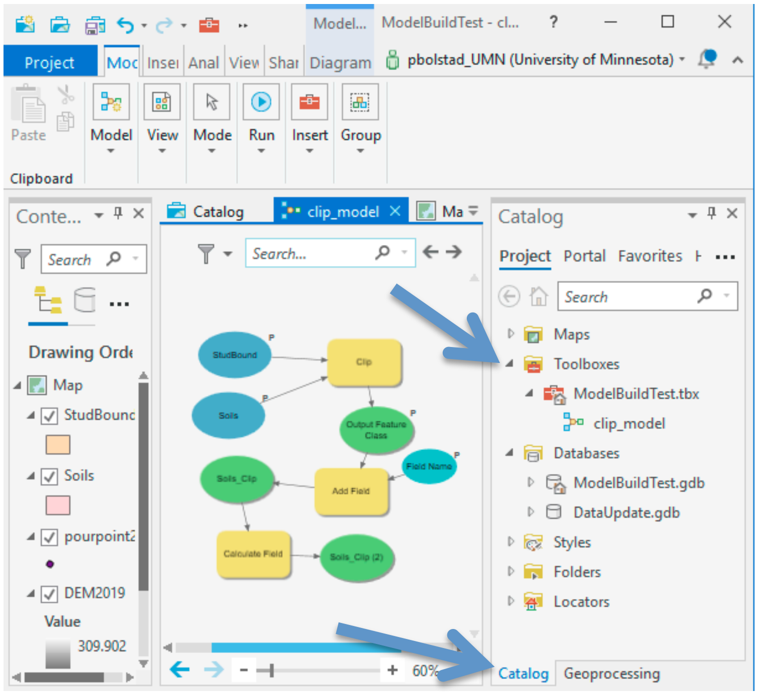
You should see new window pop up as Model Builder step through the elements of the model, with a progress message displayed in a bottom pane of this new window. When the model completes, you will see a Succeeded message if all ran well.

That's it. If you've done an example like this with a test dataset of your own, you've run your (first perhaps?) Model Builder model.

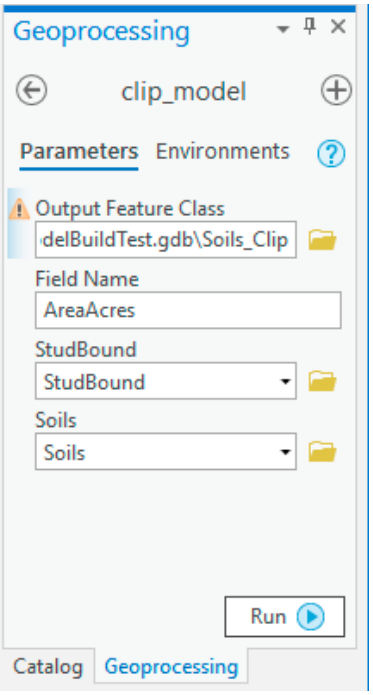
To the run the model outside of the Model Builder window, activate the Catalog tab, then expand the default Toolboxes – {defaulttoolbox}.tbx

The name {defaulttoolbox} will change to match your project name, in this case ModelBuildTest.tbx.

Here I named the model clip_model on save, and it is the only model in the toolbox.



Double clicking on clip_model starts a run, with parameters listed, as shown at right:



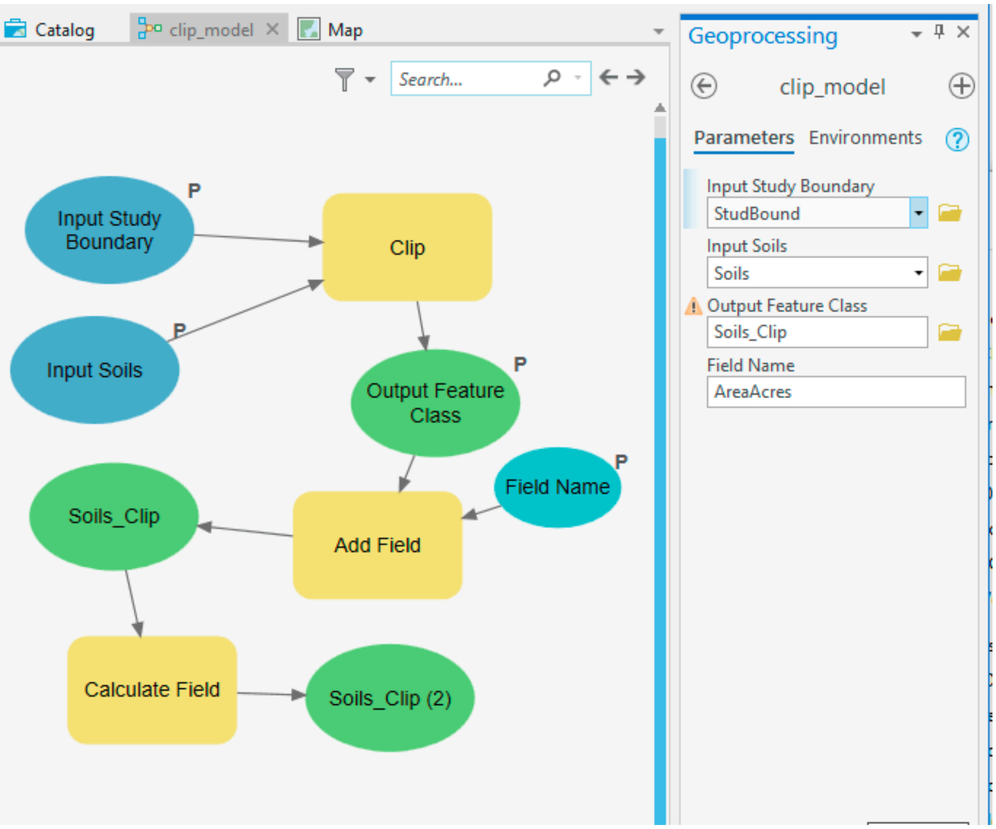
Output Feature Class: delBuildTest.gdb\Soils_Clip
Field Name: AreaAcres
StudBound: StudBound
Soils: Soils

Note that the parameters are listed in the order specified during the model build. Since I didn't specify the input data as parameters until the end, the StudBound and Soils inputs are listed last, rather than first. Also note that the names aren't very informative – they should specify that these are input feature classes.

I can modify the order of parameters by going back to the model in the ModelBuilder Canvas, right clicking on the bubbles, and then turning off all the parameter flags.

I then cycle through my parameters, turning them on in the preferred display order.

You can also rename the data sets by right clicking on the bubbles, and specifying rename from the dropdown table. After changing the names and saving the model, running it via the Catalog/project Toolbox appears as to the right.



Input Study Boundary: StudBound
Input Soils: Soils
Output Feature Class: Soils_Clip
Field Name: AreaAcres

You often create a set of models associated with a project, and keep them in the project with the data on which they operate. You can modify the data, and then re-run the model.

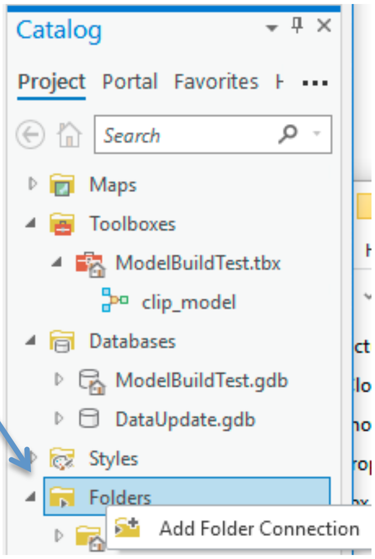
As an alternative, you can use the project as a template, and copy the project with the models and data, and if you wish to work with new data or in a new area, delete and add data as needed. You have to be careful when using this template approach that all the data are either set as parameters, or have the same name and structure as the original data and are included in the new project.

Another alternative is to save a model to a folder, and then add a folder connection to future projects. We do this via the Catalog window.

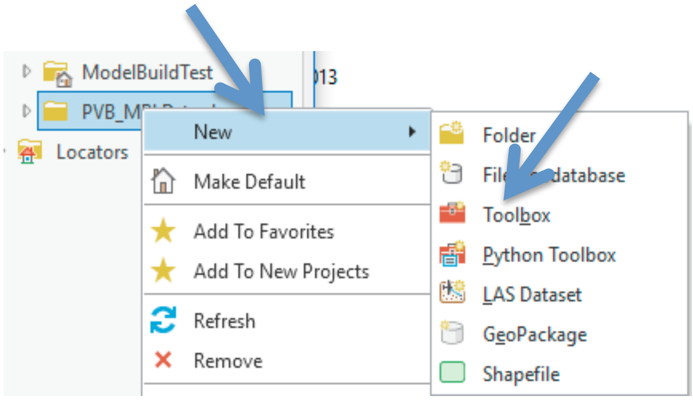
In the Catalog window, right click on Folders, then Add Folder Connection.

Within the add you can select an existing folder, or create a new folder.

In my case, I added a folder named PVB_MBLD_tools.



I then can right click on my linked folder, and create a new toolbox (see below).

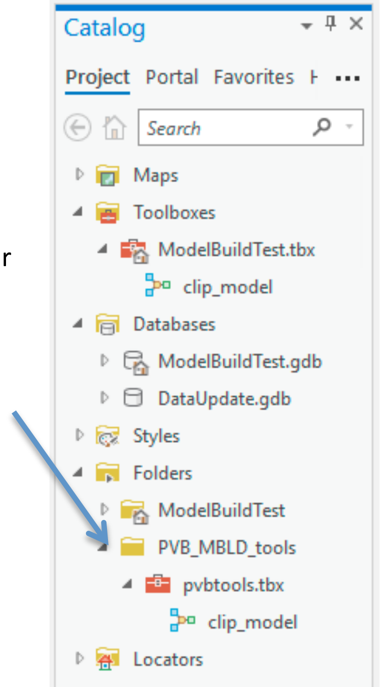


I named this new toolbox pvbtools.tbx (see figure at right, near the bottom).

I then copied the clip_model tool from my project-specific toolbox (ModelBuildTest.tbx),

and pasted it into my new folder toolbox.

I should then just be able to connect to the folder from any project, and run the tool with appropriate parameters. To make the tool truly portable, I need to sure data paths for



important data are accessible within the model. Sometimes this means they must be local to the project, or parameters, but this is a more detailed topic, best left for more advanced exercises.

To Do and Turn In:
Your assignment is to create a model that takes a DEM and a set of pourpoints as input, and generates a set of raster and vector watersheds.

You may use your own data, or use the data in the L:\ESPMx295\Data\ModelBuilder\DEM_PP_ModBld.gdb

The following should be model input parameters:

- input DEM
- input pourpoints layer
- snap distance
- output raster watersheds
- output watersheds

You should create and test your model. When it is working, make a screen grab or other graphic of your model, convert it to a pdf, and turn it in.

You should also turn in your tool, perhaps most easily by saving to a connected folder, and then zipping the entire folder within Windows file explorer, or the compression tool you've been using all semester.

Turn in your model, and a graphic of the model builder flowchart (as a pdf or jpg) in on the course Canvas site, at the time identified.