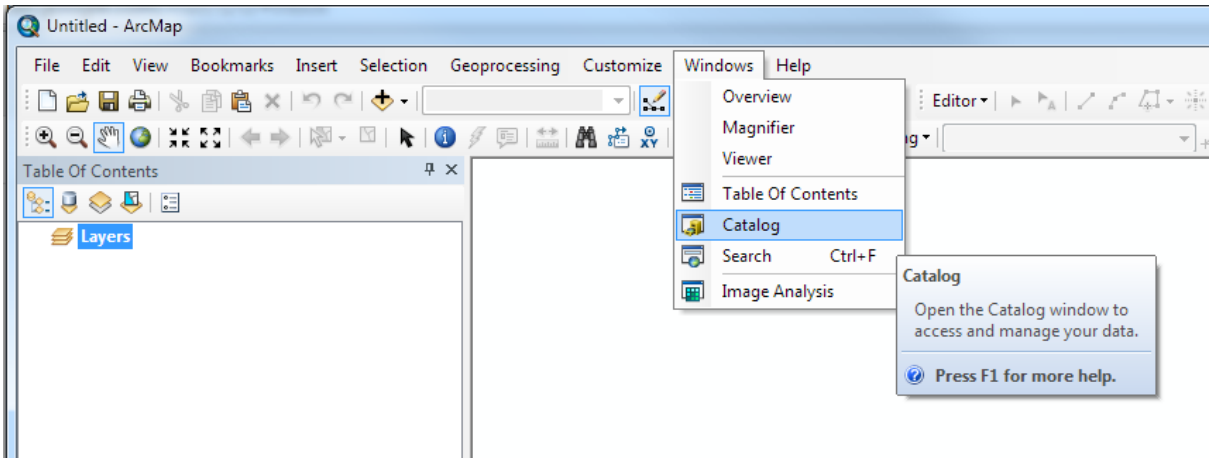
Getting Started with Model Builder

Model builder is a feature in ArcMap that can be used to automate tasks. It is especially useful for batch processing. You can create a model by dragging and dropping objects, tools, etc. and then running the model in ArcMap. You can hard code specifics like file paths into your model or have your model prompt a user for information to make it more flexible. You can also save your model and share it with others. To illustrate how you would get started and to demonstrate some model builder functionality, let's work through a couple of examples. All the files needed to follow along and try out the examples are found in the .zip file you can download from http://maps.library.utoronto.ca/datapub/modelbuilder/ModelBuilder.zip. For the following examples, these files are extracted into a C:\Test\ directory.

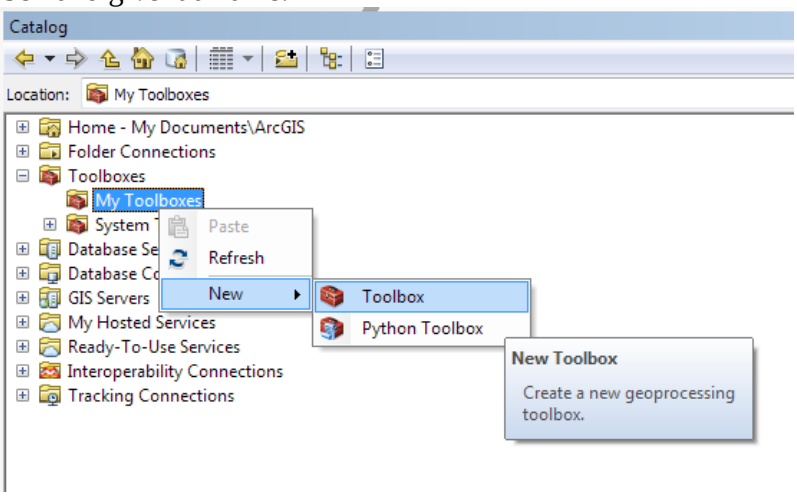
Example 1: Batch Clip

This first example model will be used to process a batch clip. It will iterate through shapefiles in a directory, and clip each one based on another shapefile's area, and save the new clipped shapefiles to a different directory.

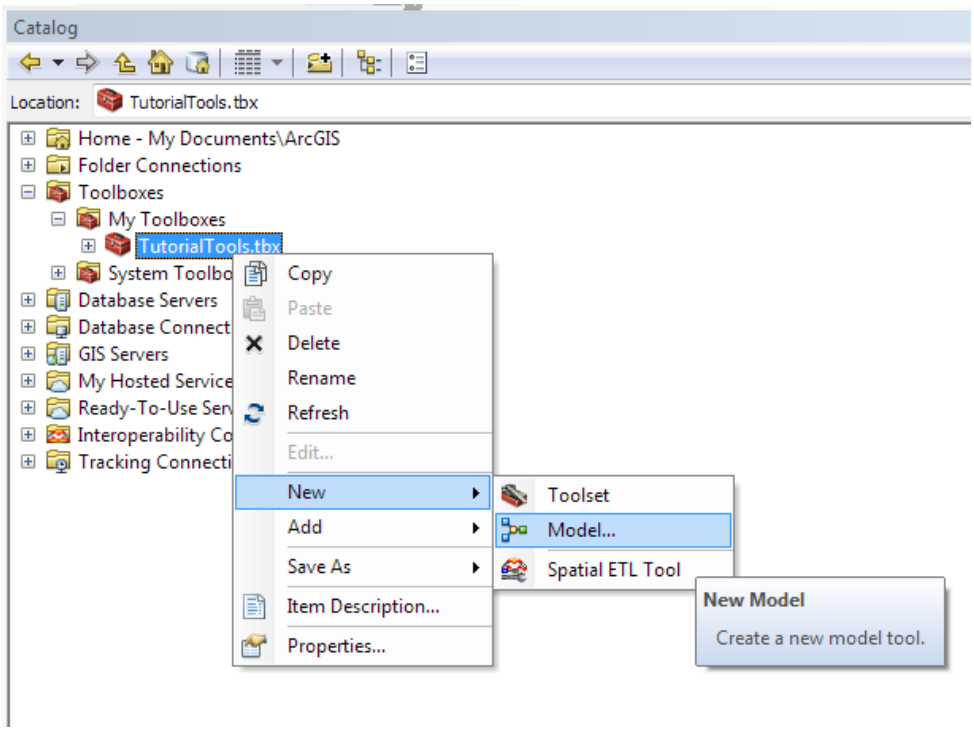
1. First we should create a toolbox where we can save our models. Start ArcMap and open the catalog (**Windows > Catalog**):



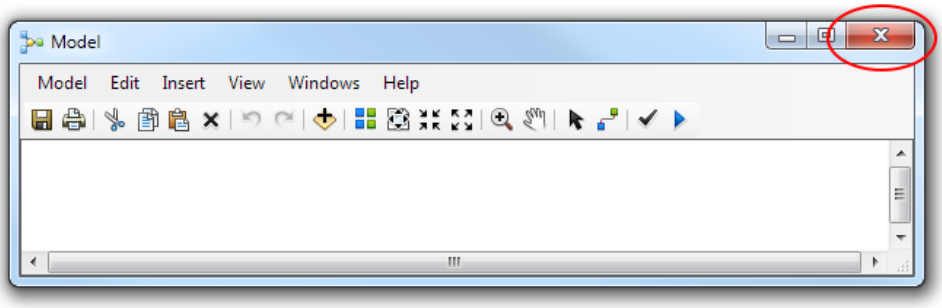
2. In the browse tree that opens on the right, expand *Toolboxes*, right click on *My Toolboxes*, and select **New > Toolbox** and give it a name.



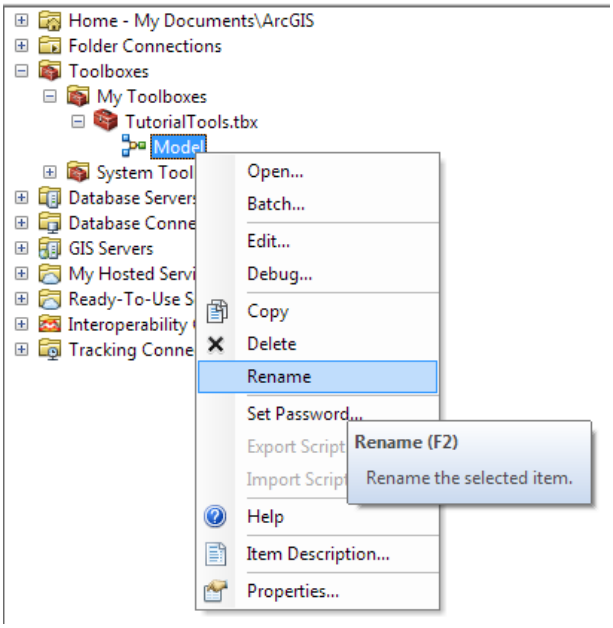
3. Next, we should create a new model. Right click on your new toolbox and select **New > Model**.



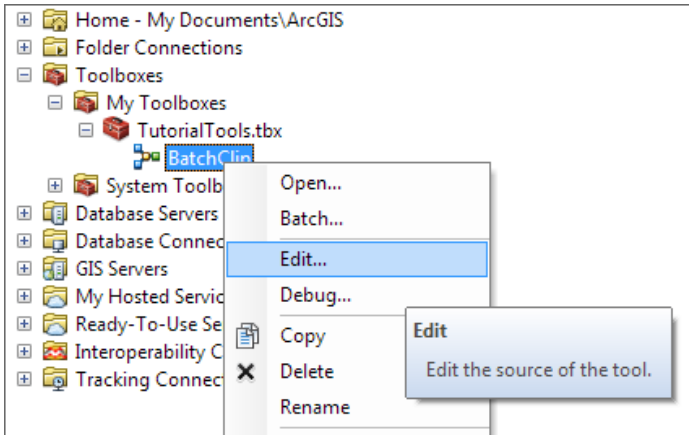
4. The model opens up in a new window. Close it.



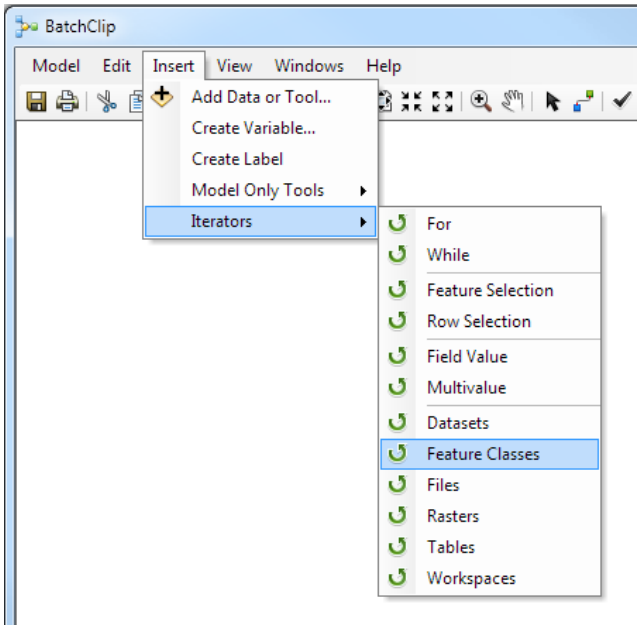
5. Find it in the list of models under your toolbox (it should be called “Model”). Right click on it and select **Rename** and give it a name, in this case “BatchClip”.



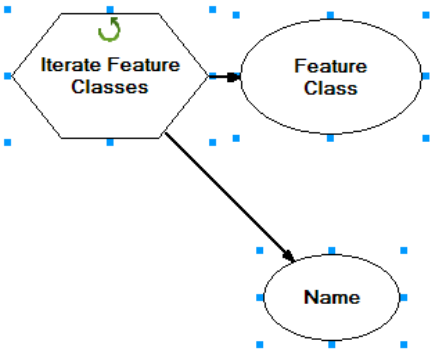
6. To open it up, right click on your new model and select **Edit**. (Note: If you double click on it, it will run the model, not open it up for editing.)



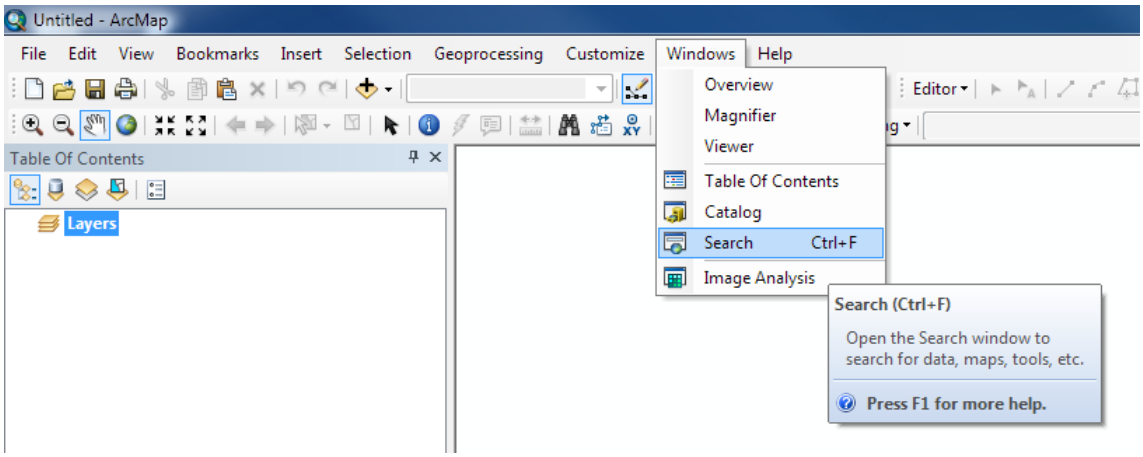
7. Next we can start building the model. First we should iterate through a folder of shapefiles that we want to batch clip. Select **Insert > Iterators > Feature Classes**:



8. You should see the iterator components in your model now. They will be filled in "white" until you have set all the parameters.

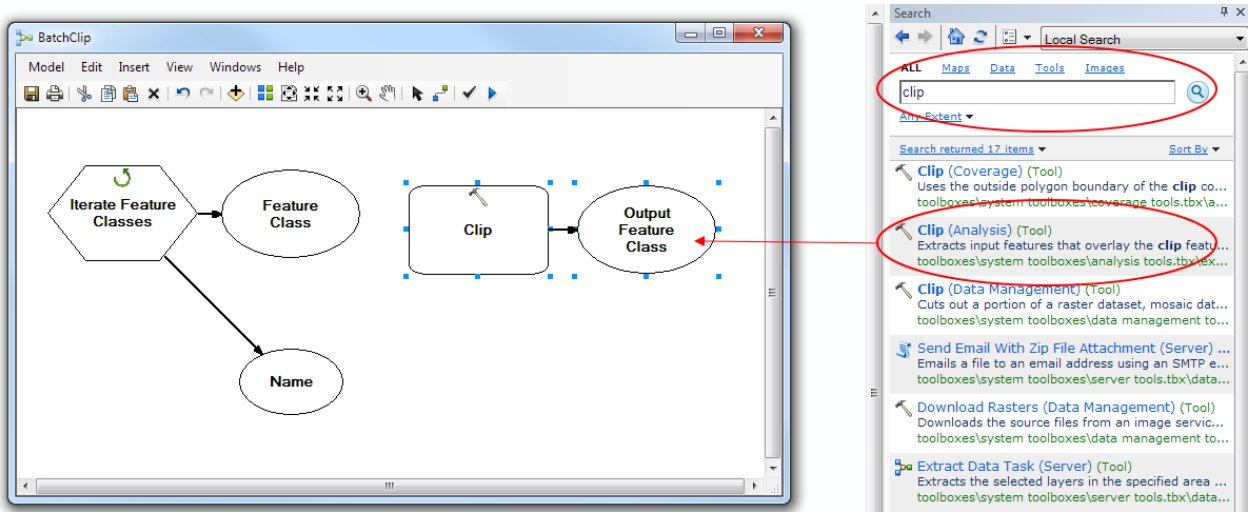


9. Now let's add the other tool we need, the *Clip* tool. Select **Windows > Search** from within ArcMap.

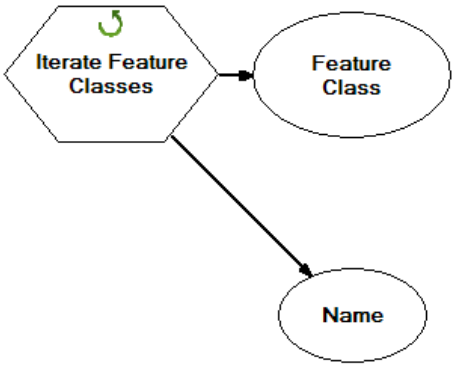


10. Search for the tool, in this case "clip."

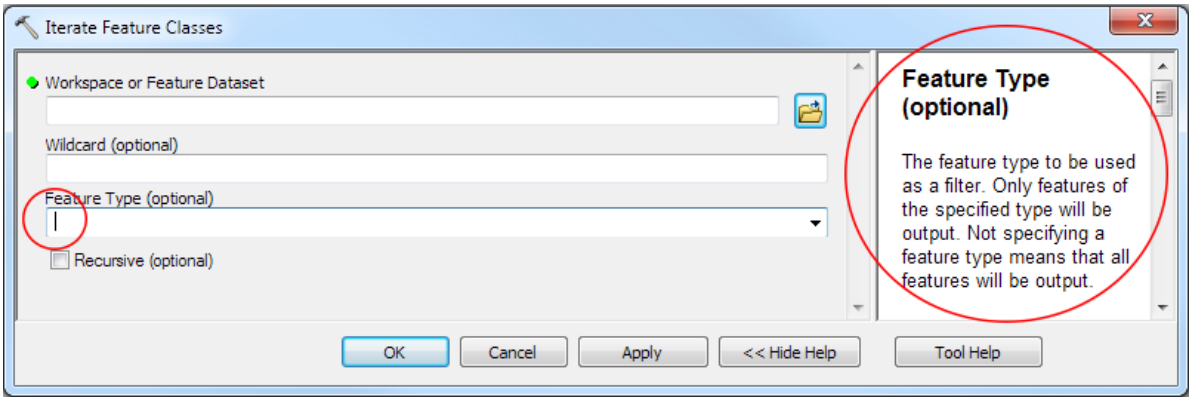
From the list, you can click on the title of the tool you want, in this case *Clip(Analysis)*, and hold down the left button and **drag** it on to the blank canvas of the model next to the iterator block:



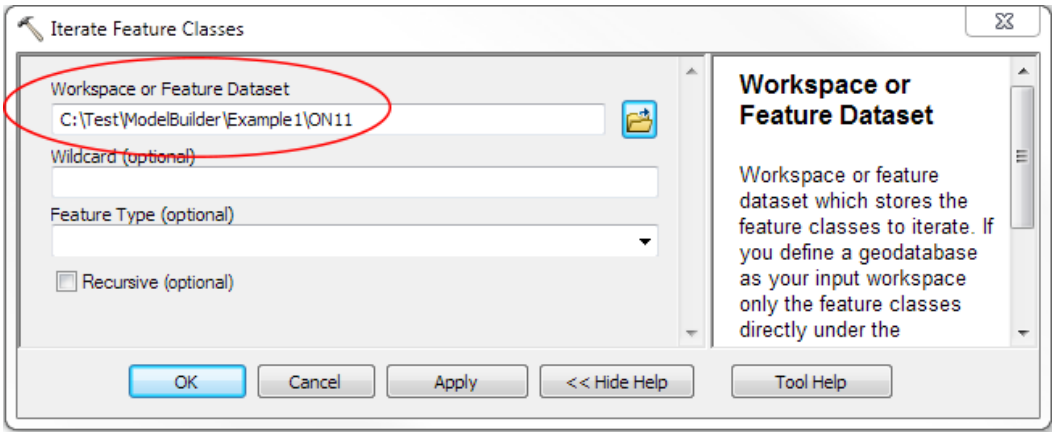
11. Next, we can set the parameters of the iterator. Double click on the *Iterate Feature Classes* hexagon.



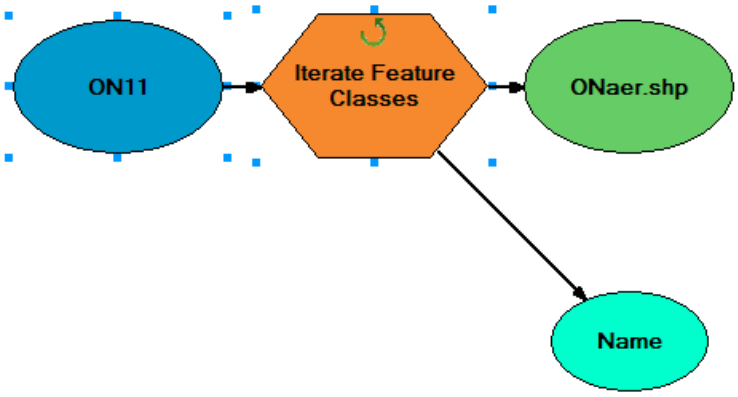
12. Click on "yes" if you are asked questions about any run-time errors. Left clicking once in the input box next to any field, brings up a helpful description on the right of what is required for that field.



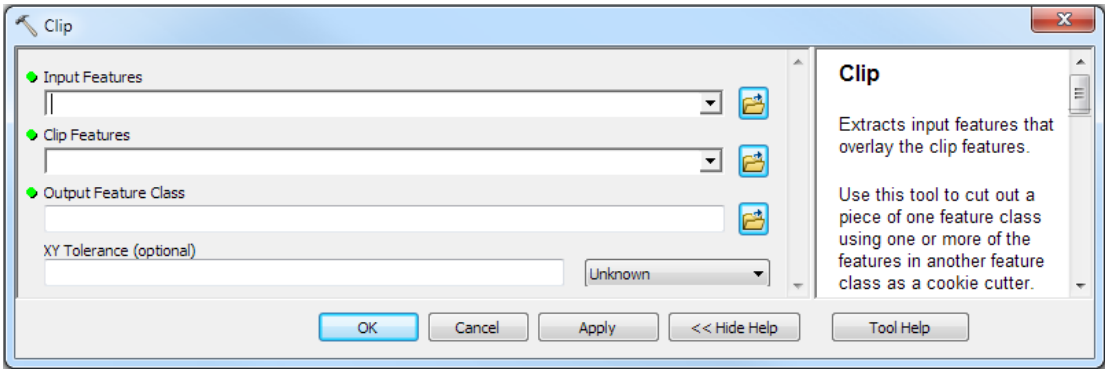
13. For the *Workspace or feature dataset* field, browse to and select the folder that you want to iterate through by clicking on the browse folders icon .
(For this example, select "ON11" from the c:\Test\Example1 directory) and click on **OK** when you are done.



14. You can now see that the iterator components are coloured in which means that this component's parameters have been set, and it is ready to run:

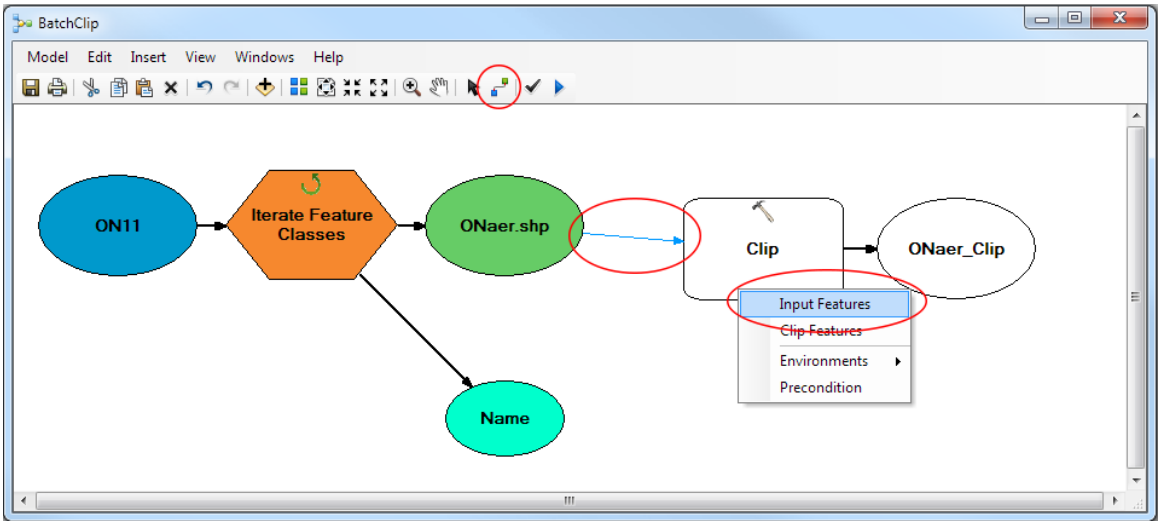


15. Next, we can connect the output of the iterator to the *Clip* tool. First, it is a good idea to see what the C*lip* tool inputs are, and so, similar to step 11, double click on the *Clip* rectangle. We can see that this tool requires *Input Features* (in this case our shapefile data), *Clip Features* (in this case, we have a shapefile that is the outline of the clip area we want), and the *Output Feature Class* file path and name. Click **Cancel** to close the window.

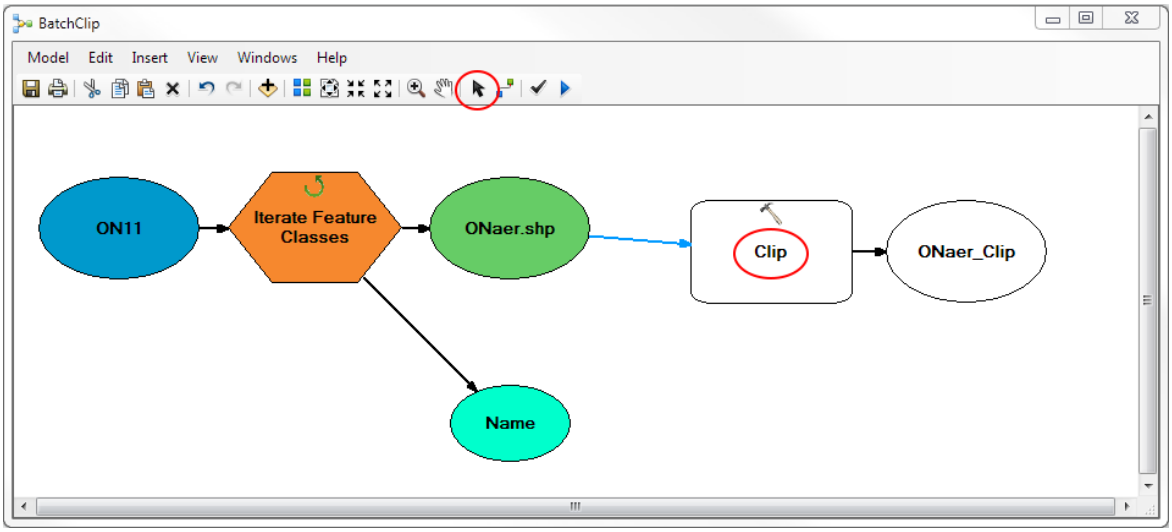


16. What we need to do is connect the *Input feature* with the output from the iterator. The green oval connected to the *Iterate Feature Classes* hexagon represents the output of iteration—its name is the first item to be looked at in the iteration (in this case, ONaer.shp).

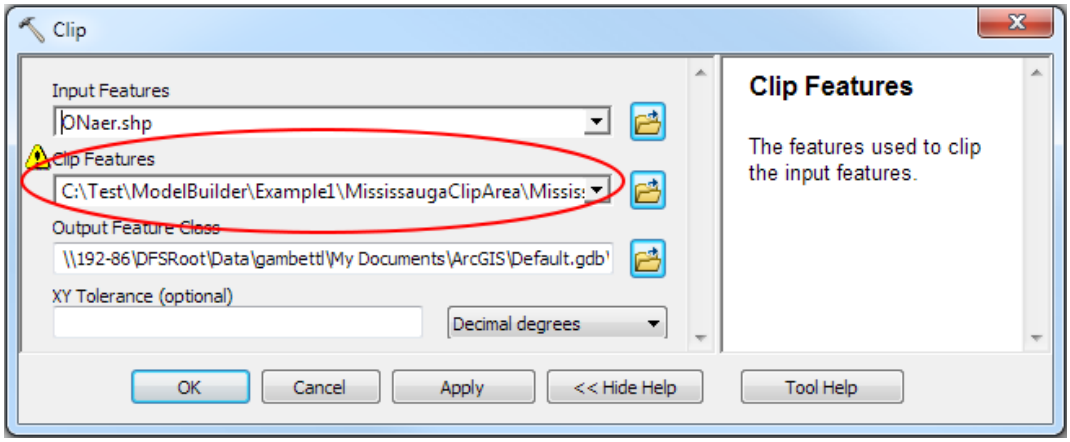
We can use the connect tool  to draw a line from this output to the *Clip* rectangle. When you do this, you should be prompted to decide which field (or other input option) the output should be associated with. In this case, we want it to populate the *Input Features* field.



17. Now that we have the input connected, switch back to the arrow tool by clicking on this icon  , and then double click on the *Clip* rectangle again to fill in the rest of the parameters.

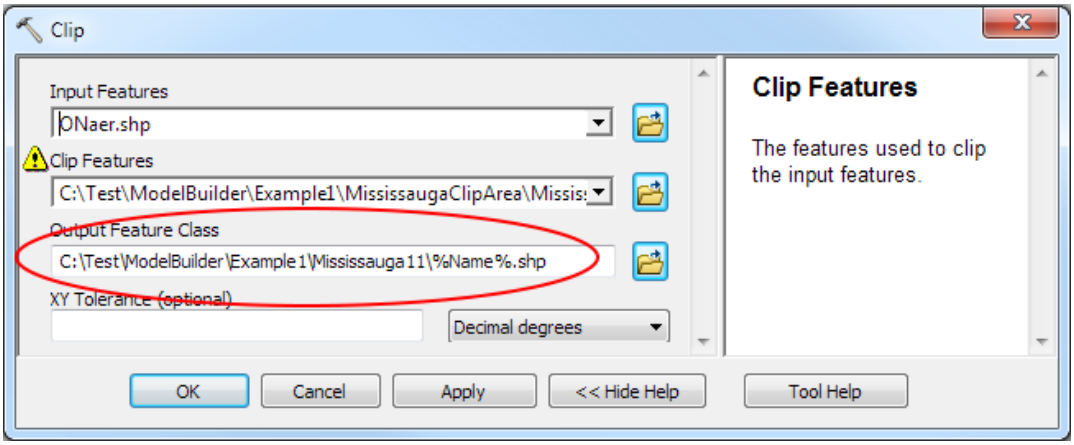


18. For the *Clip Features* field, browse—as you did in Step 13—and select the shapefile you want to use as your clip area (For this example, select “Mississauga.shp” in the “MississaugaClipArea” folder.)

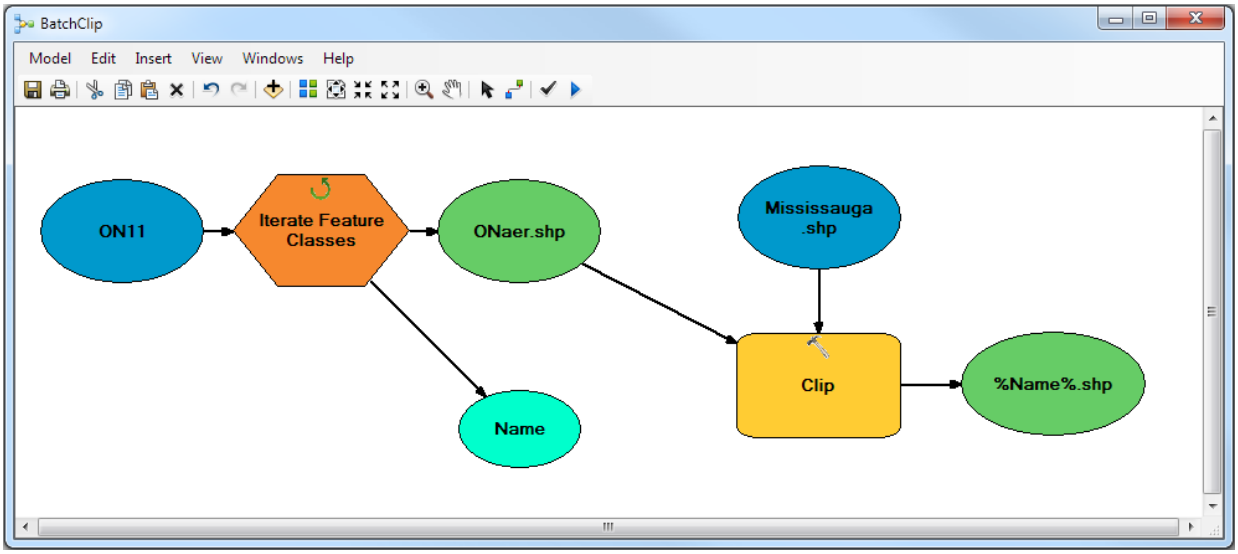


19. For the *Output Feature Class* field, you would normally type in the path where you want to save your new clipped shapefiles.The trick is that you want your output file names to match or be similar to your corresponding input file name. For example, you would like your new shapefile resulting from clipping “ONaer.shp” to also be called “ONaer.shp”, but just found in a new directory.

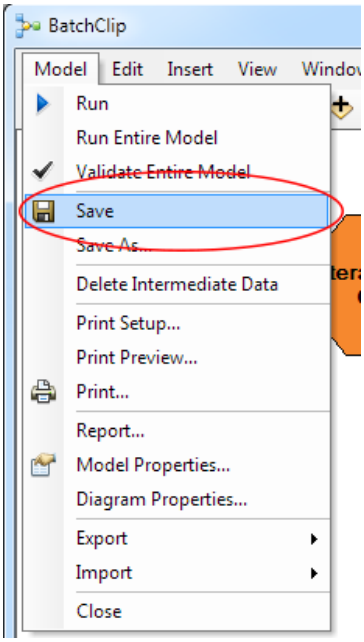
To do this, we can use the *Name* output from the iterator which outputs the name of each file it looks at (minus the file extension) in the *Name* oval.
We can use that output information when filling in the full path of the output file. To do so, you use the percent sign around the name of the output (in this case “Name”) to use its value. So in our example, the path would be “C:\Test\ModelBuilder\Example1\Mississauga11\%Name%.shp” where we are saving our output files to a new directory, but using the same names as the original files and appending “.shp” extension. (String concatenation can be done directly into the field input to create the full path.)



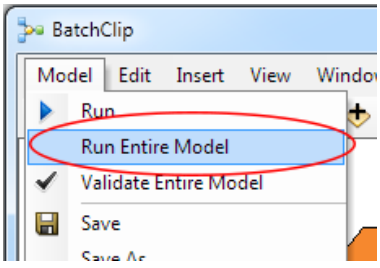
20. Now all model components are linked and coloured in, meaning that model is ready to be run:



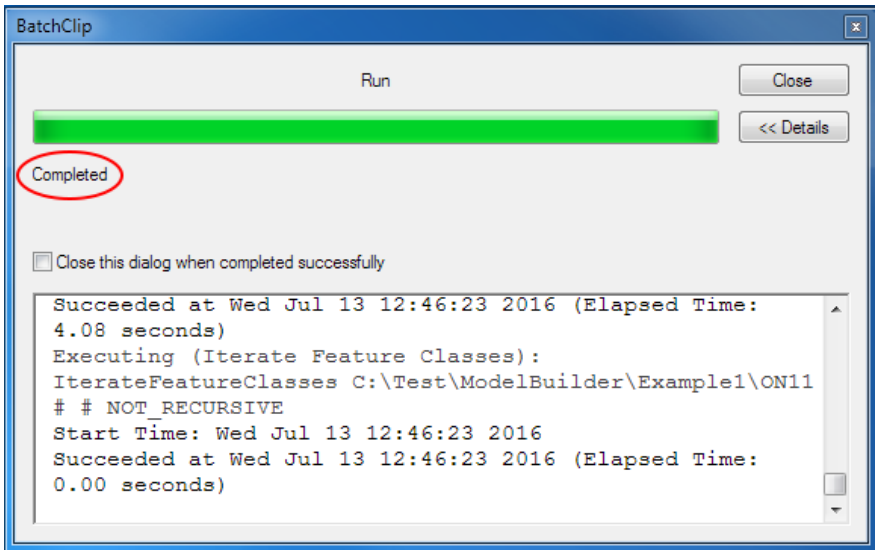
21. Save the model by selecting **Model > Save** or by clicking on the  icon.



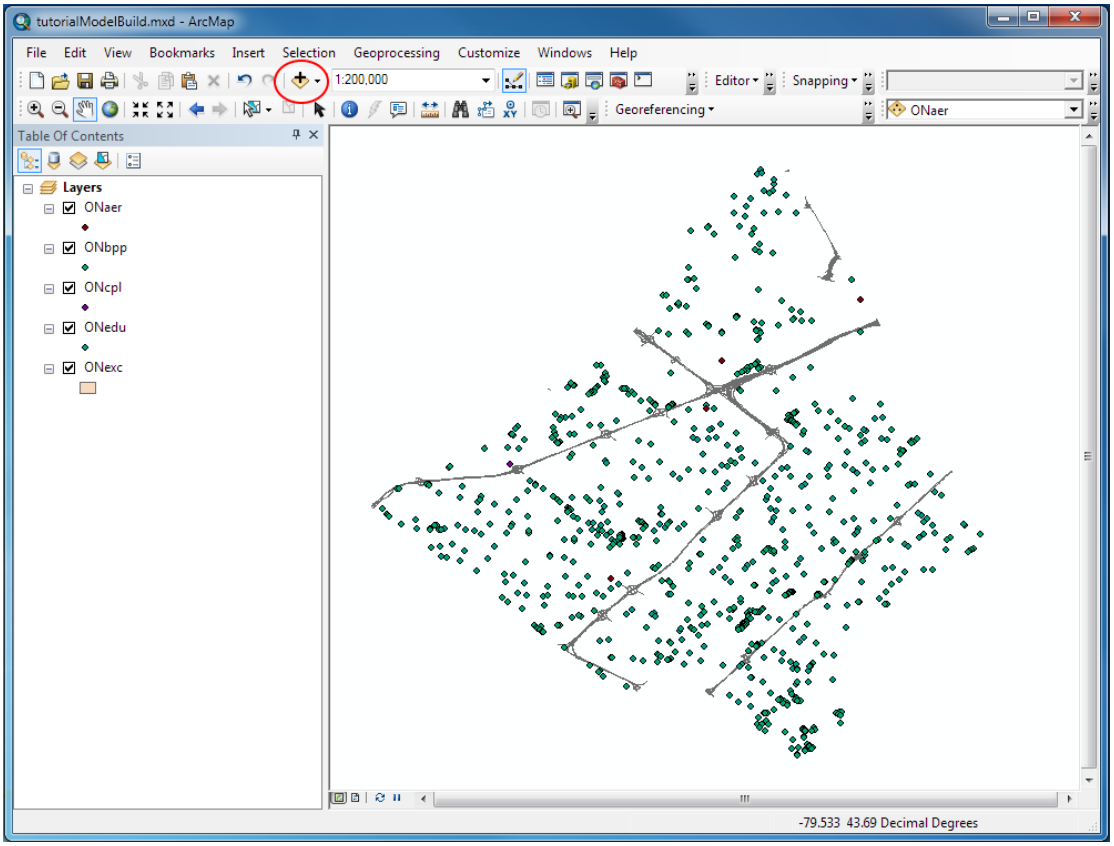
22. Now, we can run the model. Select **Model > Run Entire Model**.



23. You should see a pop-up box that tells you the model is running, gives you information about parameters passed, and will tell you when the model has completed running.



24. You should now have 5 clipped shapefiles in the C:\Test\Example1\Mississauga11 directory. You can check that the model ran correctly by adding the shapefiles as layers in ArcMap, using the  icon, and verifying that they show just coverage for Mississauga and not all of Ontario.

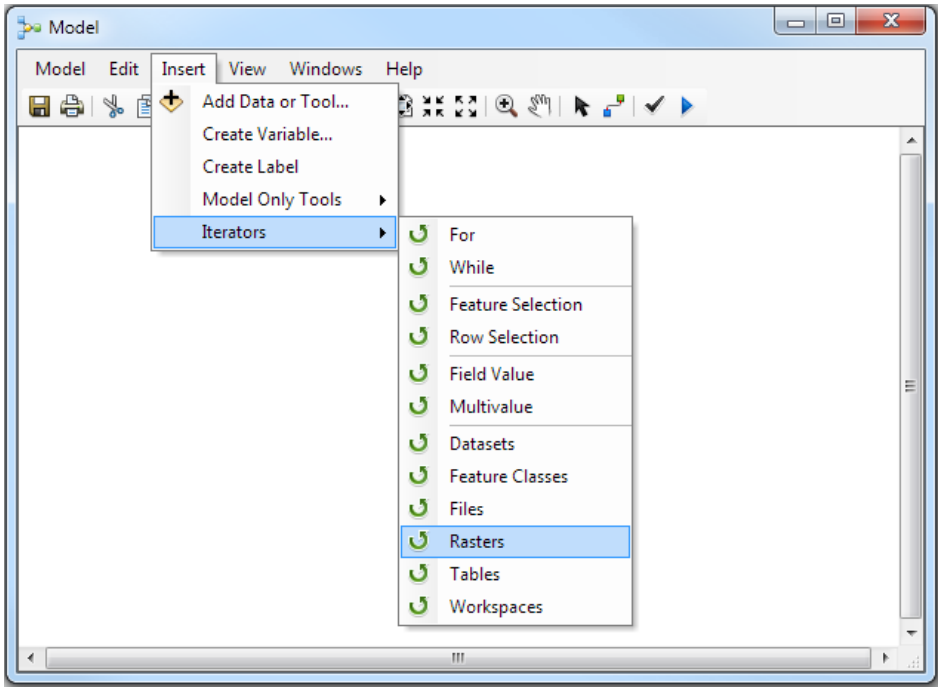


Example 2: Reproject Rasters

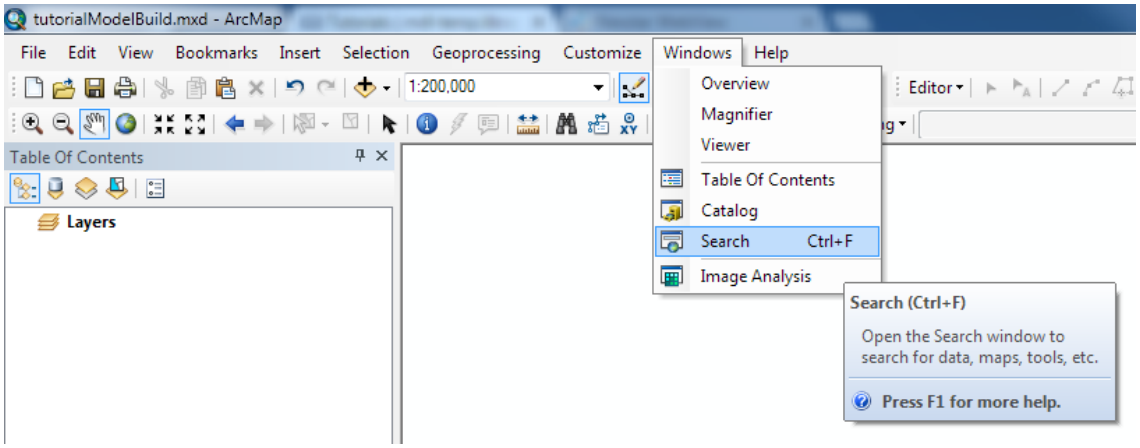
This second example will be used to illustrate some additional functionality of model builder. Concepts such as user parameters, preconditions, and using the model only tool "Calculate Field" will be demonstrated. This model will be used to reproject a number of raster images. It will iterate through images in a directory, reproject each one, and then save the new raster images to a different directory. The user is prompted for the input and output directories, projected coordinate systems, and the format of the images.

1. Follow step 3 of Example 1 to create a new model.

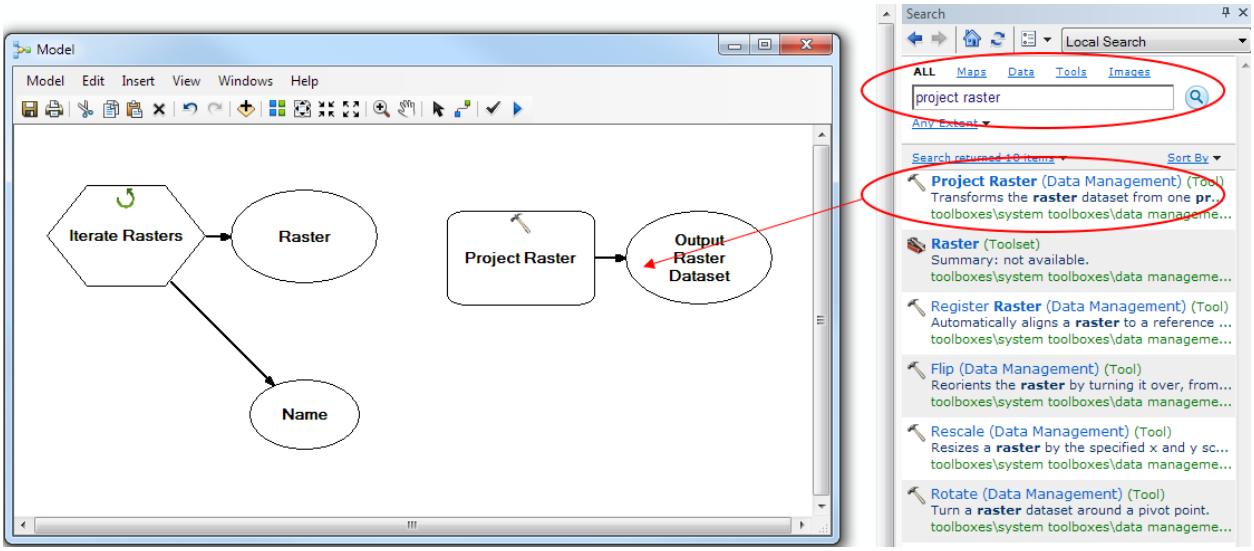
2. First we should iterate through a folder of raster images that we want to reproject. Select **Insert > Iterators > Rasters**.



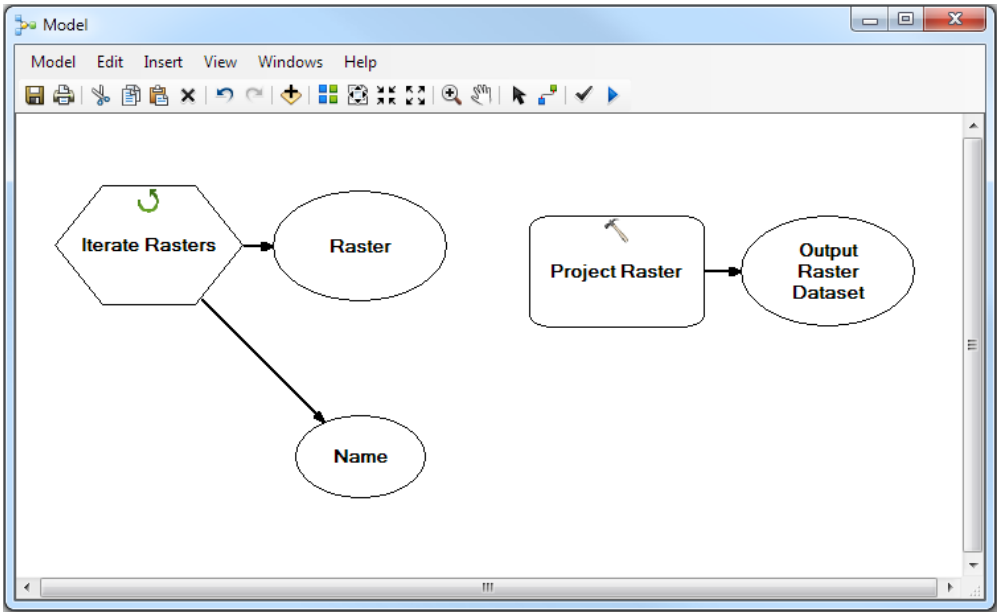
3. Now let's add the other tool we need, the Project Raster tool. Similar to steps 9 and 10 of Example 1. First, bring up the search in ArcMap, **Windows > Search**.



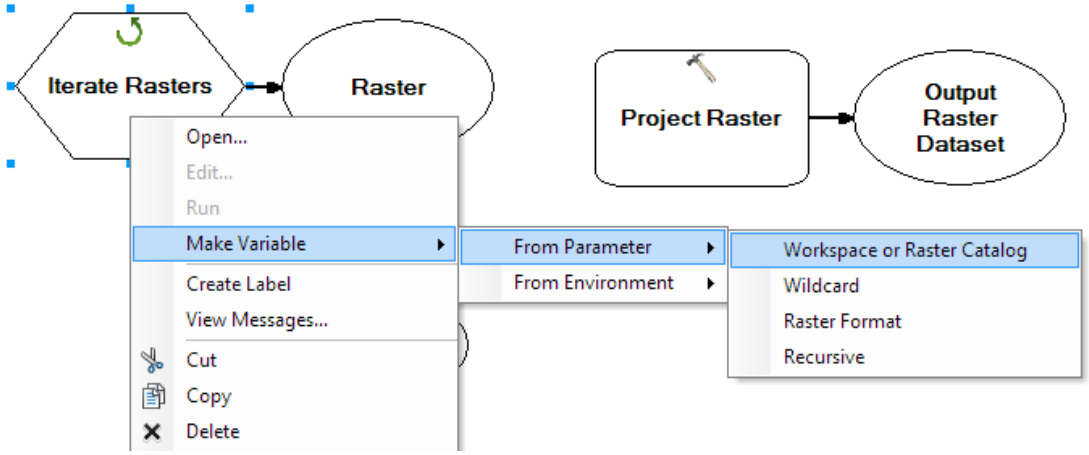
4. Search for the tool, in this case “Project Raster.” From the list, you can click and hold on the title of the tool you want, in this case *Project Raster (Data Management)*, and drag it on to the blank canvas of the model next to the iterator block.



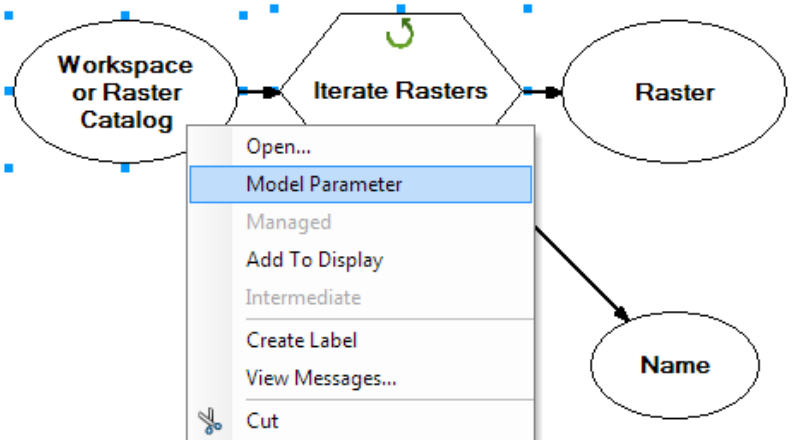
5. After dragging the Project Raster tool into the model window, your model should look like this (remember, when the objects are white, they cannot be run until you set the parameters):



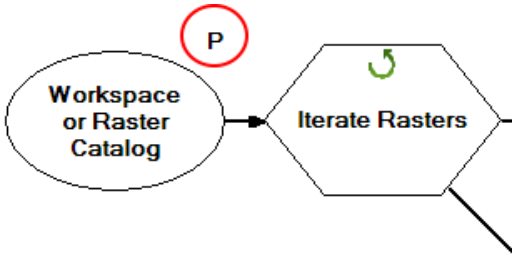
6. Next, we can set the parameters of the iterator, but instead of hard coding the parameters (like we did in Example 1), such as the input directory, we can prompt the user to give us that information. Right click on the *Iterate Rasters* hexagon and select **Make Variable > From Parameter > Workspace or Raster Catalog.** What this does is make an input parameter for one of the iterator's inputs—the directory/workspace where the raster files are.



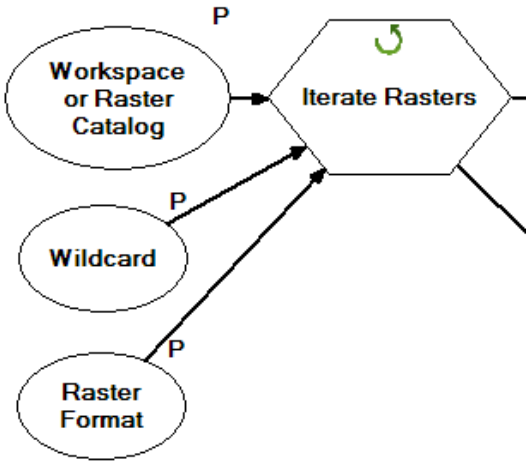
7. Next, right click on the new *Workspace or Raster Catalog* oval that is created to represent the variable and select **Model Parameter**:



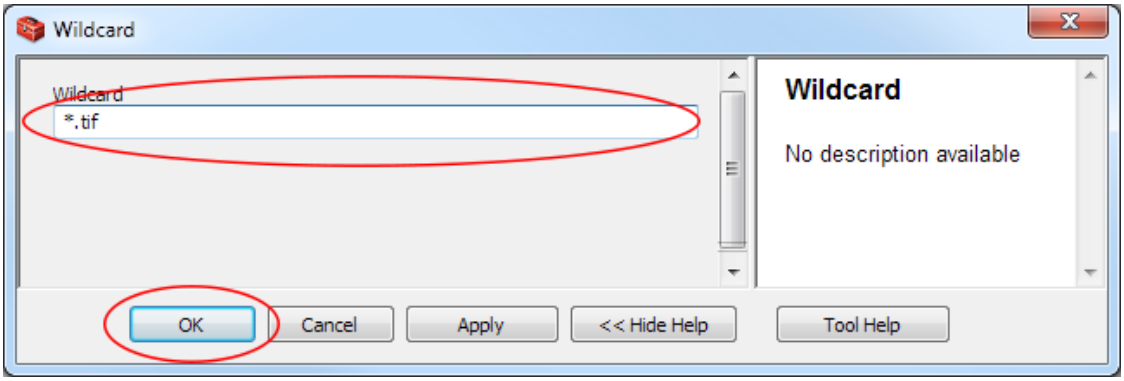
8. You should now see a "P" near the top right of the oval, which means that this variable is populated by user input. When the model is run, the user will be prompted to select a directory/workspace where the raster images are.



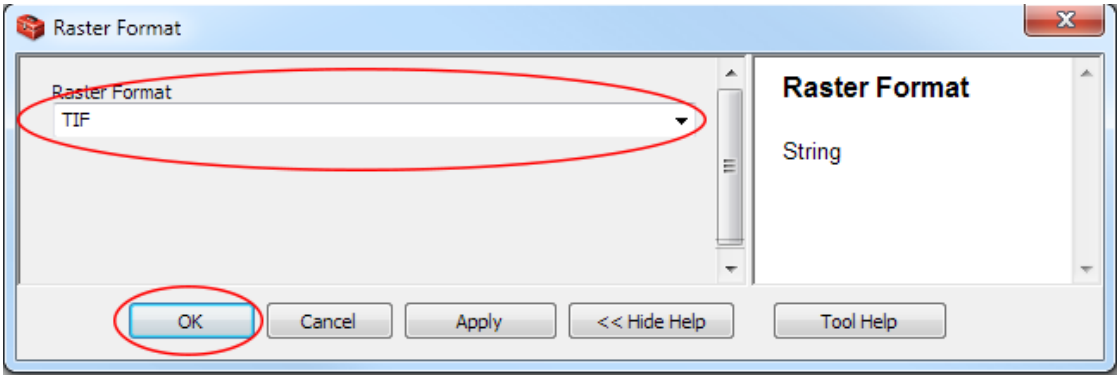
9. Create user input model parameters for the iterator's other two inputs, *Wildcard* and *Raster Format*, following steps 6 and 7. Your model should look like this:



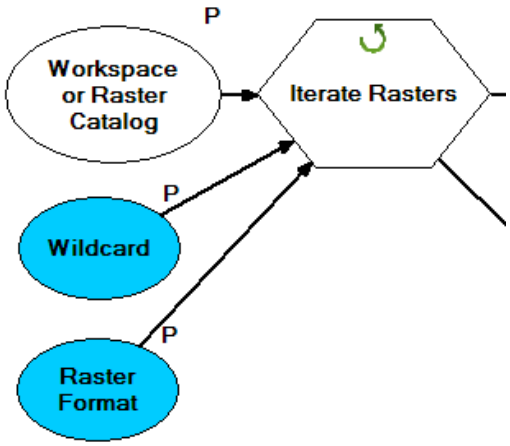
10. For the *Workspace or Raster Catalog* model parameter, we did not set a default, so the user must always provide that information. For *Wildcard* and *Raster Format* parameters, let's set default values. Double click on the *Wildcard* oval and fill in what default you would like, for this example, enter *.tif and then click on **OK**.



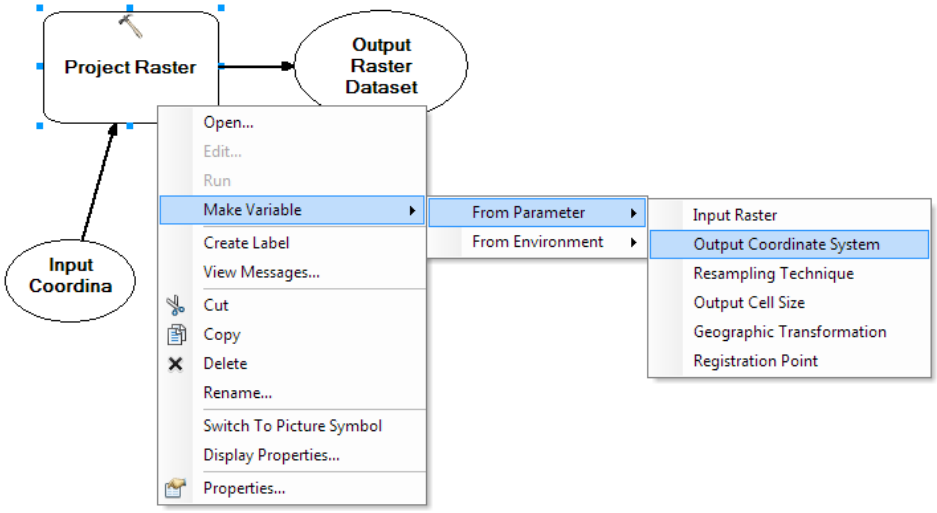
11. Double click on the *Raster Format* oval and again fill in a default. For this example, select TIF from the dropdown menu and then click on **OK**.



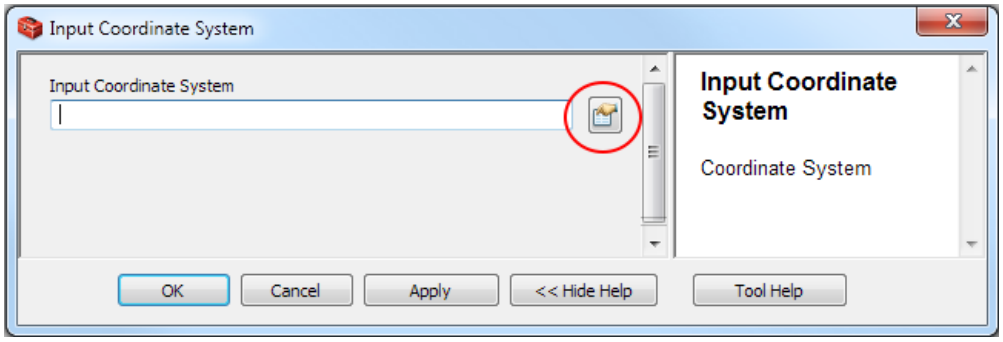
12. Those two ovals should now be coloured blue, meaning they have enough information to run without user input, if necessary.



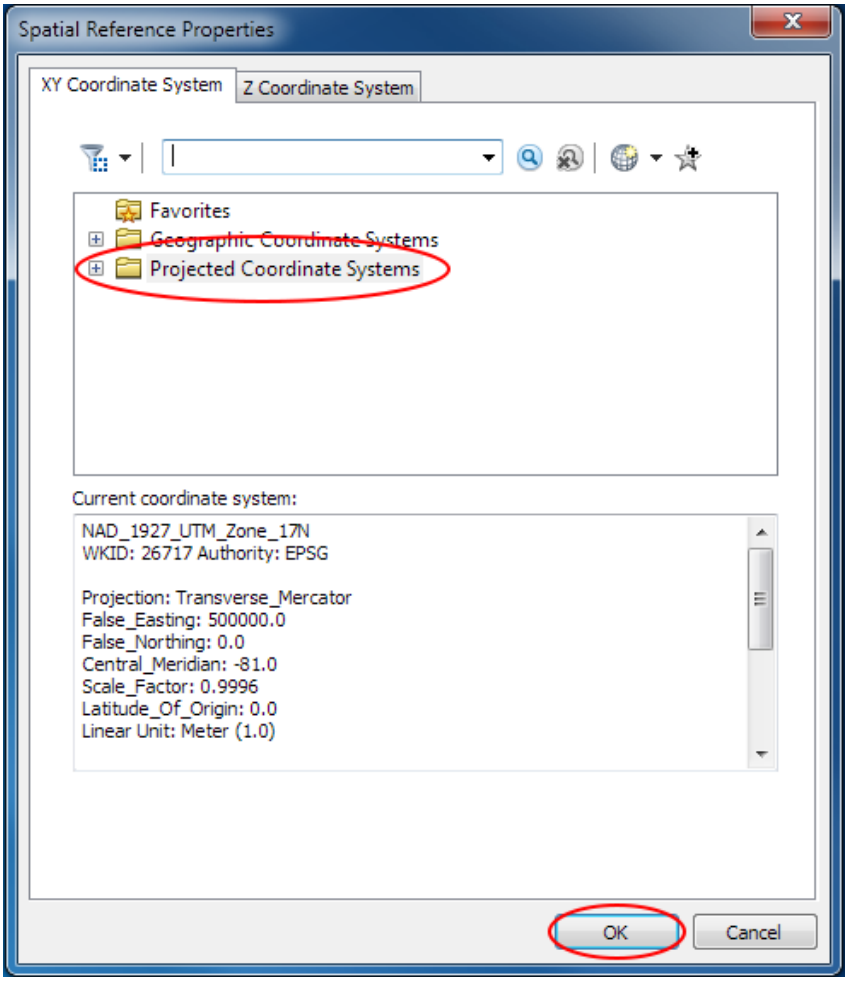
13. Next we can set the parameters of the *Project Raster* tool. First, create the user input model parameters for the *Input Coordinate System*, *Output Coordinate System*, and *Geographic Transformation* variables for the tool, following the procedures from steps 6 and 7.



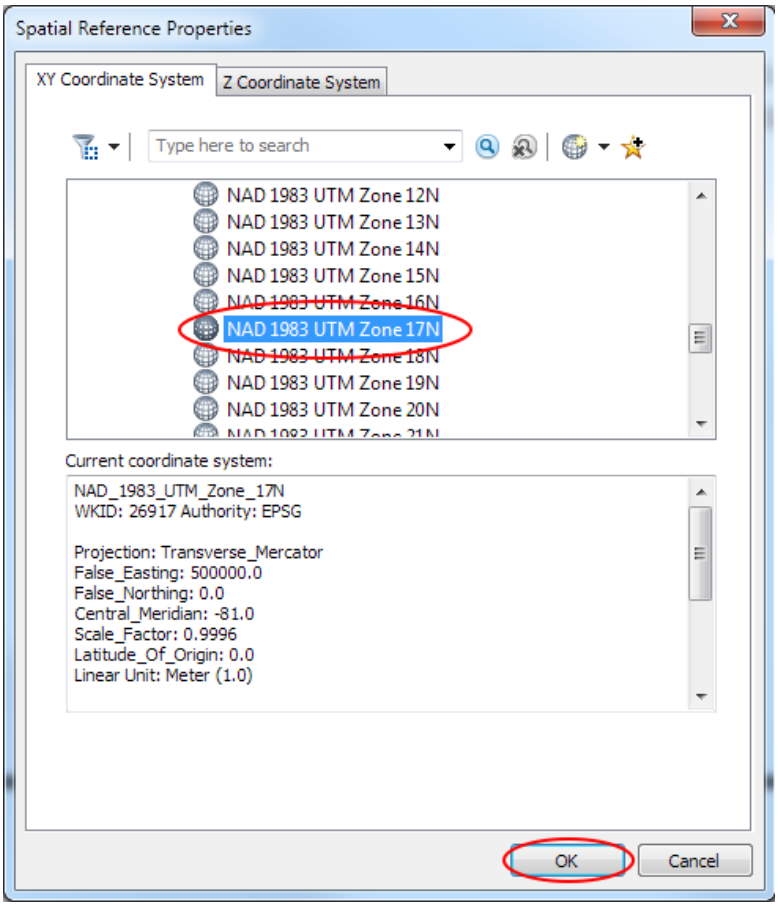
14. Next, let's set the defaults. For this example, set the input coordinate system default by double clicking on the *Input Coordinate* oval. In the window that opens, click on the  icon.



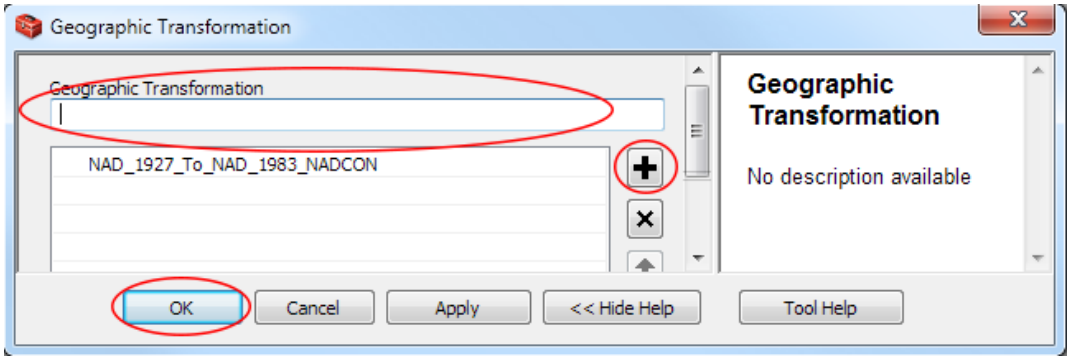
15. Then expand the folders to navigate: Projected Coordinate Systems > UTM > NAD 1927 > NAD 1927 UTM Zone 17N. Select NAD 1927 UTM Zone 17N and then click **OK**. Click **OK** again to close the Input Coordinate System window.



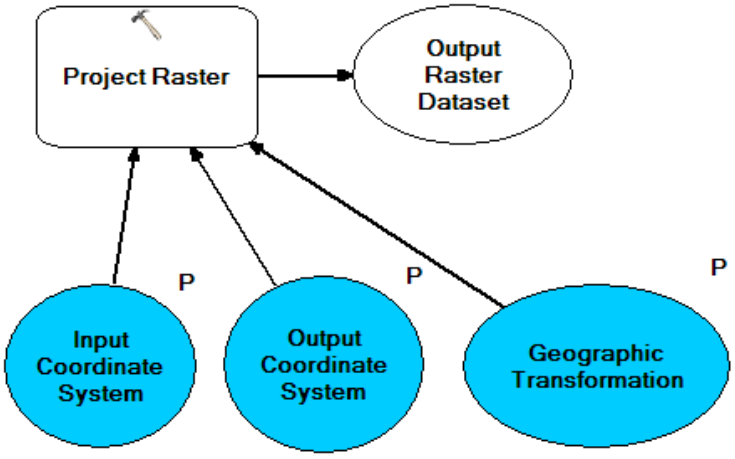
16. Set the output coordinate system in the same manner as steps 14 and 15, but browse to “Project Coordinate Systems\UTM\NAD 1983\” and select “NAD 1983 UTM Zone 17N.prj”.



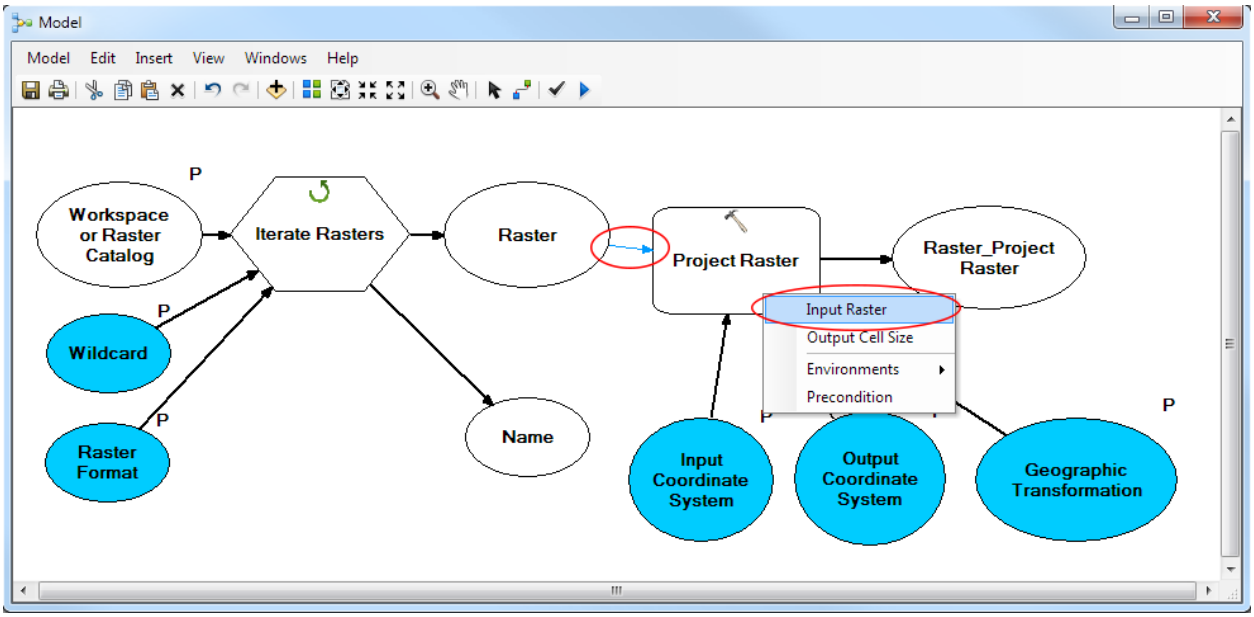
17. Set the geographic transformation default by double clicking on the *Geographic* oval. In the *Geographic Transformation* field, type in “NAD_1927_To_NAD_1983_NADCON”, click on the  icon, and then click **OK**.



18. Once you create variables for those fields, set them as model parameters so that they are user inputs, and set default values for each one, you should see the following:

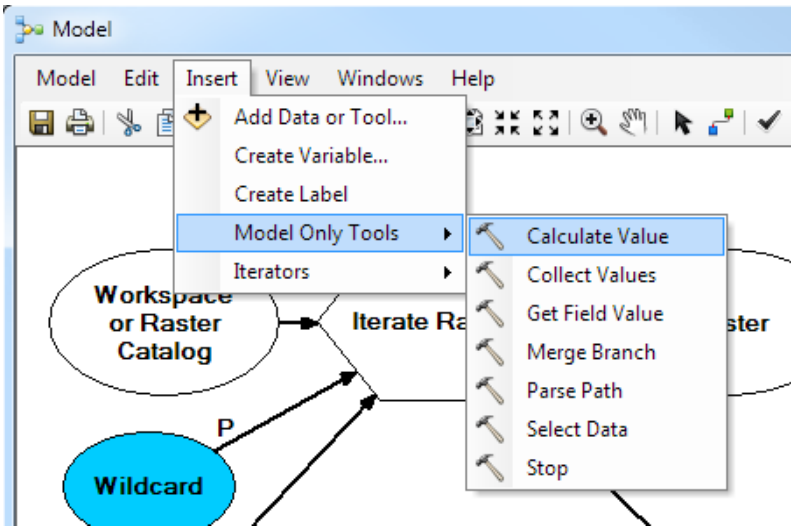


19. Next we need to set the input raster for the tool. We can use the connect tool to draw a line from the *Raster* oval (the output raster image from the iterator) and the *Project Raster* rectangle. When you do this, you should be prompted to decide which field (or other input option) the raster should be associated with (it will only list fields that are the same feature type, giving you helpful hints if there are type incompatibilities). In this case, we want it to populate the *Input Raster* field.

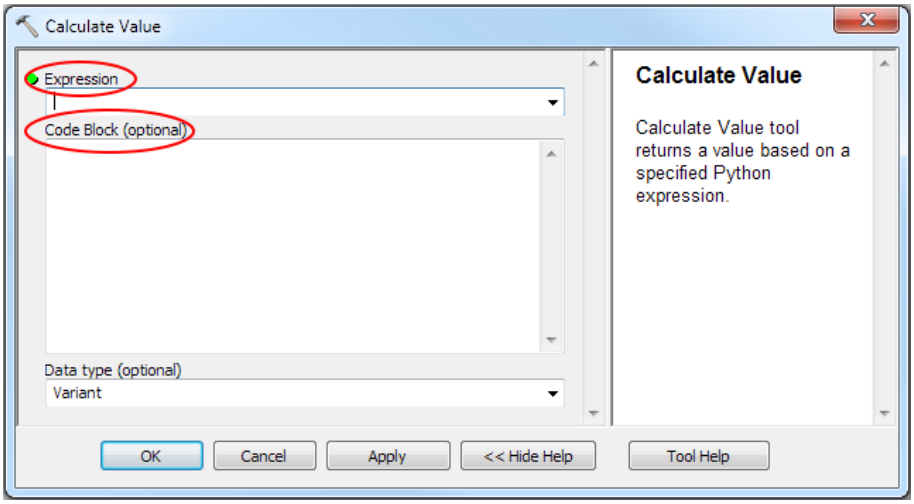


20. After using the connect tool, we must remember to switch back to the regular select tool before continuing our work on the model. Click on the icon  .

21. Next we will prepare the filename for the *Output Raster Dataset* output of the *Project Raster* tool. In order to adjust filenames, we will need to use python code. To use python in our model, we will need to use a Model Only Tool, *Calculate Value*. For our example, we will take a filename like "Test1.tif," replace the ".tif" with "reproj.jpg" so that the output files will be called "Test1reproj.jpg." First we will insert a calculate value tool, by going to **Insert > Model Only Tool > Calculate Value.**



22. The *Calculate Value* tool is used to execute short snippets of python code. If you double click on the *Calculate Value* rectangle, you can insert a python expression or even full functions in the code block section.



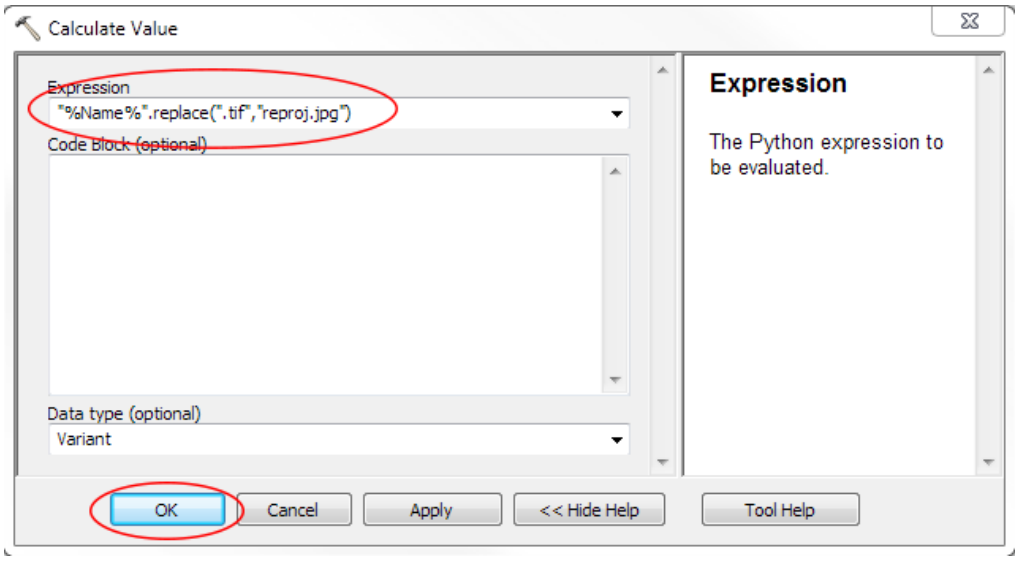
23. For our example, we want to use python string commands in an expression to modify the filename. That expression will be used to populate the *Expression* field in the *Calculate Value* tool. First we want to get the filename of the current file in the iteration, which is stored in the *Name* variable output of the iterator. To do this we can use percent signs around the variable name. We also surround this expression by quotation marks to signify that the value is a string:

“%Name%”

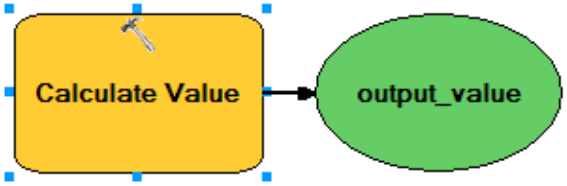
24. Taking this string, we can call the python string command “replace” to replace any instances of the string “.tif” with the string “reproj.jpg”, by adding “.replace” at the end of the string and passing it two parameters: the original string we are looking for and the string we want to sub in:

"%Name%".replace(".tif","reproj.jpg")

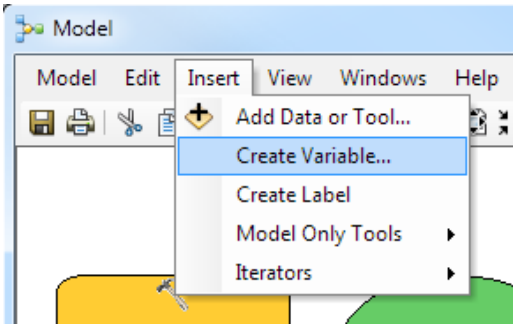
25. Fill in the *Expression* field with the full python expression we created in step 24 and click on **OK.**



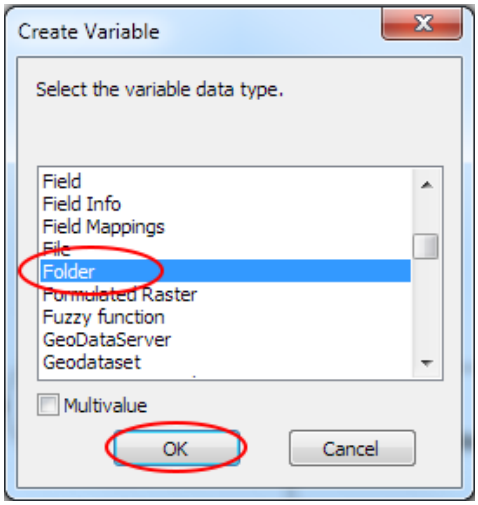
26. The *Calculate Value* components should now be coloured in saying they have enough information to run and the output of the calculation is called “output_value”:



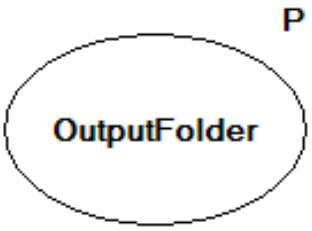
27. We will then append this file name to an output path (that we will also prompt the user for) to populate the *Output Raster Dataset* field of the *Project Raster* tool. To prompt the user for the output directory, select **Insert > Create Variable…**



28. Select "Folder" for the type and click on **OK.**



29. Right click on it and select **Rename** to give it a more meaningful name like "OutputFolder." Then right click on it again and select **Model Parameter**. Now the user will be prompted to select an output folder when the model is run. (Note: It stays white as it has no information until the user gives it some. You could set a default output path here by double clicking on it. Then it would change to blue.)



30. Next, we want to connect this information together to create an output path. Double click on the *Project Raster* tool. To fill in the *Output Raster Dataset* field we need to specify the full output including the output filename. We can create this path using the information from the *OutputFolder* variable we just created with the *output_value* variable we created from the *Calculate Value* tool.

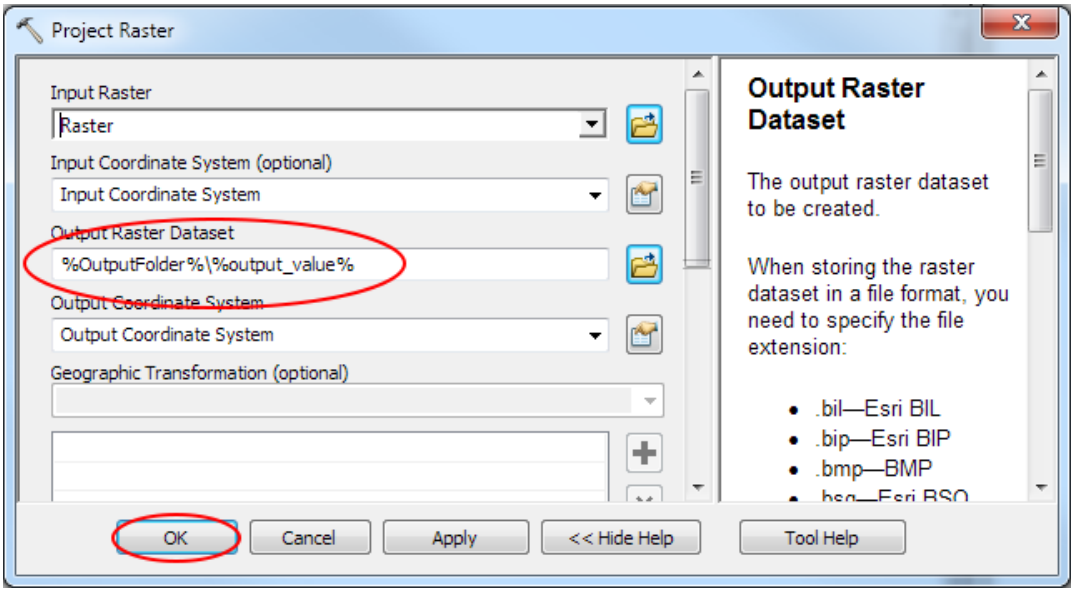
First we want to get the string indicating the path of the output folder, which is stored in the *OutputFolder* variable. To do this we can use percent signs around the variable name, similar to step 23:

%OutputFolder%

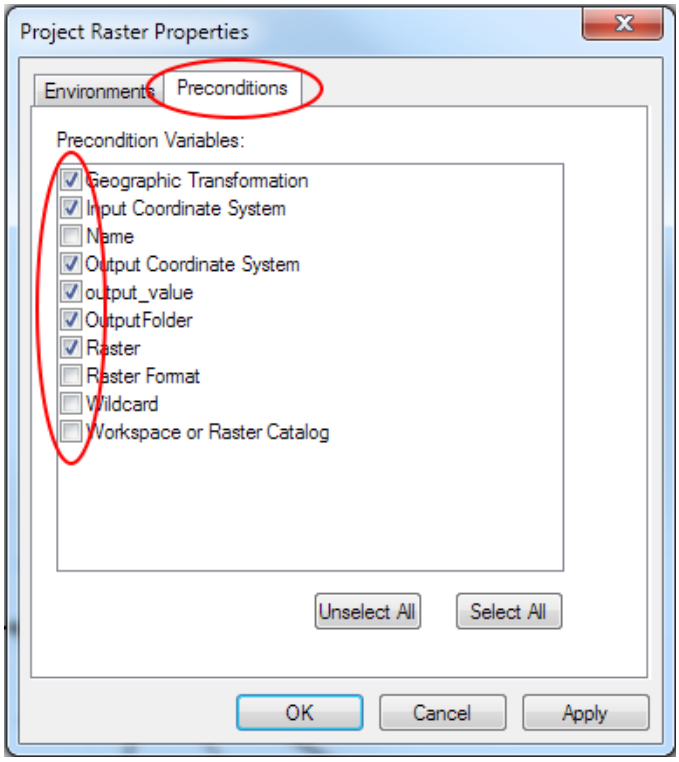
31. To get the string for the output filename, we can do the same, but with the variable *output_value*, to get:

%output_value%

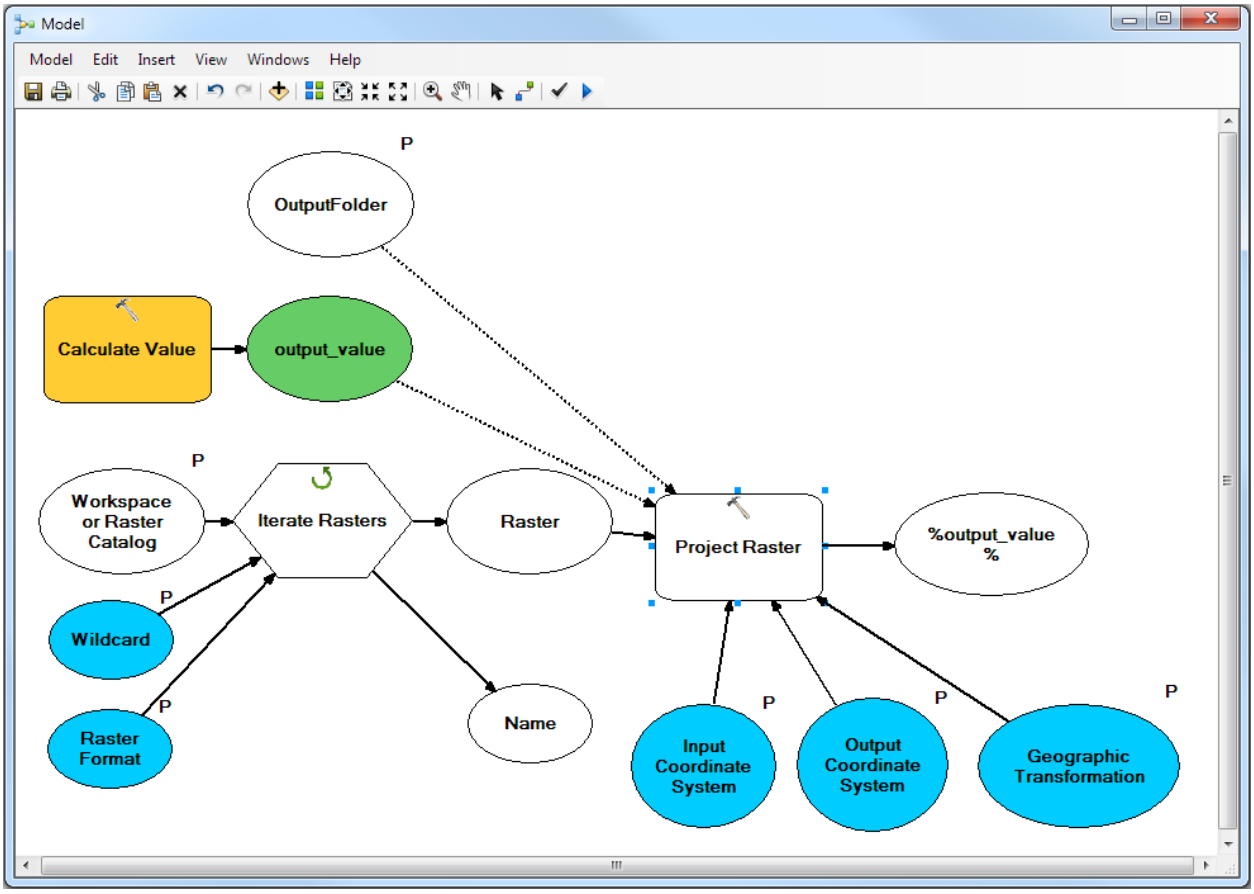
32. To put it all together, you put **%OutputFolder%\%output_value%** using a **"\"** to connect the two parts (without the quotation marks) in the *Output Raster Dataset* field. This pulls the value of the output folder (e.g., C:\Test\Example2\output) and the value of the output filename we calculated from our python expression (e.g., 16372reproj.jpg) and puts them together to create a full output path (e.g., C:\Test\Example2\output\16372reproj.jpg). Type **%OutputFolder%\%output_value%** in the *Output Raster Dataset* field and then click on **OK.**



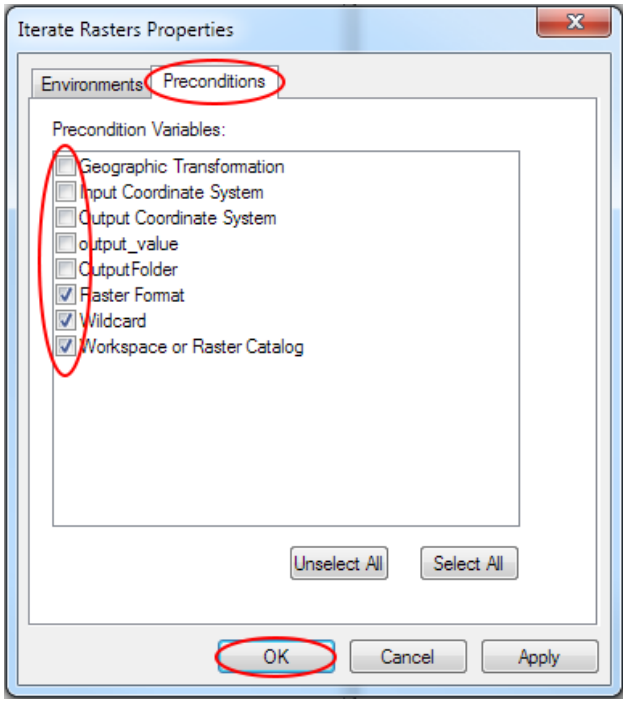
33. Next, we have to set preconditions to make the model run as we intend. If we ran this model as is, it may not run in the order we expect. For example, it may project the raster before calculating the output filename, causing the file to be incorrectly named. The *Project Raster* tool relies on a lot of other information gathered from the user and the model to run. We need to tell the *Project Raster* tool to wait until all the information is gathered before running. Right click on the *Project Raster* tool and select **Properties.** Select the **Preconditions** tab. You can put checkmarks next to items in the list that you want it to wait for (i.e. variables it uses to run). Put checkmarks next to *Geographic Transformation, Input Coordinate System, Output Coordinate System, output_value, OutputFolder,* and *Raster*, and then click on **OK**.



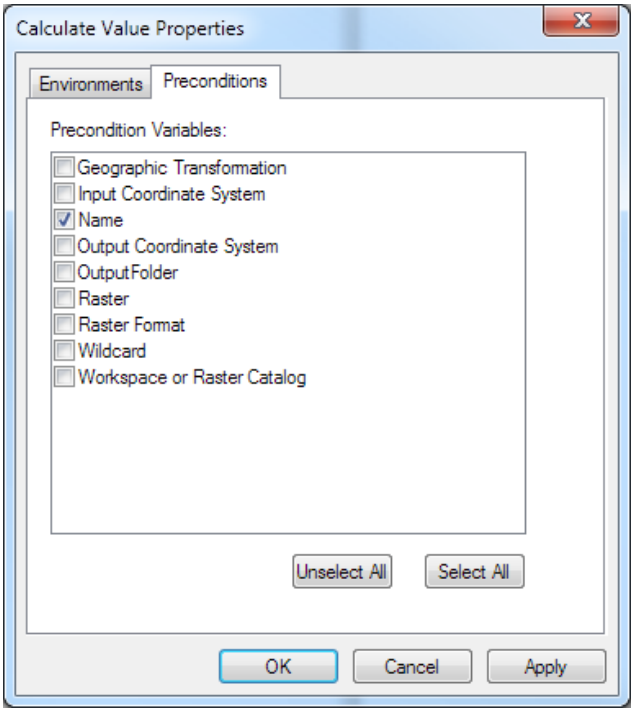
34. You will see dotted lines now connecting up the variables to the *Project Raster* tool, illustrating that the tool must wait for these variables to be set, in other words, showing its precondition relationships.



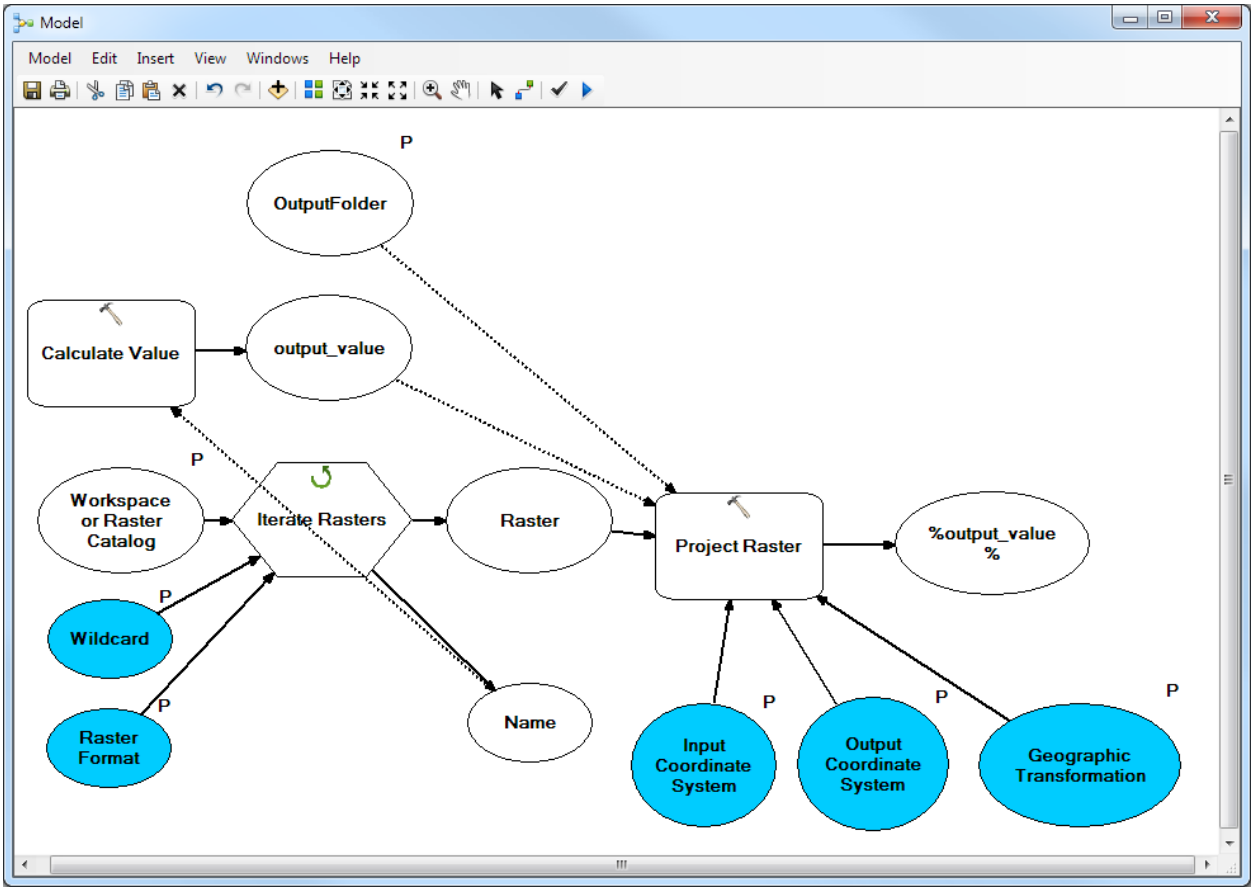
35. Preconditions should be set on each tool in this model to make sure that it waits for all the required inputs to be ready before processing; therefore, set preconditions for the *Iterate Rasters* and *Calculate Value* tools.
Following step 33, for the *Iterate Rasters* tool, select the *Raster Format*, *Wildcard*, and *Workspace or Raster Catalog* variables as preconditions.



For the *Calculate Value* tool, select the *Name* variable as a precondition.

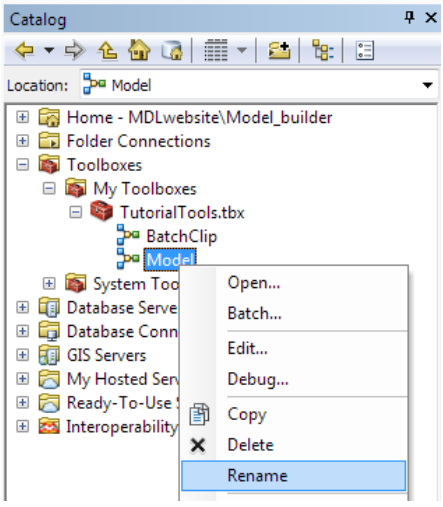


36. Once completed, your model should look like this:

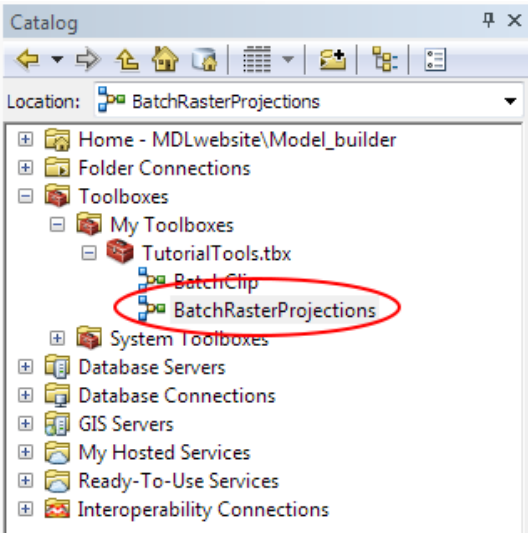


(Note: You may have noticed that the *Calculate Value* tool changed to white. This is because the precondition was set to another variable that is not ready until it receives user input.)

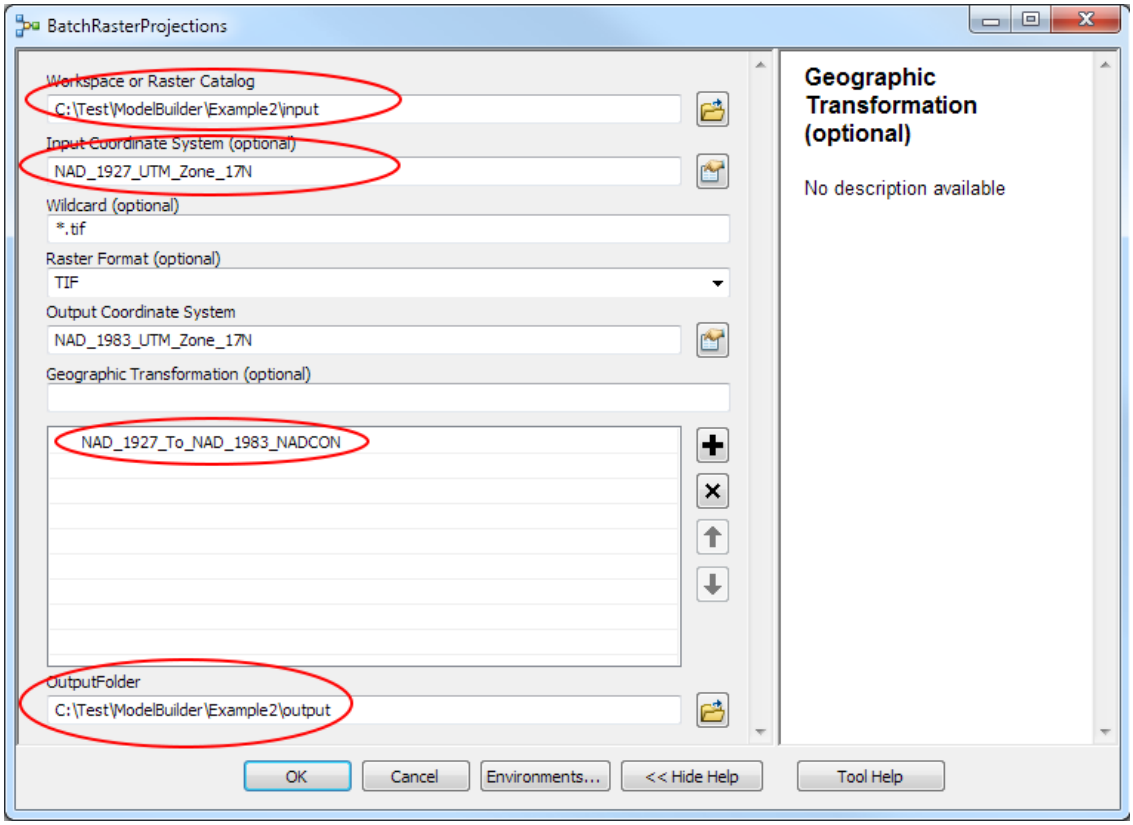
37. The model is ready to run. Make sure you save it first, and then close the model edit window. You can also give it a more meaningful name by right clicking on it in the Catalog list and selecting **Rename.**



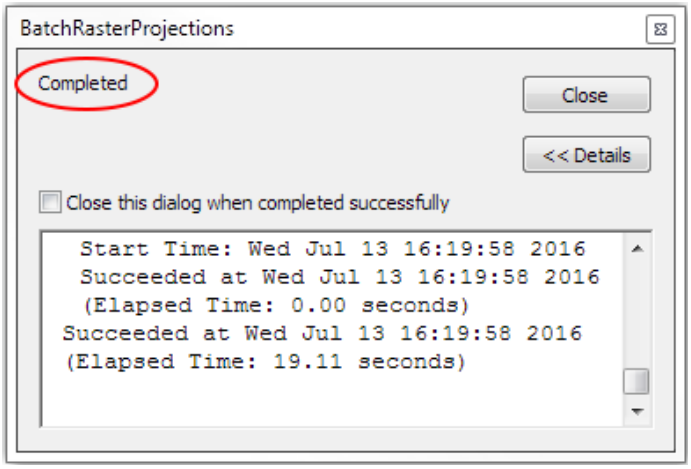
38. To run the model, double click on it in the toolbox directory on the right (in this example, the model is named **BatchRasterProjections**):



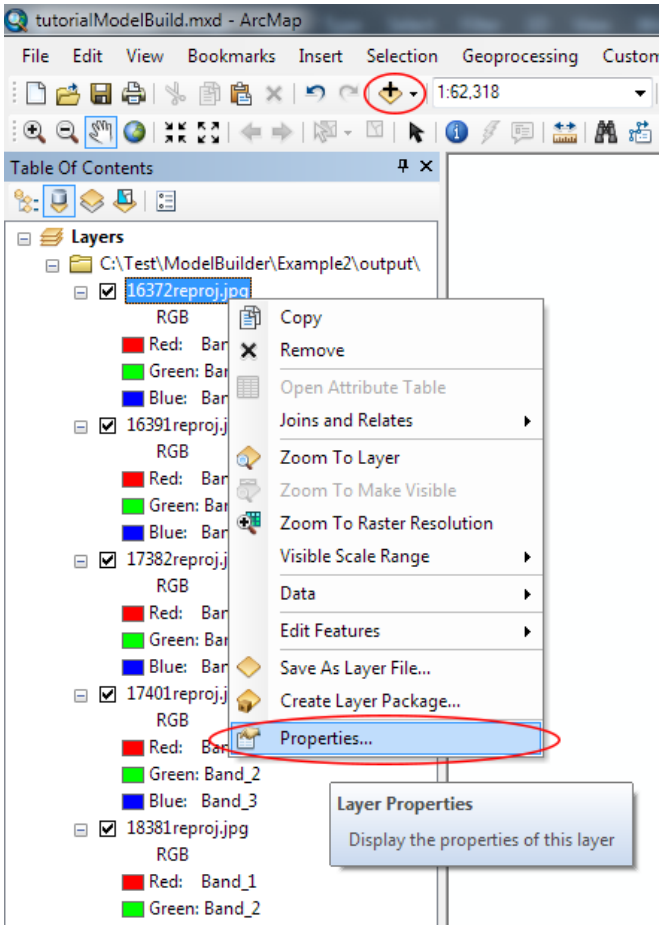
39. When you run it, the model should first prompt you for information. Fill in any variables with a green dot next to them, in this case you will need to browse to the input and output directories. You can also make any changes to the inputs that already have default values set, such as the Raster Format. (*Note: The *Input Coordinate System* field may be empty, even if you have filled in a default. This is a bug. You may have to re-enter this information.) When you have filled in the required information, click on **OK** to run the model.



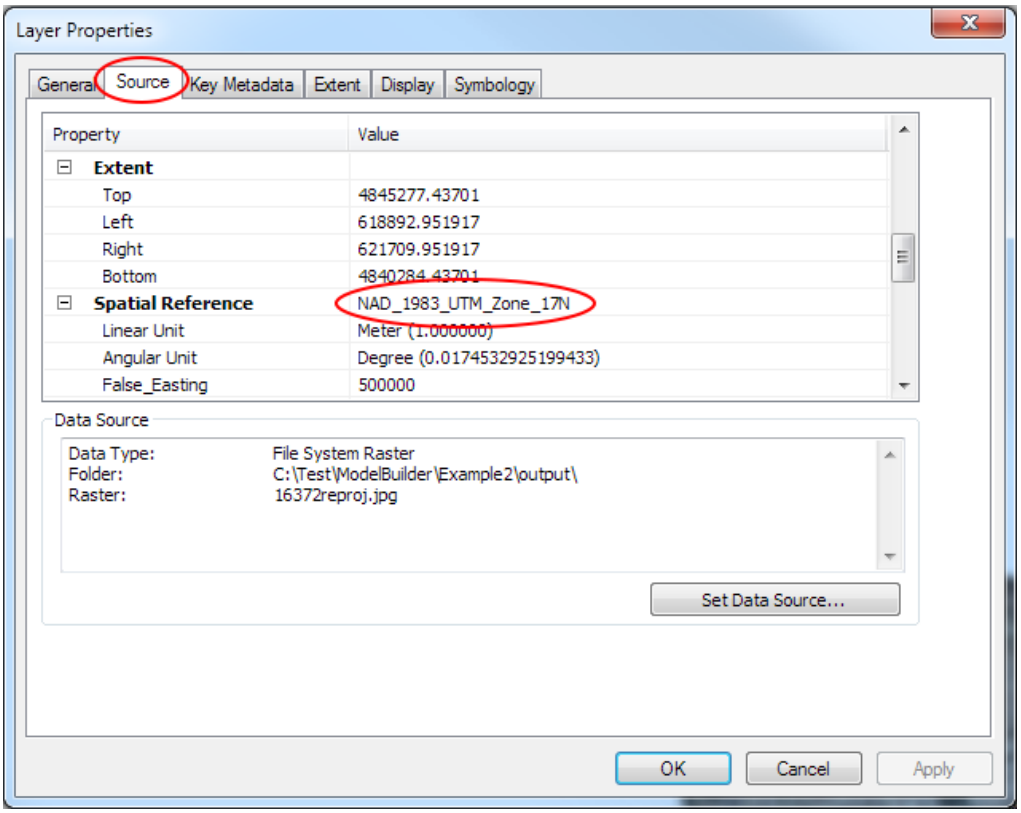
40. While the model is running it will show a status window. This status window also acts as a log and can give you helpful information to debug your model if it is not working. When the model has finished running, it will say “Completed.”



41. You should now have six jpeg files in the C:\Test\Example2\output directory. You can check that the model ran correctly by adding the jpeg files as layers in ArcMap, using the icon, and looking at their projection information. To look at their projection information, you would right click on the added jpeg file and select **Properties**.



42. From the **Source** tab, if you scroll down to *Spatial Reference*, you should see NAD_1983_UTM_Zone_17N.



If you have any questions or comments, please contact us at gis.maps@utoronto.ca.