

Creating CN Grid from NLCD Landuse and SSURGO Soil data

Prepared by
Venkatesh Merwade and Chung-Yuan Liang
Lyles School of Civil Engineering, Purdue University
vmerwade@purdue.edu

Introduction

The objective of this exercise is to process the National Land Cover Dataset landuse and SSURGO soil data to create curve number (CN) raster using ArcGIS Pro. Curve number is used in many hydrologic models for computing rainfall losses using the NRCS method.

Learning Outcomes

- 1) Create a geodatabase and export/import feature classes between geodatabases
- 2) Clip vector data for a given watershed or area of interest
- 3) Link feature class and table via join/relate in ArcGIS
- 4) Perform attribute query and field operations on vector data
- 5) Classify raster and perform mathematical operations on raster

Data

The data needed for this exercise is provided to you on Brightspace as a zip file, but the instructions for getting the data are provided on Brightspace. You can also get the data from the following link:

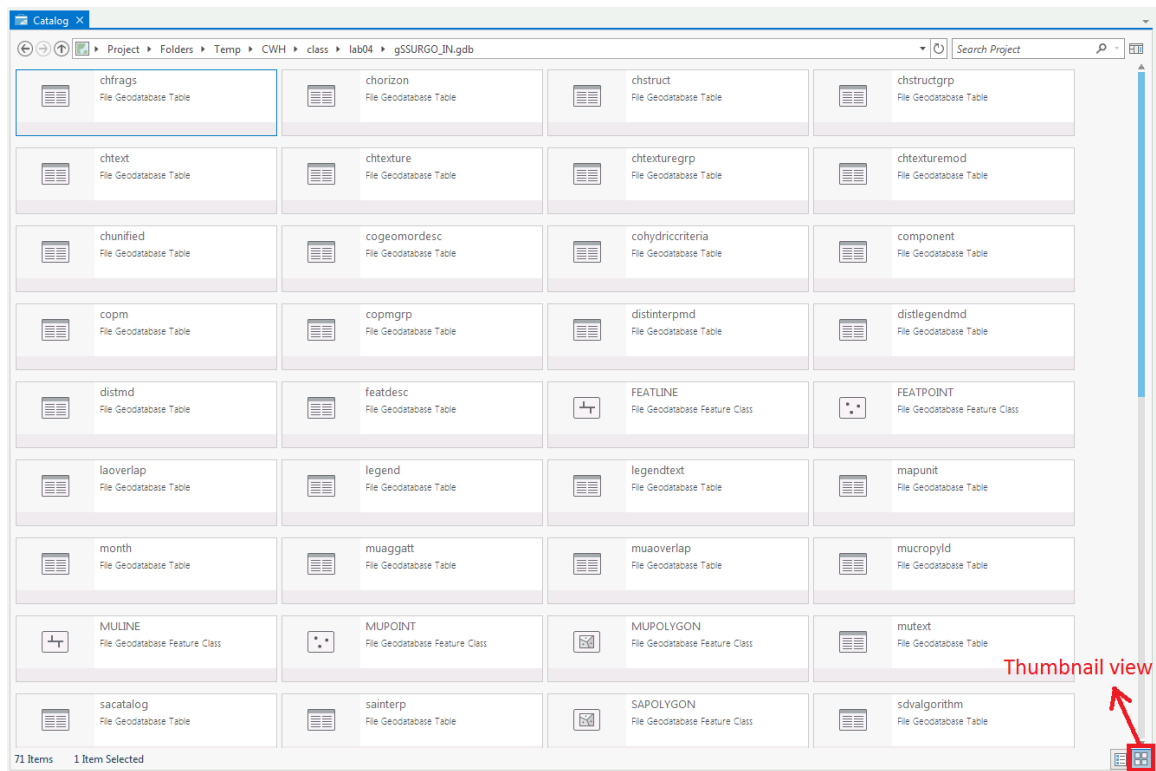
<https://app.box.com/s/2ovaawlt69i1cnicy575xv982bvn771j>. The gSSURGO data are usually provided for each state, but data provided with this tutorial is slightly modified to only include the study area of Cedar Creek.






Computer Requirements

You will need ArcGIS Pro 2.x to complete this exercise.

Instructions

Create a folder called module6 or cngrid. Copy the zip file (data) in this folder and unzip it. You will notice that it is a file geodatabase (and quite large). So, let us explore it in Catalog as shown below (thumbnail view). You can see the Catalog view by opening an empty map in ArcGIS Pro and connecting to the folder where the data is saved.



You will see that the original database has more than 60 tables (), few feature classes ( or  or ), one raster and a bunch of relationship classes () between the features and tables and between tables and tables.

Note: You may see less as some tables are removed to keep the database size small.

The most important feature for us in this database is the **mapolygon (map unit polygon)** feature class. A map unit is the basic geographic unit (an area) in the soil survey database with a unique soil name defined in terms of its components. Each map unit has a unique symbol on a soil map. Look at the names of the relationship classes and you will see that component table (which forms the map unit) is related to many tables. Open ArcGIS Pro and save the project as “*cmgrid.aprx*” in your working folder (module6).

Add MUPOLYGON feature class to the map document and you will see the map units for only three counties surrounding the Cedar Creek boundary. These data are projected in **USA Contiguous Albers Equal Area Conic**. You do not need to change the projection. The soil survey is conducted at the **county level** and each county is called soil area in the gSSURGO database. So what you see on the map are soil map units for three soil areas (or counties). The gSSURGo data is available for the entire state, but these data are modified for this lab so we can work with it easily. Open the attribute table and see how many map unit (features) you have for these three areas. Indiana has more than 1.3 million map units. In the attribute table, you will see that there is not much information about the soil itself because all that data is stored in the tables that are related to this features via multiple relationships.

| OBJECTID | Shape | AREASYMBOL | SPATIALVER | MUSYM | MUKEY | Shape_Length | Shape_Area |
|----------|---------|------------|------------|-------|--------|--------------|----------------|
| 14442 | Polygon | IN039 | 4 | BuuA | 165330 | 939.629457 | 31492.671823 |
| 14541 | Polygon | IN039 | 4 | BaaB | 165316 | 4795.402004 | 119206.199897 |
| 14578 | Polygon | IN039 | 4 | BaaA | 165315 | 4995.10754 | 350816.653481 |
| 14839 | Polygon | IN039 | 4 | CvdA | 165351 | 6558.222501 | 272234.18493 |
| 15528 | Polygon | IN039 | 4 | ReyA | 165416 | 286.575927 | 4659.322397 |
| 15592 | Polygon | IN039 | 4 | CvdB | 165352 | 30930.361349 | 1479339.454913 |
| 15739 | Polygon | IN039 | 4 | WobB | 165528 | 1270.409711 | 49557.054654 |

The most important attributes in this table are **MUSYM** (map unit symbol or the name of the map unit) and the numeric key for each map unit, **MUKEY**. These two fields are **used to relate these geographic features to all the tables**.

Now that you know little bit about how the data are organized in gSSURGO, let us create a file geodatabase with just the information we need for hydrologic modeling. In our case, we will only use the hydrologic soil group to help us compute the curve number.

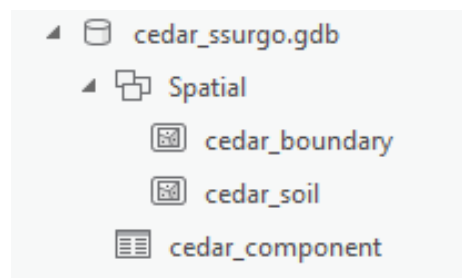
Organizing the Geodatabase

There are many feature classes and tables in the gSSURGO geodatabase. However, the most important data that we are going to use in this module only includes “MUPOLYGON” feature class and “component” table. To make our working space clear and easy to access, we create a new working geodatabase to store the data we need.

Follow the steps below to organize the geodatabase:

- 1) **Create** a **geodatabase** inside your working folder named *cedar_ssurgo.gdb*
- 2) **Create** a feature dataset (name *Spatial*) inside *cedar_ssurgo* and **assign** NAD 1983 UTM zone 16N projected coordinate system. You can ignore the vertical coordinate system or assign one if you wish.
- 3) In the *Spatial* feature dataset, **import** “*mupolygon*” (Map Unit Polygon) feature class from the *gssurgo* geodatabase (you can do this by right clicking on the “*Spatial*” feature dataset and click Import→ Feature class. Name the new imported feature class as “*cedar_soil*”). Alternatively, you can also export *mupolygon* from *gSSURGO_IN.gdb* to *Spatial* feature dataset within *cedar_ssurgo.gdb*.
- 4) Now import the “*component*” table from the *gSSURGO_IN* database to the *cedar_ssurgo* geodatabase (right click on the database, it should be outside “*Spatial*” feature dataset) by using the import option but use table instead of feature class. Name the table as “*cedar_component*”
- 5) Finally, import the *cedar_boundary* shapefile from your folder as *cedar_boundary* feature class inside *Spatial* feature dataset

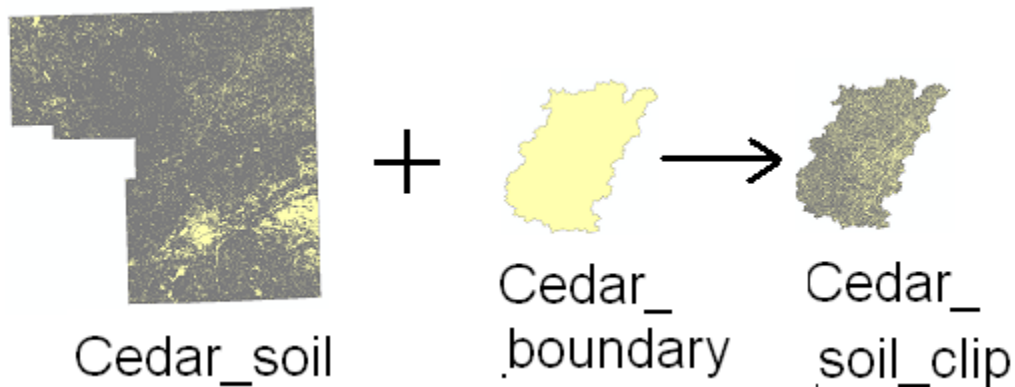
The *cedar_ssurgo* geodatabase including the spatial features and the component table should look like the figure below in the Catalog pane in ArcGIS.



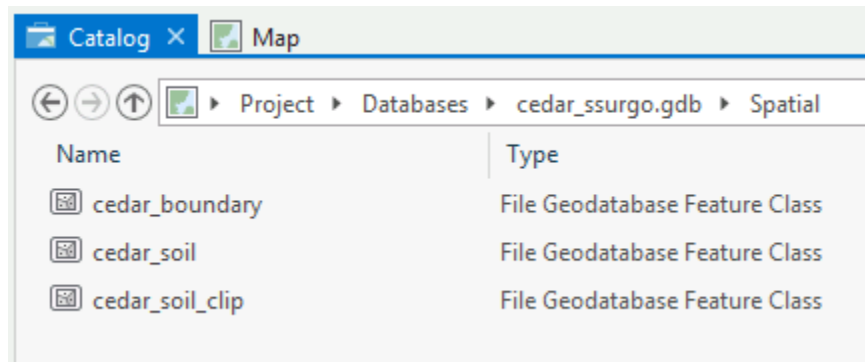
Clipping features

Next, clip the “cedar_soil” polygon feature class dataset with the “cedar_boundary” feature class. We can use Toolbox in the Analysis tab to accomplish the clipping task. As it is a feature class dataset, use the *Clip* tool in the *Analysis Tools* toolbox. With *cedar_soil* as input features, *cedar_boundary* as clip features, *cedar_soil_clip* as output feature class, save it inside the “Spatial” feature dataset within *cedar_ssurgo.gdb*.

Note: For raster dataset, you can use Clip in Data Management Tool → Raster → Raster Processing → Clip.




The Catalog view of the “Spatial” feature dataset is shown below:

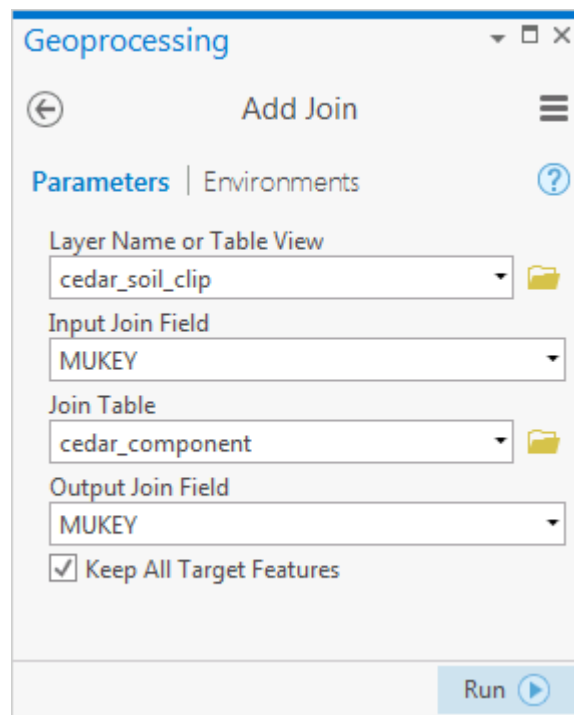


Preprocessing data in the tables


The spatial (*cedar_soil* or *cedar_soil_clip*) and tabular data can be joined/related by using common fields. For example, *cedar_soil_clip* and *component* can be linked by using the “MUKEY” field. Let us link the hydrologic soil group to all the polygons and create a map of Hydrologic group.

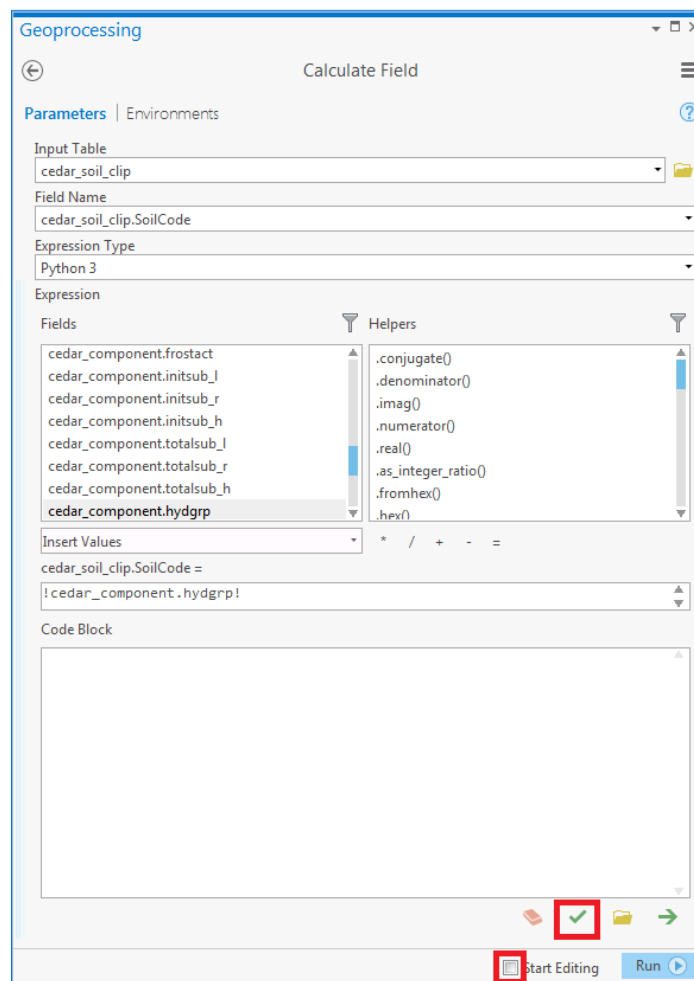
Create a field named “SoilCode” (Data Type: Text) in *cedar_soil_clip*. You can create a field in a feature class by opening its attribute table and selecting Add Field  Add . Remember to press Save edits in the top ribbon to add the new field.

The soil group data are available in the component table (in *hydgrp* field). Make sure that *cedar_component* table is added from *cedar_ssurgo.gdb* to the map document. Join the polygon features in *cedar_soil_clip* with component table through MUKEY field. **Right click** on *cedar_soil_clip* and **click** *Join and Relates* → *Add Join* Join component table to *cedar_soil_clip* by using the MUKEY field as shown below:



Click *Run*. You may get a message asking you to index fields. You can respond to this message by selecting either yes or no - it does not matter in this exercise because we will join the table only once.

After you create the join, **open** the attribute table of *cedar_soil_clip*, and you will see that the fields from component table are now available in *cedar_soil_clip* feature class. Now populate the SoilCode (or *cedar_soil_clip.SoilCode*) field in *cedar_soil_clip* by equating it with *cedar_component.hydgrp* field. With the attribute table open, **right click** on *SoilCode* field to open the field calculator and then equate *SoilCode* to **cedar_component.hydgrp** (alias name Hydrologic Group) by **double clicking** this field as shown below. Click on verify  to ensure the expression is valid. If asked, make sure to press the Save button to confirm the edits.



If there are rows in component with “Null” values (which is the case for this dataset), you may get an error message saying the values are too large for the field. Just ignore this message and continue. After the calculations are complete, you should see *cedar_soil_clip.SoilCode* populated with letters A/B/C/D. Now **remove** the join (right click on the *feature class* → *joins and relates* → *remove all joins*) and **save** the project.

Converting soil hydrologic group to an integer field

Our goal is to have the soil data in a raster format so that both landuse and soil rasters can be analyzed to create curve numbers. To create a soil hydrologic group raster, we need to convert the soil hydrologic group (A-D) to numbers (1-4).

We now have the soil data with the SoilCode field populated with hydrologic soil group from the component table. As you see in the attribute table of cedar_soil_clip, the SoilCode field is populated, but many rows have NULL values. In practice, you can use your field knowledge or judgment to replace these NULL values with appropriate groups. One easy option is to replace NULL with the dominant soil group in the watershed. In this case, we will replace the NULL values with "C" soil hydrologic group.

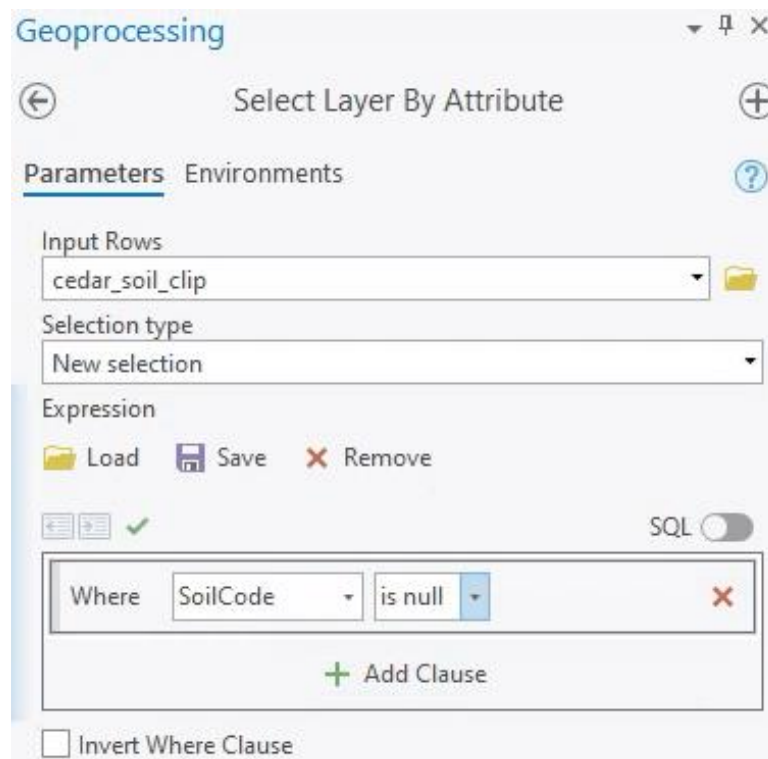
Additionally, you will see some dual soil groups, such as A/B and C/D, assigned to some units. This group means that part of the units behave as A and part as B. Here we simplify the analysis by ignoring the dual hydrologic groups by assigning them as their first group. For example, A/D group will be assigned as A group. In practice, you may want to consider these differences, but it is better to keep it simple for learning purposes.

We provide two ways to convert the soil hydrologic groups into integers. One is querying and modifying by built-in tools. Querying features by attributes values or conditions is an essential and useful skill to manipulate the data by ArcGIS pro. Although this method takes some time, we still recommend you practice it. The other method applies a few Python codes to encode. This method is very efficient and faster, but you have to be familiar with the syntax of Python code.

Method 1: querying and modifying by built in tools (demonstrated in video on Brightspace)

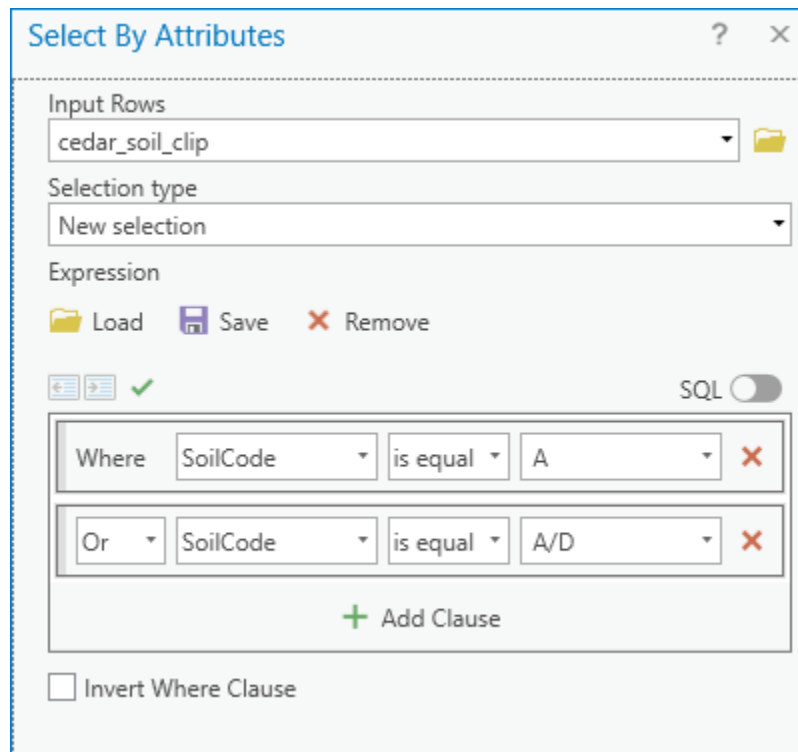
1. Replacing Null values

To replace the Null value in the SoilCode field, we use *attribute query* or *select by attributes* to select all the rows that satisfy the condition (is null). The *attribute query* can be defined by **right clicking** the *feature layer*→*Properties*→*Definition Query*. Yet, we suggest using *select by attributes* that can be accessed from the top tab (*Table View*) when you open a table. Select rows by the expression where **SoildCode is null** as the figure below. Then, use the *field calculator* to assign the SoilCode='C' for those rows. Save your edits and clear selection.



2. Encoding codes as integers

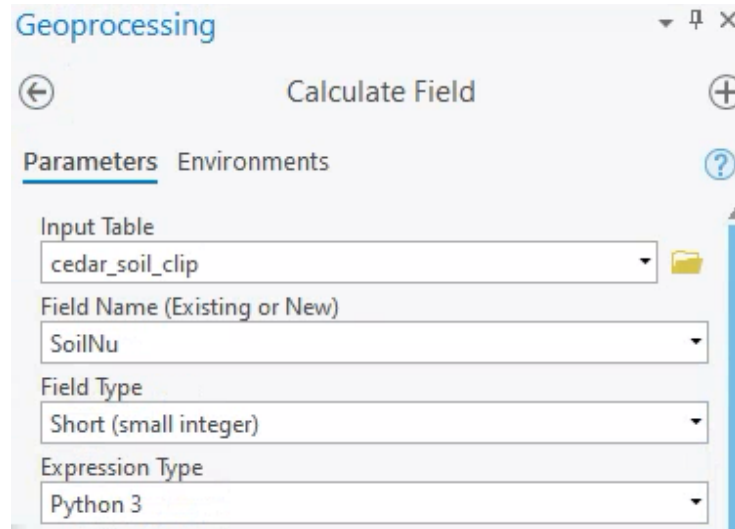
After we get a SoilCode for all rows in cedar_soil_clip attribute table, we are going to encode the soil groups (A-D) as numbers (1-4). First, create a new field called *SoilNu* to store integer values (Data Type = **Short**). Next, you need to select all features with SoilCode = A or SoilCode=A/B (using Select by Attributes) as shown below. Push Run to select all features that satisfy the query.



With the rows selected in the attribute table of cedar_soil_clip, use the field calculator to assign SoilNu = 1. Once this is done, clear the selection. Repeat this process to assign SoilNu = 2, 3, and 4 for B, C, and D respectively. Make sure SoilNu field is populated for all features. Unselect all features. Save your map/project.

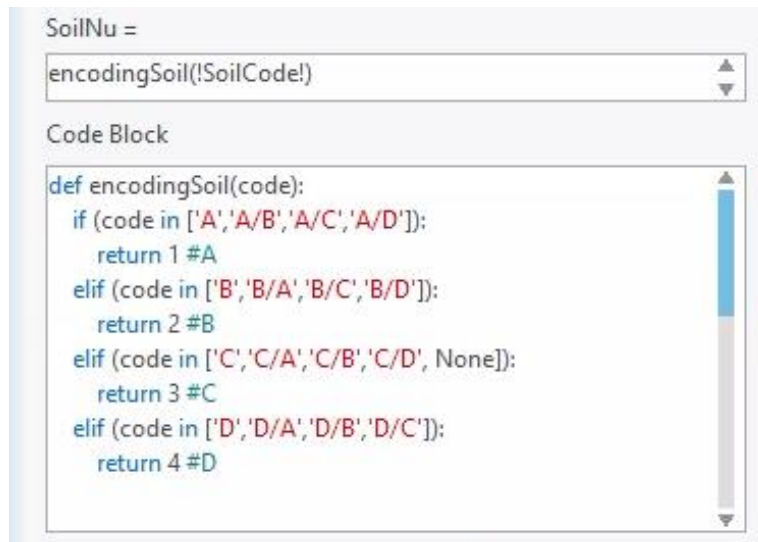
Method 2: Using Python code to convert SoilCode to integers (not included in the video)

In many Geoprocessing tools in ArcGIS Pro, we can use a small piece of Python code to accomplish certain tasks can be used in some tools in ArcGIS pro. Here, we are going to write a function for encoding the soil groups. Open *attribute table* and click *Calculate Field*. We can create a new field by filling in a new field name and select its type as **Short** in *Calculate Field*.



1. Write the function for encoding with conditions

The function is defined in the code block. The if and elif statements check whether the value of the argument (code) is the same as any element in the list ([...]). Note that a Null value can be detected as None in Python. **Note that Python uses indentation (tab) to indicate a block of codes**, you should carefully keep the indentations in the code as shown in the figure below. Write the code into your code block as shown below.



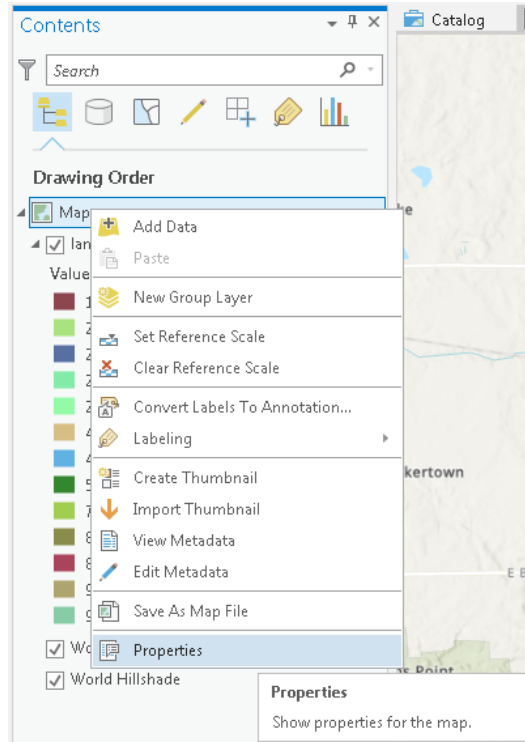
2. Call the function for returned values.

We call the function we just defined. It takes the SoilCode field as an input argument. Type your function name (here we use “encodingSoil”) in the box followed by a pair of brackets with the arguments. You can add the argument by **double clicking** the *SoilCode*. After setting down all of these, click run and you will get a new field with integers.

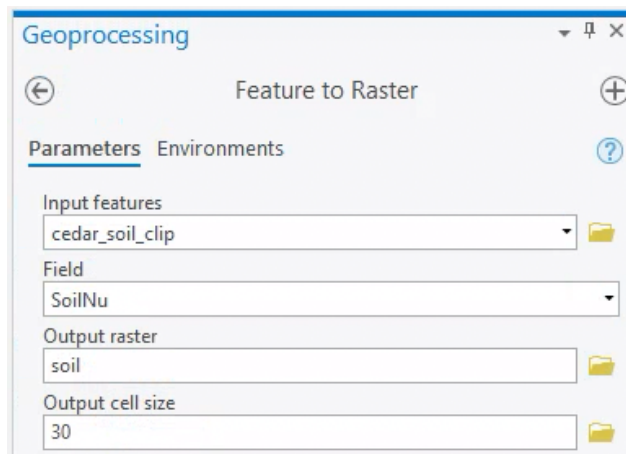
Note: if you have problems with writing the Python code, you can use notepad to open encodingSoil.txt (provided in the data folder) and copy the script into the code block.

Converting soil map unit polygons to hydrologic soil group raster

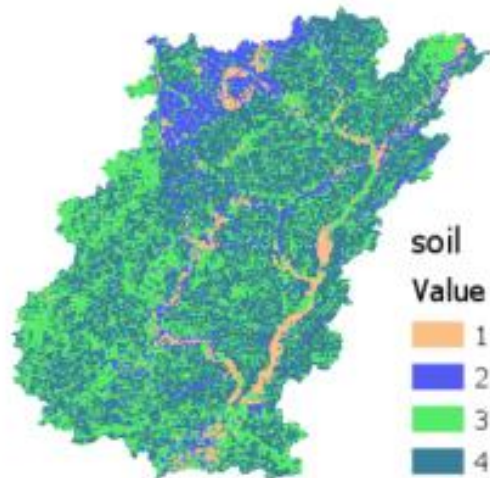
Once we have a SoilNu assigned to each soil polygon, the next step is to convert the soil layer (polygon layer) to a raster. Before proceeding make sure all rows in the SoilNu are populated. We want to match the properties of the soil raster to the landuse raster we have for Cedar Creek. Add cedar_lu landuse raster to the map and explore its properties for spatial reference and cell size. Right click on Map → Properties → Coordinate Systems as shown in figure below. Make sure the spatial reference matches with soil data and the cell size is 30 m.



From the toolbox, search or select the Feature to Raster or Polygon to Raster tool. Use your soil polygon layer (*cedar_soil_clip*) as input features, *SoilNu* as Field and Output Cell Size as 30. Name the output raster as *soil* and store it outside .gdb. The horizontal and vertical resolution of this raster must match with the land use raster, which is 30 m in this case. In the *Environments* tab, use *cedar_lu* raster for the output extent and snap raster and push **Run** to create the soil raster.



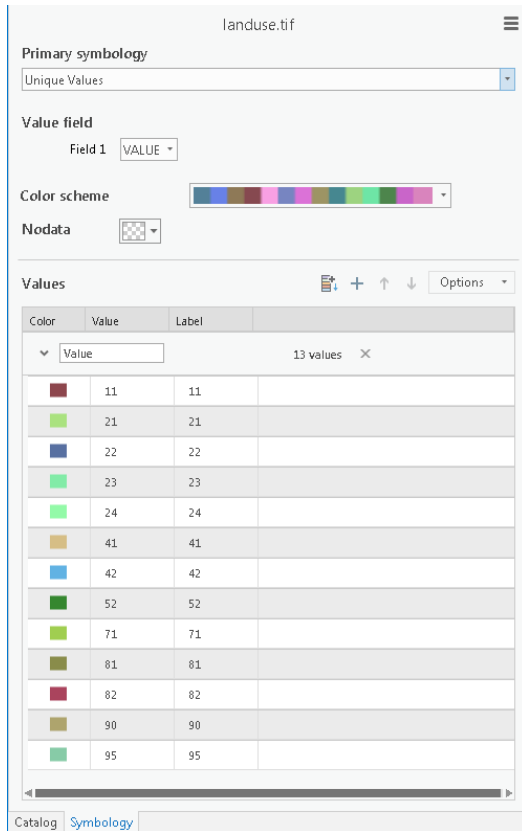
After the process is complete, the raster (*soil*) will be added to the map.



Before proceeding, make sure that extent, cell resolution and the number of row/columns matches for both soil and lu_reclass raster. This is important for the next step. If the extent or the number of rows/columns do not match between the two rasters for some reason, use the smaller dataset to clip the other.

Processing landuse data or raster for creating CN Grid

Check the symbology of cedar_lu raster and make sure you see all land use classes. If you do not see 13 different classes in the content page, change the symbology (**right click on the landuse layer**→**Symbology**) of the landuse layer. Use *Unique Values* in Primary symbology so that you can see the different landuse with unique colors. Different landuse types has a unique number ranging from 11 to 95 which represents a landuse type. The correspondence between these numbers and landuse types is shown below.



| NLCD Land Cover Classification Legend | |
|---------------------------------------|---------------------------------|
| [Blue] | 11 Open Water |
| [Light Blue] | 12 Perennial Ice/ Snow |
| [Light Brown] | 21 Developed, Open Space |
| [Brown] | 22 Developed, Low Intensity |
| [Red] | 23 Developed, Medium Intensity |
| [Dark Red] | 24 Developed, High Intensity |
| [Grey] | 31 Barren Land (Rock/Sand/Clay) |
| [Green] | 41 Deciduous Forest |
| [Dark Green] | 42 Evergreen Forest |
| [Light Green] | 43 Mixed Forest |
| [Brown] | 51 Dwarf Scrub* |
| [Light Brown] | 52 Shrub/Scrub |
| [Light Green] | 71 Grassland/Herbaceous |
| [Light Green] | 72 Sedge/Herbaceous* |
| [Light Green] | 73 Lichens* |
| [Light Green] | 74 Moss* |
| [Yellow] | 81 Pasture/Hay |
| [Brown] | 82 Cultivated Crops |
| [Light Blue] | 90 Woody Wetlands |
| [Blue] | 95 Emergent Herbaceous Wetlands |

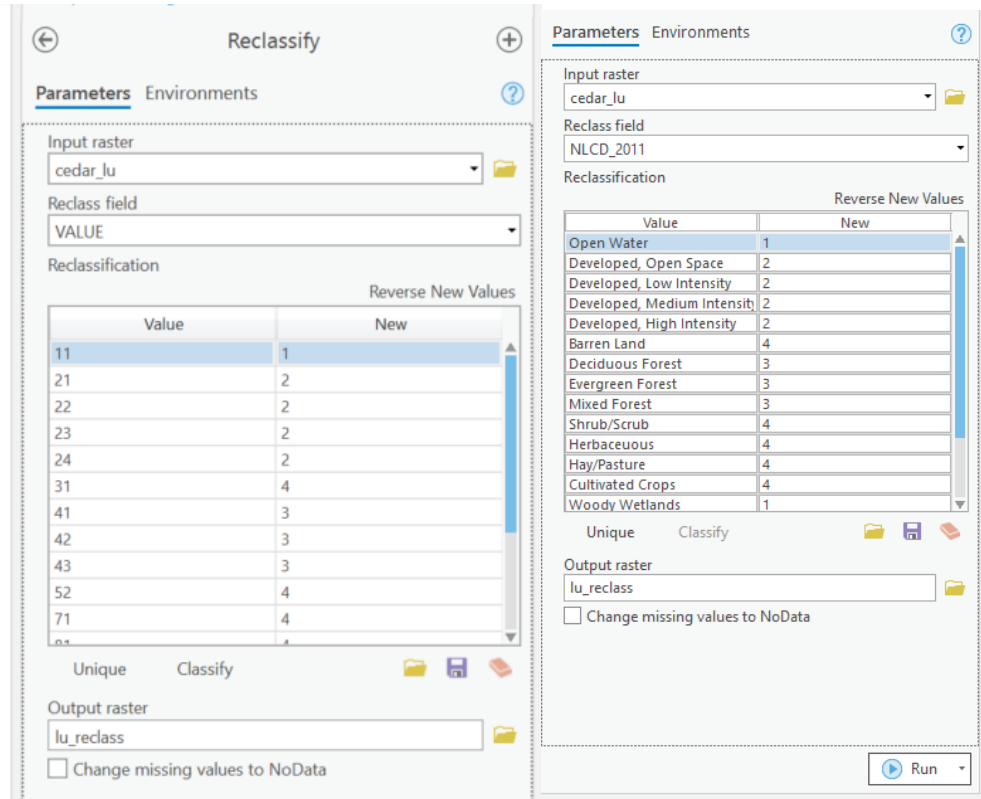
* Alaska only

You may notice that the landuse data (cedar_lu) provides many landuse categories. You can either leave these categories unchanged or reclassify the grid to reduce the number of landuse classes to make the future processing tasks easier. If you look at the visualized data in the map, you will see that most of the area includes grass/crops, followed by forest, developed land, and then water. We will reclassify the landuse to represent these four major classes. The following table shows how we will accomplish the reclassification of the landuse raster (you are free to have more or less classes).

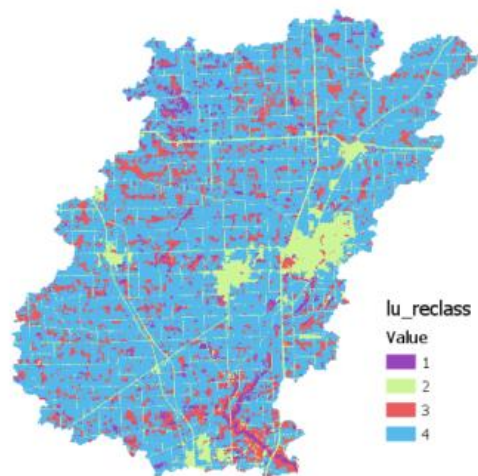
| Original NLCD classification | | Revised classification (re-classification) | |
|------------------------------|------------------------------|--|--------------------|
| <i>Number</i> | <i>Description</i> | <i>Number</i> | <i>Description</i> |
| 11 | Open water | 1 | Water |
| 90 | Woody wetlands | | |
| 95 | Emergent herbaceous wetlands | | |
| 21 | Developed, open space | 2 | Medium Residential |
| 22 | Developed, low intensity | | |
| 23 | Developed, medium intensity | | |
| 24 | Developed, high intensity | | |
| 41 | Deciduous forest | 3 | Forest |
| 42 | Evergreen forest | | |
| 43 | Mixed forest | | |
| 31 | Barren land | 4 | Agricultural |
| 52 | Shrub/scrub | | |
| 71 | Grassland/herbaceous | | |
| 81 | Pasture/hay | | |
| 82 | Cultivated crops | | |

To reclassify the landuse raster, **select** the *Reclassify* tool (**Spatial Analyst Tool**→ **Reclass**→ **Reclassify**) from the Spatial Analysis Toolbox.

In the **Reclassify** tool, select the Input raster as *cedar_lu* and the Reclass field as *Value or NLCD_2011*. Manually change the values according to the table shown above. Save the output as *lu_reclass* in your working folder (outside .gdb). After making all these changes, push **Run** to reclassify the raster.



The reclassified raster will be added to the project after the processing is complete. You now only have four landuse types in the reclassified raster as shown below. If the landuse raster is properly clipped, you would not see any values outside the given boundary. Otherwise, there might be a value of 128 for all cells outside the given boundary. This completes the processing of landuse raster for creating the Curve Number.



Note: You can also use the “Reclassify by table” or “Reclassify by ASCII File” tool for reclassifying the raster.

Creating CN grid using Soil and Landuse Raster

Now, you have two rasters (landuse: lu_reclass and soil) in the map document. The next and last step is to query each cell of these rasters and assign a CN value based on the published CN tables. A customized form of the CN table for the current scenario is given below.

| Land Use (LU) Description | LU value | Soil Hydrologic Group | | | |
|------------------------------|----------|-----------------------|------|------|------|
| | | A(1) | B(2) | C(3) | D(4) |
| Water | 1 | 100 | 100 | 100 | 100 |
| Medium Residential | 2 | 57 | 72 | 81 | 86 |
| Forest | 3 | 30 | 58 | 71 | 78 |
| Agricultural | 4 | 67 | 77 | 83 | 87 |

We will use this table and the raster calculator to query each cell in the soil and landuse grid to create the CN raster. The query is slightly long and is provided to you in the *cnquery.txt* file. You can use Raster Calculator within Spatial Analyst Tool (*Spatial Analyst Tools* → *Map Algebra* → *Raster Calculator*) to carry out this operation.

Now, open *cnquery.txt* file from the folder and copy the query statement. The operation & corresponds to Boolean “And” operation.

```
Con(("lu_reclass"==1),100,Con(("lu_reclass"==2) & ("soil"==1),57,
Con(("lu_reclass"==2)&("soil"==2),72,Con(("lu_reclass"==2)&("soil"==3),81,
Con(("lu_reclass"==2)&("soil"==4), 86, Con(("lu_reclass"==3) & ("soil"==1), 30,
Con(("lu_reclass"==3)&("soil"==2), 58, Con(("lu_reclass"==3)&("soil"==3), 71,
Con(("lu_reclass"==3)&("soil"==4),78,Con(("lu_reclass"==4)&
("soil"==1),67,Con(("lu_reclass"==4)&("soil"==2),77,Con(("lu_reclass"==4)&("soil"==
3),83, Con(("lu_reclass"==4)&("soil"==4),87,0))))))))))
```

Basically, it is a conditional statement. Paste the query statement in the syntax box in the raster calculator interface and save the output as *cngrid* outside geodatabase as shown below. Push **Run** button to generate the CN grid (*cngrid*).

