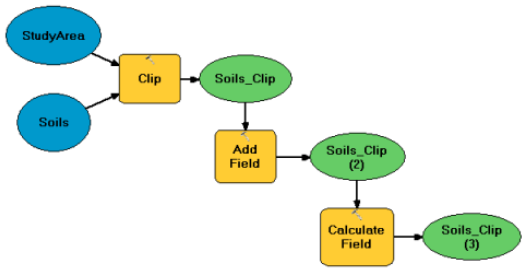
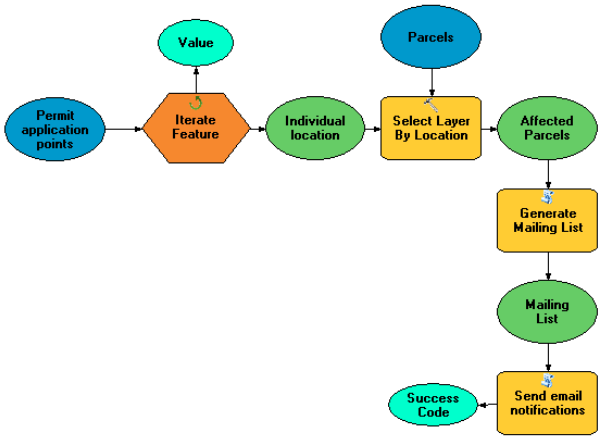
GE3238 GIS Design and Practices
Semester 2, AY2024-2025
Handout: 3 Mar 2025, Due: 12 Mar 2025 (Wed), 23:59

Lab03: ModelBuilder and ArcGIS Customization

ModelBuilder is an application you use to create, edit, and manage models. Models in the context of this course are **workflows** that string together sequences of geoprocessing tools, feeding the output of one tool into another tool as input. ModelBuilder can also be thought of as a visual programming language for building workflows.



While ModelBuilder is very useful for constructing and executing simple workflows, it also provides advanced methods for **extending ArcGIS Pro functionality** by allowing you to create and share your models as tools. ModelBuilder can even be used to **integrate** ArcGIS Pro with other applications. An example is provided below:



The above model is used by a municipality to send e-mail notifications to all addresses within 1 mile of an address for which a building permit application is filed. The model starts with a feature class of multiple permit application point locations. This feature class is fed into an iterator that loops over each individual point and feeds the point into the Select Layer By Location tool, where all addresses (parcels) within 1 mile of the point are selected. These addresses are then passed to a custom script tool (one that you or your colleague created), Generate Mailing List, that executes Python code to output a mailing list in HTML format. Finally, the mailing list is fed to another custom script tool, Send Email Notifications, which runs a custom executable that sends e-mail notifications and produces a success code.

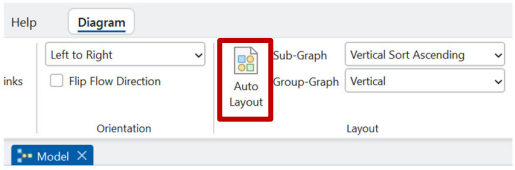
Task 1: Executing tools in ModelBuilder

1. Download the lab03data.zip from Canvas and unzip it in your working folder. Start ArcGIS Pro and open the Lab03_ModelBuilder.aprx.

2. Click the ModelBuilder button in the Geoprocessing group under the Analysis ribbon. This opens the ModelBuilder window for editing. This step creates a new (empty) model in ArcGIS Pro.

3. To assemble a workflow using existing ArcGIS tools, you need to add these tools into the ModelBuilder canvas. In this step you will add *buffer* and *clip* into your model.
    a. In the same Analysis ribbon, click Tools to open up the Geoprocessing pane.
    b. Find the Buffer tool, which is part of the Analysis Tools, and drag the buffer tool onto the ModelBuilder canvas in the white space. This adds the tool and the output data variable to the model. The output variable is connected to the tool by a connector. Both the tool and output data are empty (i.e., without color) as none of the tool parameters have been supplied.

4. Add the Clip (Analysis Tool) (Clip (Analysis) in a similar manner. If the two tools overlap, click the Auto Layout button under the Diagram ribbon.



5. Now that the tools are added to the model, you will fill in tool **parameters**.
    a. In ModelBuilder, double-click the Buffer tool to open its tool dialog box. You should be familiar with the parameters presented there.
    b. For the Input Features parameter, click the browse button and navigate to the input geodatabase Lab03.gdb. Choose the PlanA_Roads feature class and click Add.
    c. Once you fill in the input features, the Output Feature Class parameter is automatically generated and filled in for you. Replace this autogenerated output name with BufferedFC in Output.gdb (\Scratch\Output.gdb), and then click Save.
    d. For the Distance parameter, choose the Field option and choose the Distance field from the drop-down list. Click OK.
        - Note that the input data is added as a variable to the model (the blue oval) and is automatically connected to the Buffer. The input variable (the blue oval), the tool (the yellow rectangle), and the output variable (the green oval) change colors to indicate that all parameter values have been supplied and the tool is ready to run.
    e. Now add Clip tool into your model and double-click it to open its tool dialog box. For the Input Features parameter, click the Browse button, navigate to Lab03.gdb, and choose the vegtype feature class and click Ok.
    f. For the Clip Features parameter, click the arrow and choose BufferedFC from the drop down list (note that it is part of the model variables). Click the browse button for the Output Feature Class parameter, navigate to the output

geodatabase \Scratch\Output.gdb), type ClippedFC for the dataset name, then click Ok.

g. Click OK on the Clip tool dialog box. The output variable of the Buffer tool is automatically connected using a connector as an input to Clip.

6. After all the parameters of the tools have been filled in, the model is ready to run.
    a. Before you run the model, select the ClippedFC variable, right-click, then click the Add To Display option. Setting this option ensures that when the model is run from ModelBuilder, the output ClippedFC will be added to the default map in ArcGIS Pro for quick viewing of the results.
    b. From the ModelBuilder ribbon, click Run in the Run group. The model runs and the output is added to the display. After the model has finished running, both the tools (the yellow rectangles) and output variables (the green ovals) show a drop shadow around them indicating that these tools have been run.

7. Now, assume you wish to run the model on different roads (such as PlanB or PlanC feature classes), you have two easy routes: (1) Double-click Buffer and navigate to another feature class, or (2) Drag the data from the Catalog pane onto the ModelBuilder canvas to create a new data variable, then connect this variable to Buffer. Follow the steps below to see how this works.
    a. Choose the PlanB_Roads feature class and drag it onto the ModelBuilder.
    b. Click on PlanB_Roads. Hold the mouse button and move to Buffer before releasing the button. A shortcut menu appears containing the choices of available parameters for the tool. Choose the Input Features parameter from the list.
    c. Click the Select icon (an arrowhead cursor). Double-click Buffer and choose the correct distance field from the PlanB_Roads feature class.
    d. Run the model by clicking the Run button. The output should be added to the display after model completes.
    e. After the model has finished running, both the tools and output variables show a drop shadow around them indicating that these tools have been run. The existing output ClippedFC layer (from the previous run with PlanA_Roads) in the ArcGIS Pro Contents pane is updated with the results from PlanB_Roads.

8. Saving is the last step for this tutorial.
    a. Click the Save tool on the ModelBuilder toolbar and go to your working folder.
    b. A model can only be saved in a toolbox. Each ArcGIS Pro project comes with a default toolbox Default.atbx. It is the place your first model is stored. You could alternatively choose to create your own toolbox in the output.gdb to save your model.

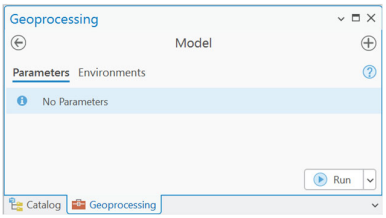
Q1) Assume you wish to obtain the total area affected by roads in plan B in ModelBuilder, how will you change the model to do so? Please save your proposed solution as a new tool named AreaB in myTools.tbx. (2 marks)

c. Switch the input variable from PlanB_Roads back to PlanA_Roads.

Q2) If you were to identify areas affected by all three road proposals (PlanA_Roads, PlanB_Roads, and PlanC_Roads) in one model, how would you go about it? Please save your proposed solution as a new tool named TotalArea in myTools.tbx. (2 marks)

Task 2: Creating tools with ModelBuilder

1. Double-click the model in the Catalog pane. The tool dialog box opens but shows no parameters, as illustrated below.



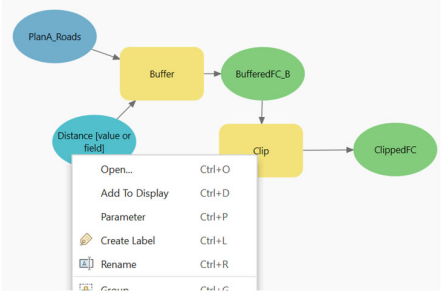
2. If you run the tool by clicking Run, the model will run. The output of the model (ClippedFC) will **not** be added to the Contents pane, even though Add To Display was checked for the output variable. The reason is that when a model is run from its tool dialog box, the Add To Display setting is ignored. To add the output to the display, you must make the output variable a model parameter.

   Another reason for creating model parameters is that you want to run the model with different inputs without having to open ModelBuilder every time.

3. Once the model parameters have been created, you can execute the model from its tool dialog box, supplying different values for the Input Roads, Buffer Distance, Input Vegetation, Output Clipped Feature Class, and XY Tolerance parameters. You don't have to open ModelBuilder each time you want to run the model—you can use the tool dialog box instead. An added advantage of running a model from its tool dialog box is that a result in the Results window is created; running a model within ModelBuilder does not create a result in the Results window.

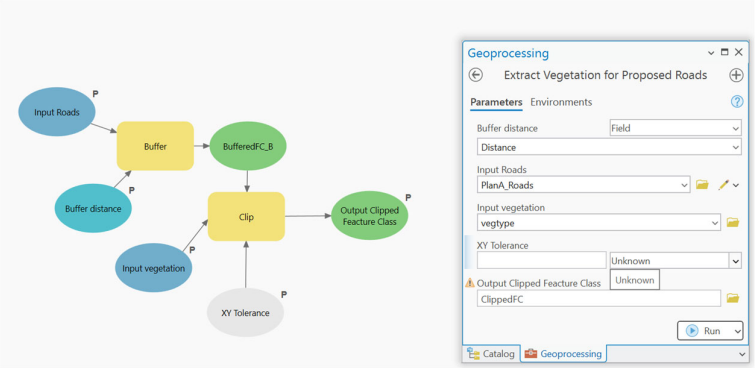
The remaining steps will expose model variables as parameters, so they appear on the model tool dialog box as shown above.

4. To define the Buffer Distance and XY Tolerance parameters, you need to create model variables for them.
   a. In the Catalog pane, right-click the model and click Edit. This opens the model in ModelBuilder.
   b. Right-click Buffer. Click Create Variable > From Parameter > Distance [value or field]. This adds the Distance parameter as a variable in the model.
   c. Right-click Clip. Click Create Variable > From Parameter > XY Tolerance.

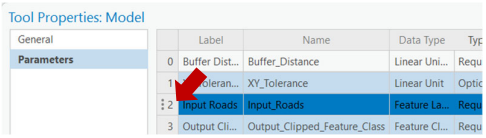


5. Now that you've created variables for Distance and XY Tolerance, you are ready to make model parameters.
    a. Right-click Distance [value or field] and check the Parameter option. The letter **P** appears beside the variable, indicating it is a model parameter. This model parameter then also appears on the model tool dialog box.
    b. Create model parameters for the following variables: (1) PlanA_Roads, (2) vegetype, (3) XY Tolerance, and (4) ClippedFC.

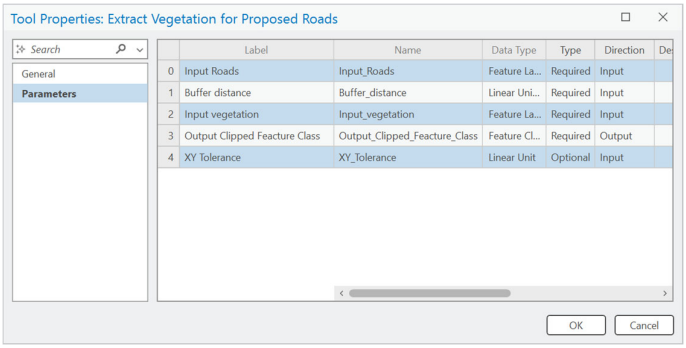
6. ModelBuilder assigns default names to variables. Variable names are used for parameter names on the model tool dialog box. It is good practice to rename variables, especially if they are model parameters.
    a. Right-click PlanA_Roads and click Rename.
    b. Type Input Roads and click OK. The name of the variable changes to Input Roads.
    c. Rename the remaining variables as follows:
        - Rename Distance [value or field] to Buffer Distance.
        - Rename vegetype to Input Vegetation.
        - Rename ClippedFC to Output Clipped Feature Class.
    d. Save the model. You do not need to exit ModelBuilder.
    e. Double-click the model from the Catalog pane to open the model tool dialog box. The model tool dialog box should look similar to the illustration below. The order of the parameters may be different on your dialog box, but this is not an issue since you'll change the order below.
    f. Close the model tool dialog box.



7. As illustrated above, the order of the parameters is not ideal. The standard practice is to order parameters as follows: (1) Required input datasets, (2) Other required parameters that affect tool execution, (3) Required output datasets, and (4) Optional parameters.
    a. In ModelBuilder, click Model > Model Properties. Click the Parameters tab.
    b. Choose the Input Roads parameter and move it to the top. This can be done by clicking on the row header and drag it to the top.



    c. Change the position of other parameters as shown below: Keep the dialog box open after you're done. You will need it for the next two steps.



8. Once the model parameters are set in the correct order, change the type of parameter. If a parameter is a required parameter of a tool in the model, you will not be able to change the type to optional from these settings.
    a. Click the cell under the Type category for XY Tolerance. A list appears with two options. For this example, keep XY Tolerance as an optional parameter and the rest as a required parameter type.

9. You can restrict the type of input to any parameter by applying filters to parameters. The model in this example expects the Input Roads parameter to be line features. In the following steps, the parameter is modified by applying a filter so that it only accepts line features.
    a. Click Model > Model Properties. Select Parameters from the left of the dialog box.
    b. Choose Input Roads and click the cell under the Filter category. A Feature Type Filter box pops up.
    c. Uncheck all the types except Polyline and click OK.
    d. Click OK on the Model Properties dialog box to apply the filter.

When you run a model, output data is created for each process in the model. Some of the data created is of no use after the model is run since it was only created to connect to another process that creates new output. Such data is called intermediate data. Note that all outputs except the final output, or those that have been made model parameters, are automatically made intermediate data in the model. In this example, the output of the Buffer tool is only useful as an input to the Clip tool and is not used after that, so the Intermediate option is checked.

10. You can set the model name, label, and description for the model.
    a. Right click on the Model and select Properties from Tool Properties box.
    b. Click on the “General” tab.
    c. Type ExtractVegetationforProposedRoads for the name of the model. No spaces are allowed in the model name.
    d. Type Extract Vegetation for Proposed Roads in the Label text box. Spaces are allowed in the model label. This label is used to display the model name in the Catalog window.
    e. Type the desired text in the Description text box.
    f. Save the finished model and exit ModelBuilder.

11. Double-click the model from the Catalog pane to open the model tool dialog box. Since the model was saved with the predefined values, all the parameters on the dialog box are filled in. You can change the value of any parameter here by entering a new value. Click OK to run. The final output of the model (Output Clipped Feature Class) is added to the display by default, and the model messages are posted in the Results window.

Q3) Describe a practical case in which ModelBuilder can improve workflow efficiency. You may use your final project as an example, but members of a group should provide independent answers. (3 marks)

In the next part of the lab assignment, we are going to build a model that dynamically processes user input through a few geoprocessing tools and output some results with a set of predefined symbology. Specifically, we are going to simulate school query based on a location a user clicks on a basemap. The experience is similar to the school query on onemap.sg.

Task 3: Prepare to build a model in ArcGIS

Before we start, we need to think about the processes and tools we need to derive the necessary result. Conceptually, we will need a layer of schools to begin with, and a point layer representing the location that the user clicked. Using this point, we can run a buffer of 1km and 2km, then use this buffer polygons to perform an intersect operator with the layer of schools. Finally, we need to design the symbology to present the results in a similar fashion.

Let's begin by creating the feature class schema to contain the point data that the user will click on.

1. Start a new empty map in ArcGIS Pro and name it as Schools.

2. In the Catalog pane, right click on output.gdb, select New>Feature Class.

3. Select the type of feature to be stored as Point Features and name it UserInput, then click Next.

4. No attributes are required for now so you may click Finish to create the feature class.

5. Accept the default XY Tolerance (i.e., 0.001) and resolution, then click Next.

6. Search for SVY21 and select the SVY21 Singapore TM projected coordinate system, then click Next.

7. Accept the default configuration keyword and click Next.

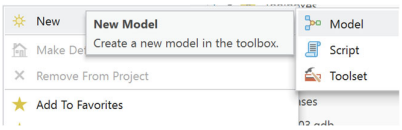
We now have an empty UserInput point feature class. However, enabling interactive input means an users will digitize data on the spot, and thereby an editing task. In ArcGIS Pro, to do so, you will need to use a layer along with variable (more instructions below).

8. To create a layer from UserInput, right click the UserInput layer in the Contents pane, select Sharing, then Save as Layer File in your working folder. Name it UserInput.lyrx or whatever file name that you can recognize easily.

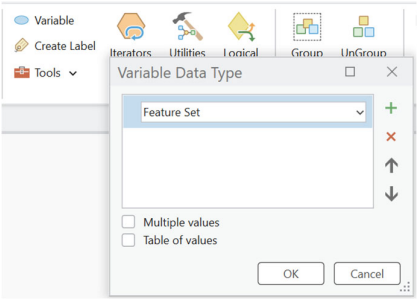
At this point, while a layer meant for storing symbology, i.e., how a layer should be displayed on a map, and it can be used to apply a consistent symbology across multiple layers, you could also use it as a template or a filter, as you will see later in the next Task.

Task 4: Build a model

1. In the Catalog pane, right click on the Default.atbx, select New > Model.

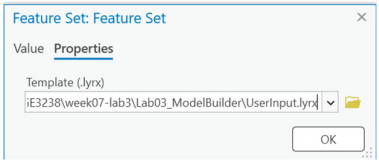


2. From the ModelBuilder ribbon, select Variable. In the next dialog box, first browse through all available variable types and then select Feature Set.



   The feature set is meant to support interactive feature input capability[1].

3. Double click on the newly created Feature Set variable. In the next dialog box, browse to UserInput.lyrx in your working folder to be import as schema and click OK. To recognize it easily, rename the feature set to User Input.

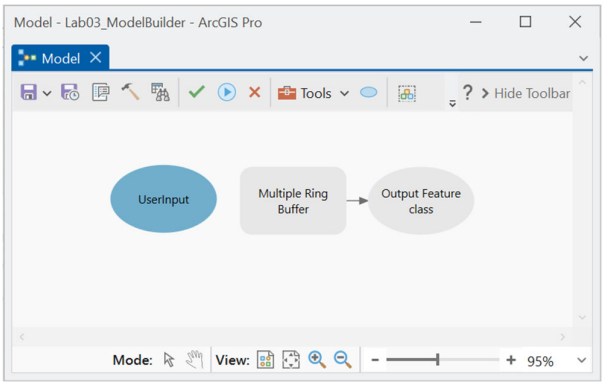


From this point we need to create our 1km and 2km buffers.

4. Search for the multiple ring buffer (analysis) tool in the Geoprocessing Search window.

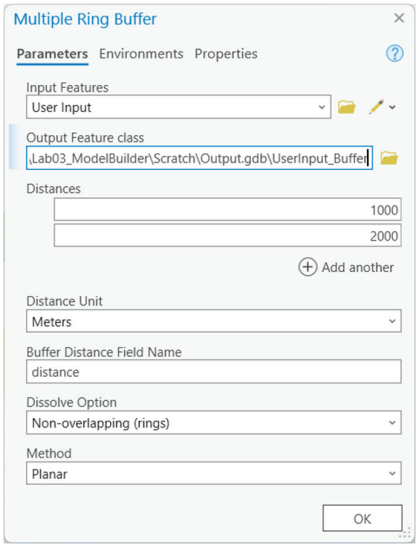
5. From the Search window, drag the Multiple Ring Buffer (Analysis) tool into ModelBuilder.

[1] https://pro.arcgis.com/en/pro-app/latest/help/analysis/geoprocessing/basics/interactive-feature-input.htm



6. Using the Connect tool, draw an arrow from the Feature Set variable (i.e. User Input in the screen capture above) to the Multiple Ring Buffer tool, and select it as the Input Feature when prompted.

7. Double click on the Multiple Ring Buffer tool, set the output to store as Buffers feature class in output.gdb, add 1000 and 2000 as buffer distances, set the unit to Meters, then set the rest of the parameters as shown:



8. Once parameters are set, click OK.

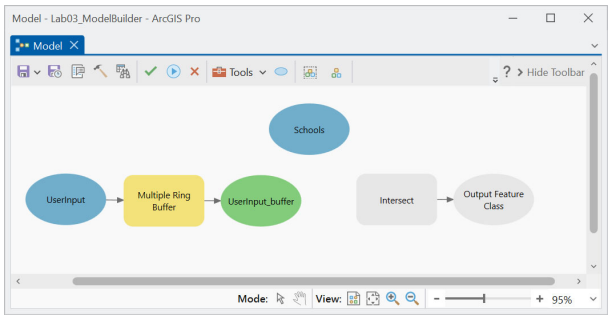
9. For the Output Feature Class, you need to specify a feature class name even if it is an intermediate data. Double click on the Output Feature Class oval to provide the place where this feature class is temporarily stored.

Noticed that when variables and processes are ready to run, they will change from no fill to a colour fill.

10. In the Search window, search for the intersect (analysis) tool.

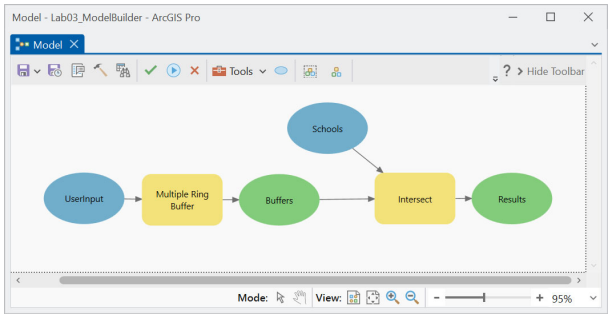
11. From the Search window, drag the Intersect (Analysis) tool result into ModelBuilder.

12. From the Catalog pane, drag the Schools feature class from Lab03.gdb into ModelBuilder.

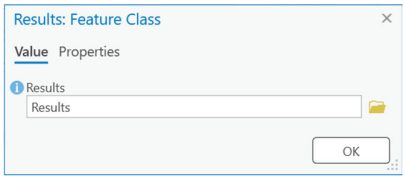


13. Using the Connect tool, draw connecting lines from both the Buffers and Schools variables respectively to the Intersect tool, and select them as Input Feature when prompted.

14. Right click on the output variable of the Intersect tool, select Rename and rename it as Results, then click OK.

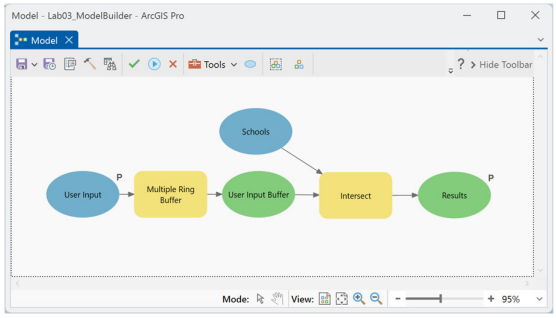


15. Double click on the Results variable and set it to Results feature class in output.gdb and click OK.



16. Right click on the Feature Set variable, select Model Parameter.

17. Set both the Feature Set variable and the Result variables as model parameters. Further, enable Add To Display for the Results variable.

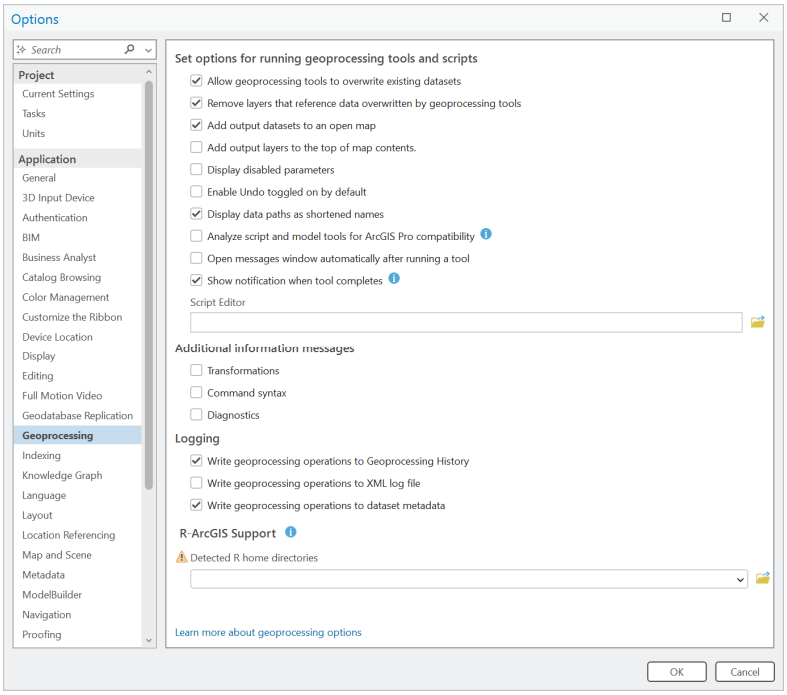


The above shows how your model should look like up to this step. Notice the letter P on the top

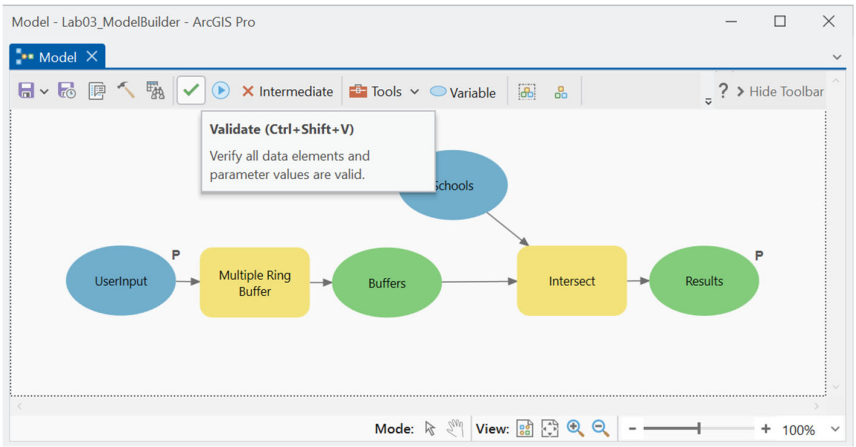
right of any variable that is set as a Model Parameter. This allows users to interactively change these items in the user interface when executing the model. Save your model before you proceed.

Task 5: Test run the model

1. Before testing your model, it would be useful to take a look at the options available for Geoprocessing and ModelBuilder. To do so, click on Project on the ArcGIS Pro menu and select Options. The screen capture of Geoprocessing is shown below. Some of the options may streamline your workflow when the tool has to be run multiple times, e.g., overwrite existing datasets, while others may improve clarity of your model, e.g., (unchecked) Display disabled parameters or Display data paths as shortened names.



2. From the ModelBuilder menu, select Validate to check if the model is sound.



3. If there is no error, from the menu, select Run Entire Model.

Observe that there is a warning of empty output generated, which is normal as there was no point data to buffer with in the first place. There is also no option provided for you to interactively click a point on the map when a model is executed from ModelBuilder.

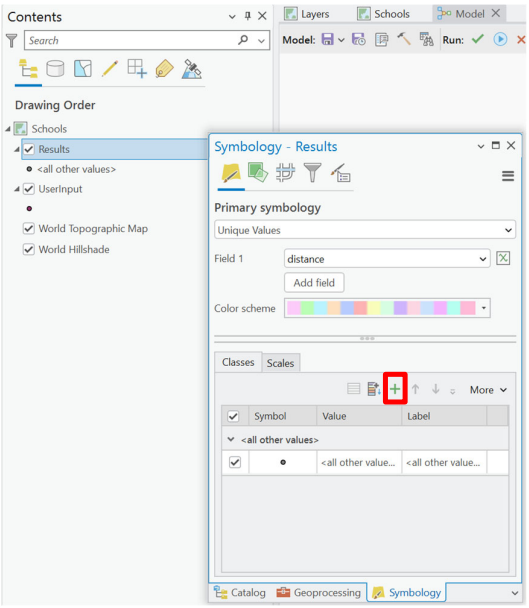
4. Close the message box. Also observe that once a model has been executed, there will be 'shadows' created behind the output variables and processes. This indicates that there is data created from the execution of the model.

Recall that the output Results variable was set to Add To Display, and thus it now appears in the Contents pane on the left of the ArcGIS Pro window though with nothing to render.

Task 6: Prepare symbology

Ideally, the schools returned for 1000 and 2000 metres should be shown in two different colours. We need to prepare a similar symbology in a layer file for use in ModelBuilder.

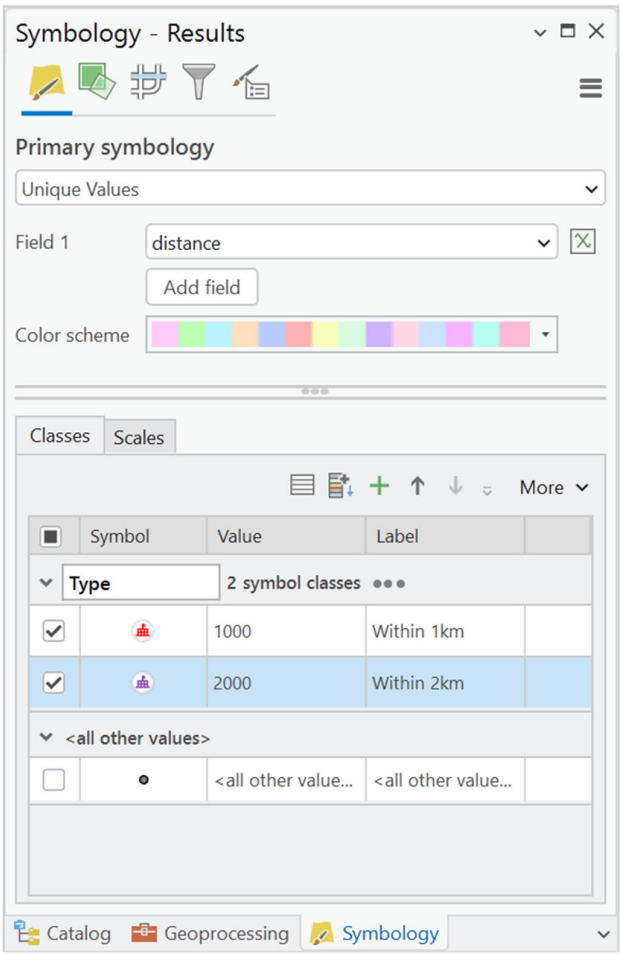
1. In ArcGIS Pro, change the symbology of the Results layer to unique values by distance.



2. You will need to manually add 1000 and 2000 metres as right now the layer UserInput has no records. To do so, click the "Add unlisted value" (see the button highlighted by the red box above). Please explore the dialog box a bit to see how you could come up with a result similar to the screen capture below.

   Note the following three cosmetic changes to improve accessibility of the analysis results: (i) <all other value…> is unchecked, (ii) the heading 'distance' is replaced with 'type', (iii) the labels were changed to the ones more accessible to human, and (iv) the graphic symbol representing schools and the colors to differentiate schools within different radii.

   These changes will be reflected in the Contents pane as well.

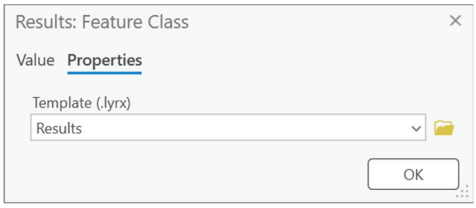


3. In the Contents pane, right click on the Results layer, select Sharing and then Save as Layer File, and save it as Results.lyrx under your Lab03 ModelBuilder folder.

   Note: A Layer file should be saved into a folder and not into a geodatabase!

4. In ModelBuilder, double click on the Results variable and select Properties.

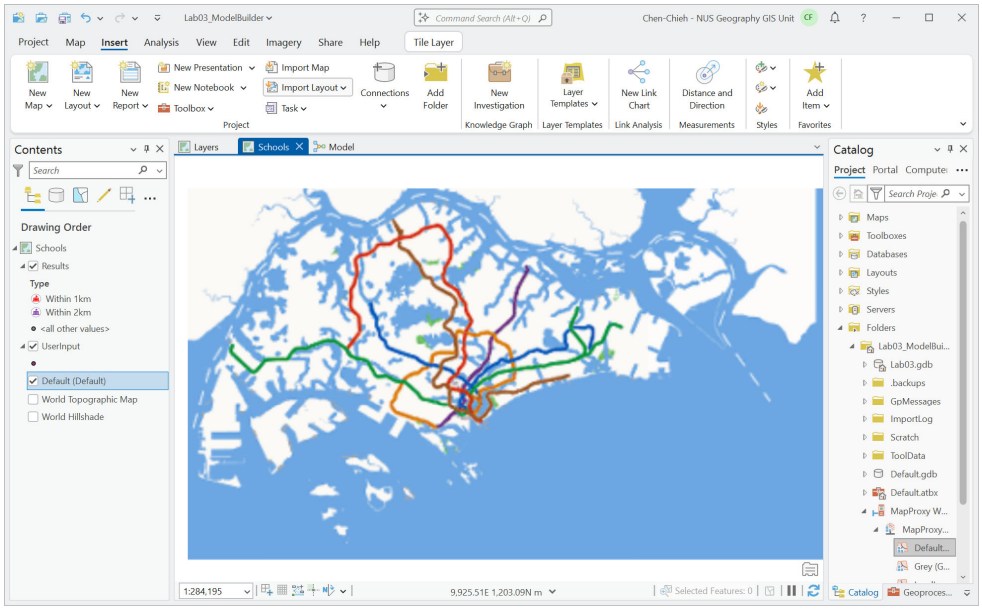
5. In the Results Properties window, select Results layer file that you have created. You could alternatively load the Results.lyrx directly from your working folder. Click OK.



6. Save your model and close ModelBuilder.

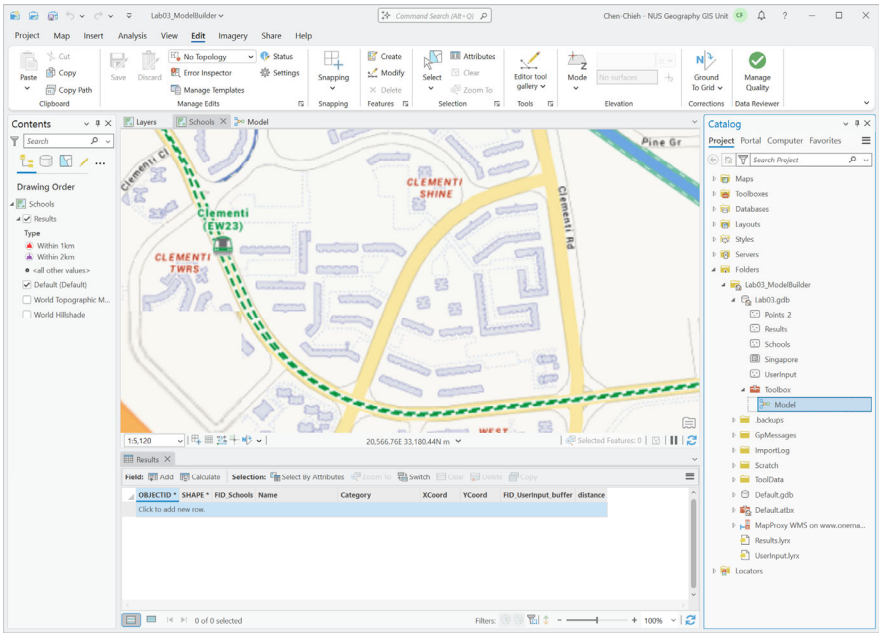
Task 7: Verify the effect

1. Using knowledge from previous lab assignments, make a connection to OneMap's basemap using WMTS.

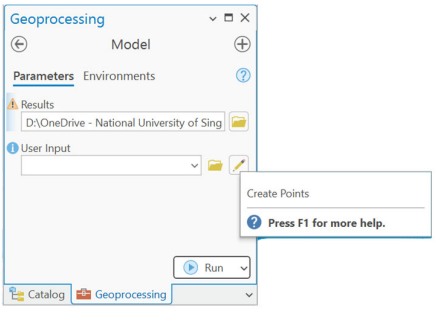


2. Zoom in to see buildings in any housing estate of Singapore, for example, Clementi. Keep in mind that the basemap is now served directly from the Singapore Land Authority's OneMap platform. Observe the speed of maps loading from the Internet.

   Note: Usage of OneMap basemap is entirely for aesthetic purposes and is not technically required. If rendering of the basemap is slow, please remove the basemap and use the supplied Singapore polygon feature class in the Lab03.gdb instead.

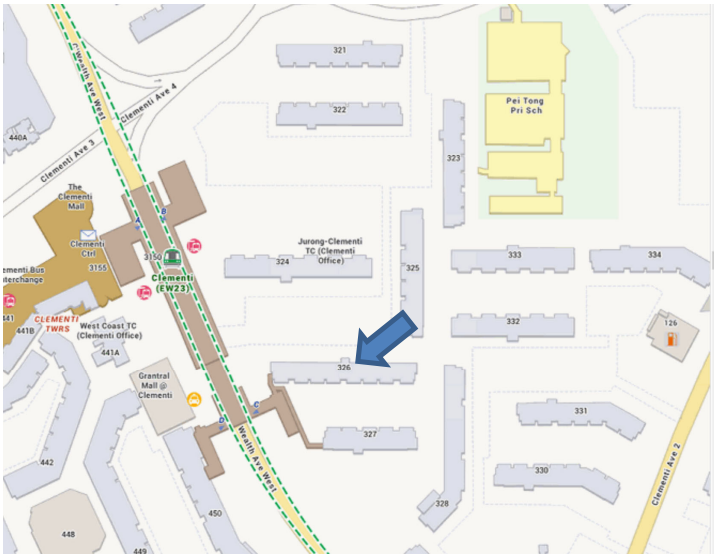


3. From the Catalog pane, double click on your model from Lab03.gdb>Toolbox. Note that for User Input (which is the name of the feature set), you have an option to edit the feature class or layer.



Click on the Create Points button. ArcGIS Pro would generate a new feature class layer for accepting user input. Given that a layer (UserInput.lyrx) is used as the template for user input, the system knows that the input has to be a point layer.

4. Click on any building, for example, block 326 beside Clementi MRT Station.



5. Click Run in the model interface.

6. Close the Model dialog box, zoom out slightly on the map to observe the results.

Note: The results we have may not be exactly the same as what you get from OneMap as the underlying set of school data supporting the operation is different.

- ***Q4) Assume you would like to provide a Find Buildings Near a School function. Describe the key modifications required to allow your current model to find buildings in the proximity of a school instead. Clearly state all your assumptions. (3 marks)***

That is it for lab03.

For your information, in November 2024, Esri made ModelBuilder available on ArcGIS Online. This means that customized tools can now be hosted in Esri's cloud platform, specifically with the map viewer. According to Esri's blog (https://www.esri.com/arcgis-blog/products/arcgis-online/analytics/introducing-modelbuilder-in-map-viewer-beta-release/), ModelBuilder is "designed for users who prefer a no-code approach." While we know about this point as a difference between ModelBuilder and Python from the lecture, the development is nonetheless exciting as you now have the chance to *customize* in a cloud environment, and this new capability may be useful for some final projects.

Reminder:
Please submit the following files to Canvas **Lab03 Assignment**. Your files should name should follow the **last_name-first_name-matriculation_number.zip** convention.

(1) Typed answer in Word document;
(2) All files (geodatabases, project files, etc) in your working folder.

Late penalty applies (10% per day).

End of lab03 instruction.