

# Data Management & Geoprocessing

## 1. Introduction

Data assembling and manipulation are big part of a GIS project. In this lab, you will be using FME, interactive tools, batch process and model-builder to prepare spatial data for the study area of Aleza lake research forest.

**Data locations:** L:\labs\geog413\data\management

<i>bdy</i>	Study boundary
<i>093j10.saf, 093j20.saf, 093i01.saf, 093i11.saf</i>	TRIM data in SAIF format
<i>93i-dem25m-utm10.asc, 93j- dem25m-utm10.asc</i>	Raster DEM in ASCII format
<i>fc93j10.e00, fc93j20.e00, fc93i01.e00, fc93i11.e00</i>	Forest coverage data in E00 files
<i>landuse</i>	Landuse
<i>viewpoints.csv</i>	View point locations in CSV format

## 1. TRIM data

Terrain Resource Information Management (TRIM) provides the base data for the Province of British Columbia. TRIM is a set of three-dimensional digital files that support development and management of land-related information.

- Create a folder for this course (eg. Geog413) and create a sub folder **management**
- Convert TRIM data listed above to SHAPE files with FME software
- Open FME Translator and run **trim2\_to\_shap.fme** from C:\Program Files\FME\gallery\trim2000
- Set the parameters as the following

TRIM SAIF input file name:	Click the Browse button and select <i>93i001.saif, 93i011.saif, 93j010.saif, 93j020.saif</i> (use the SHIFT key to add all datasets) from course folder
Shape Directory	K:\geog413\management\trim
Version	Updated
Attribution Type	Full Attribution

Examine the results from this conversion. You should have a list of ground features (*tctrl, tlake, trivr, troad, twtrl...*)

## 2. Clipping

- Open a new map file and add TRIM layers *tctrl, tlake, trivr, troads, twtrl* and *bdy*
- Change the default data frame **Layers** to **Aleza**
- Open the properties window of data frame **Aleza**, set Coordinate System to NAD\_1983\_UTM\_Zone\_10N.
- Save the map file as **management.mxd** in your local folder.

Notice that the TRIM data cover way larger area than the study boundary. We need to clip them to fit to the study boundary

You have done clipping work with CLIP tool interactively from entry level GIS course. Here we will use geoprocessing tools in more efficient way to perform the task.

Geoprocessing is based on a framework of data transformation.

The fundamental purpose of geoprocessing is to allow you to automate your GIS tasks. A typical geoprocessing tool performs an operation on a spatial dataset and produces a new dataset as the result of the tool. Geoprocessing allows you to chain together sequences of tools, feeding the



output of one tool into another. You can use this ability to compose an infinite of geoprocessing models that help you automate your work and solve complex problems.

Before precede, we will set working environments to make sure all output data go to the same spot.

- Set the working environment as the following  
 Current Workspace & Scratch Workspace: your local folder for this lab  
 Processing extent: same as **bndy**
- Make sure you have 3D and Spatial Analyst extensions enabled
- Examine the properties of each dataset
- As all TRIM layers and **bndy** are both in NAD83 UTM 10N coordinate system so they are ready to go for clipping.

### Batch Process

Next we will clip each trim layers we need with the boundary in batch process

- Right-click **CLIP** tool to open batch mode and fill the rows with the following setting.
- You can drag and drop the input dataset to each cell for input feature, select from dropdown list or manually type in.

Input Feature	Clip Feature	Output Feature Class
<i>tctrl.shp</i>	<i>bndy.shp</i>	K:\geog413\management\tctrl_clip.shp
<i>tlake.shp</i>	<i>bndy.shp</i>	K:\geog413\management\tlake_clip.shp
<i>trivr.shp</i>	<i>bndy.shp</i>	K:\geog413\management\trivr_clip.shp
<i>troad.shp</i>	<i>bndy.shp</i>	K:\geog413\management\troad.shp
<i>twtrl.shp</i>	<i>bndy.shp</i>	K:\geog413\management\twtrl_clip.shp

## 3. Extracting Features from TRIM data

### Extract Streams and River Banks

- Add TRIM layers *tctrl\_clip*, *tlake\_clip*, *trivr\_clip*, *troad\_clip*, *twtrl\_clip* (bath process does not add resulting layers to ArcMap automatically)
- Examine the TYPE field of *trivr\_clip* which contains water bodies. There are several types of water:

"Left Bank" and "Right Bank" are the outlines for big rivers which should be formed up as polygon feature and the rest of lines can be treated as streams. We will separate these features to form up new datasets

- Make attribute selection on *trivr\_clip* to extract all none river bank features
- Export the selected features to *streams* in your local folder

Now examine the rest of river lines. The river lines with "River-Left Bank" or "River-Right Bank" type indicate that these are big rivers and should be formed up as polygon features.

- As all the selected features are none river banks, the river banks can be selected by simply switching the selection. Right-click *trivr\_clip* ->Selection->Switch Selection.
- With the selection on, build polygon features with **Feature To Polygon tool**. Make sure to give two input *trivr\_clip* and *bndy* in order close the feature at edge

Input Features: *trivr\_clip*, *bndy* (add each)  
 Output Feature Class: K:\geog413\management\rivers.shp

This tool allows you to build polygon features from lines (selected)

You may notice that we use two layers to form up the polygons: *trivr\_clip* and *bndy*. If you look at the edge of *trivr\_clip* on both left and right sides, you should see that the two river bank lines are open, not closed. The *bndy* can be used to make up this part. Of course, you can manually change/modify the features to make it closed in Editing mode.

The resulting dataset **rivers** will have polygon features from both **bndy** and the selected features from **rivers\_clip**. You probably notice that **rivers** includes **rivers** and **bndy**. Some parts of **bndy** are not rivers. We need to remove these parts to keep only the river polygons.

- Turn on Editor, in Editing mode, set **rivers** as the editing layer. Delete the parts that are not rivers. Save the edits and stop editing. The resulting dataset **rivers.shp** contains the only river polygons.

### Build polygon features from **tlake\_clip**

The **tlake\_clip** contains lake outlines which can be used to build up lake polygons

**Note:** Check the dataset to see if any of them are not closed, especially at edge of the data. If Yes, you can either turn on the editing mode to close those features or use **tlake\_clip** and **bndy** together to build polygons (like above). Otherwise **tlake\_clip** itself is good enough.

- Follow the same steps above build polygon features from **tlake\_clip**. Note: there is no need to do any selection as all lines are lake lines. Save the final output as **lakes** in your local folder.

### Extract Island and Flood Land

Examine **twtrl\_clip** which contains water related information, such as flood land, island, marsh and etc.

Examine the data field CLASS in the attribute table of **twtrl\_clip**. We want to extract island and flood land to make a separated layer for each type of feature class.

- Select all Island feature from **twtrl\_clip** and build polygon features. Save the output polygon features as **island**
- Select all Flood Land feature from **twtrl\_clip** and build polygon features. Save the output polygon features as **floodland**
- Export **twtrl\_clip** as **contours**, **troad\_clip** as **roads**

In the table of content, keep the layers **bndy**, **streams**, **rivers**, **lakes**, **island**, **floodland**, **roads**, **contours** and remove the rest of data layers.

### Update Attributes

The spatial operations that modify the spatial locations/geometry do not automatically update the attributes related to the geometry such as AREA, LENGTH, and PERIMETERS etc. These information need to be updated manually.

- Calculate the area for each polygon features **lakes**, **rivers**, **floodland**, and **island**
- Calculate length for each line features **streams**, **roads**
- Save the map.

### ModelBuilder

**ModelBuilder** is a user-friendly way to automate a series of tasks. It provides functionality to build, test, validate and run a model. ModelBuilder is a part of ArcGIS geoprocessing framework. And it can run any tool in the ArcToolbox, including scripts, custom tools, and other models. ModelBuilder supports GDBs, shapefiles, tables, coverages, raster, CAD. ModelBuilder help you automate your tasks, layout processes and chain everything together and the model can be saved for future use.

**What is Model?** A model is how you automate your work. When you create a model, you are preserving a set of tasks, or a workflow, that you can execute multiple times. There are an infinite number of workflows you can automate using models. Model allows you to run a task consisting of several steps in one. A model can be built and run in command line, ModelBuilder, or script.

The created model is added to ArcToolbox as a model tool, which can be executed at any time in either dialog or the Command Line window. You can also execute the model within ModelBuilder to see the progress

- Use batch process to convert all forest cover files to coverage dataset with **IMPORT FROM E00** tool. Give a proper name (*fc93j10*, *fc93j20*, *fc93i01*, *fc93i11*) for the output

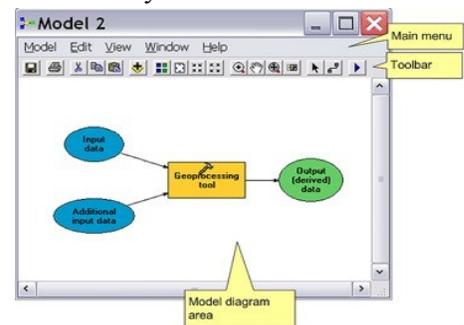
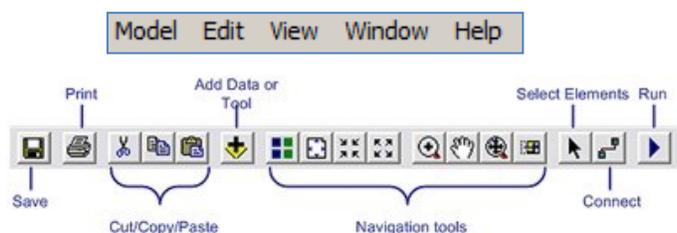
Examine the properties of forest coverage dataset *fc93j10*, *fc93j20*, *fc93i01*, *fc93i11*. These datasets are in BC Albers projection while the boundary file *bdy* is in UTM10 projection. You need to reproject forest coverage data to UTM 10 coordinate system and then clip each one with boundary *bdy*.

Models exist in ArcToolbox, before you create a new model, you need first create a new toolbox.

You will create a model to automate the tasks of reprojection and clipping for each forest cover dataset to fit to study area (PROJECT and CLIP)

- In ArcCatalog tab, right-click your **management** folder and choose New->Toolbox.
- Name the toolbox as **aleza\_tools**. This will create a new toolbox
- Now right-click on the **aleza\_tools** Toolbox and choose New->Model. The ModelBuilder window pops up. Here you can layout all your tasks and chain them together.

The Model window comes with main menu bar, toolbar and display area for model layout



## Set Model properties

- Click Model from the top menu and choose Model Properties
- Click General tab and type in **project-clip** for the name of the model and **Reprojection & Clip** for the label. The name is used for reference purpose such as to run it from command line/script, while label will be shown in the toolbox.

### NOTE: Do NOT use space and underscore for the name

- Click the Environments tab, scroll down and click Workspace. Here you can specify the working directory
- Check both current and scratch workspace. Click Value button located at lower right to set the work space to your local folder. That means all input and output will come or go to this place. Click OK to close the Model Properties window.
- From the menu bar, click Model-> Save. If you look at in the catalog tree, you should see the new model named **Reprojection & Clip** showing under the **aleza\_tools** toolbox. Refresh the folder if necessary

## Adding elements to a model

The easiest way to add an element to a model is by using the drag and drop

- In ArcCatalog tab, navigate K:\geog413\management folder and drag all fcover datasets drop to the model window. The input datasets are showing in blue.

The datasets added in the model window are showing in an oval shape but in different color (blue for input and green for output with appropriate setting). The tools are shown in an orange rectangle box with appropriate setting.

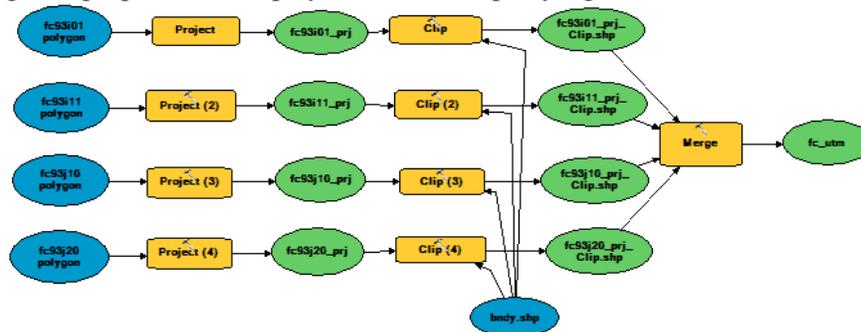
- Drag the **PROJECT** tool from ArcToolbox->Data Management->Projection and Transformation->Feature and drop it to anywhere in model window. Initially the tool is showing in box without color filled as the does not have appropriate parameter setting( input and output )
- Drag and drop the **PROJECT** tool three more times as we have four input datasets need to be reprojected.

- Arrange fcover datasets in a column at left and put four PROJECT tools in the second column. Make the input data and the PROJECT tools displayed side by side

The Add connection button  located at second right in the top menu allows you to add connection between dataset and tool to link input and output datasets to the tool.

- Click the Add Connection button and put the mouse cursor inside **fc93j10** and drag a line to a PROJECT tool and choose Input dataset. An arrow between two elements makes the dataset as the input of the tool.
- Double click on the PROJECT tool in the model window to fill the rest of parameters.
- Set the output coordinate system to NAD 1983 UTM10. Save the output as **fc93j10\_proj.shp**. Now the PROJECT tool box turned to orange and the output oval shape turned to green.
- Repeat the steps to finish the setting for the rest of three datasets. Name the output dataset like **fc93j20\_proj.shp**, **fc93i01\_proj**, **fc93i11\_proj** respectively

The display on the oval shape (input or output) will stay as it is. If you want to change the look of each oval shape (text) you can change the properties of display of an oval shape by right-click on an oval and choose Rename.



The second part you will do is to clip each reprojected dataset to the boundary.

- Drag and drop the CLIP tool to the model window four times and arrange them in the third column beside the each output dataset.
- Drag the **bndy** to model window. This will be used as clip feature
- Click the Add Connection button and draw a connection between each output datasets from PROJECT to a CLIP tool as input feature. Draw a connection between the **bndy.shp** to each of CLIP tool as clip feature
- Double click the CLIP tool to fill the output dataset to K:\geog413\management\fc93i01\_clip.shp.
- Repeat the previous step to set the output for CLIP tool **fc93i11\_clip.shp**, **fc93j10\_clip.shp**, **fc93j20\_clip.shp** respectively.

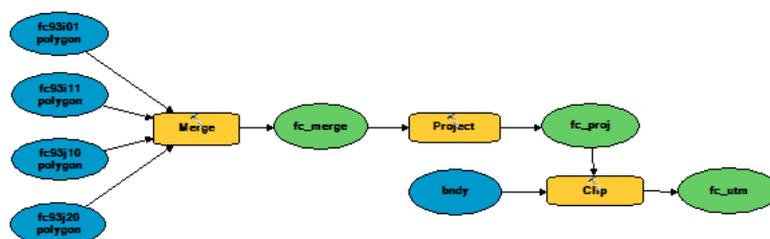
You should have a model built looks like the image above.

- You can change the display label for each toolbox or dataset by right-click the tool/dataset->Rename. Rename “Project” to “Reproject fc93i01” and rename Clip to “Clip fc93i01”.
- Rename the rest to the operation performed for each toolbox
- From the top toolbar, click save button to save the tool(the first button from left on the toolbar)

To finish up the process, we also want to merge all clipped fcover datasets into a single dataset with MERGE tool

- Drag MERGE tool to the model window and connect each clipped fcover dataset to the MERGE tool as input datasets. Save the output feature as **fc\_utm**

OR you can have the model as the following



You can run the model inside the ModelBuilder or in command line or as a script.

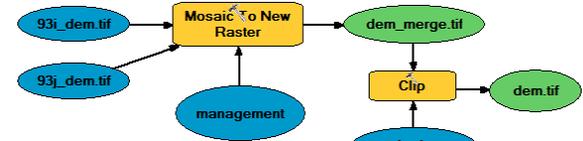
- To run the model from ModelBuilder, click the RUN button (the last button at right on the toolbar). You can tell the progress by color. The task currently working will show in red

If you run the model from ModelBuilder, you will have all dataset including the intermediate datasets. If you run the model by double-clicking the model in ArcToolbox, you will have only the final output datasets. That saves disk space during the processing. In our case now, you should have all datasets under your management folder including the intermediate datasets

You can always go back to make changes. To Edit the model, right-click on the model and choose Edit. Here you can make changes

Now build a model call it **createDEM** to merge two raster dems into one raster dataset with MERGE TO NEW RASTER and them clip the merged raster with *bdy* using CLIP tool located under raster processing (Data management->Raster->Raster Processing->Clip)

- Using batch mode to convert *93i-dem25m-utm10.asc* and *93j-dem25m-utm10.asc* to raster with ASCII to Raster tool. Save the output as *93i\_dem.tif* and *93j\_dem.tif* respectively
- Open a new model window and drag *93i\_dem.tif* and *93j\_dem.tif* to model window. Drag MERGE TO NEW RASTER and CLIP tool (Note: use Data management->Raster->Raster Processing->Clip) to model window.
- Make connections from each DEM to MERGE tool as input feature. Double click the MERGE tool to fill other parameters.
  - Output Location: K:\geog413\managment
  - Raster Dataset Name: *dem.tif*
  - Pixel Type: 32\_BIT\_SIGNED
  - Number of Bands: 1
- Leave the rest of parameters as the default
- Save the mode as **createDEM** and run the model to get the final version of DEM



### Building a general tool with variables and parameters

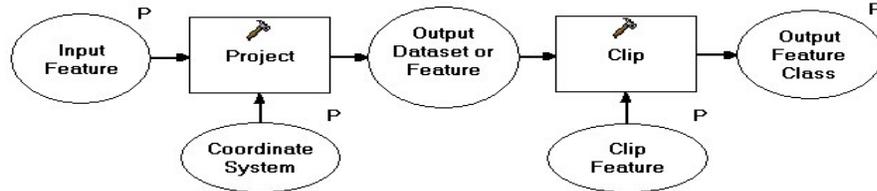
Reprojecion and clipping operations are common tasks for most GIS project. The model we just built is very specific as the input, clip feature and output datasets are all fixed. It will be nice to have a general tool that allows working on any datasets. This can be done by using parameters in the model.

Now we will build a tool that perform reprojection and clipping on any dataset.

We can make this model come with four parameters to allow accepting user's input when you run the model. They are: input feature, coordinate system, clip feature, and output feature. This gives the tool flexibility.

- Create a new model under **aleza\_tools**
- Set the name to **reproject-clip-general** and the "**Reproejcion & Clip General**" for the label and the same workspace as before. Save the model
- Drag PROJECT and CLIP tools to Model window and arrange side by side (PROJECT as left and CLIP at right)
- Make a connection between the output from PROJECT to the input of CLIP
- Right-click in anywhere in the Model window and choose Create Variable. Variable is a place holder which allow you to hold different thing each time
- Set the variable type to **Feature Layer** where will hold the input dataset.
- Right-click the variable and choose Rename. Change the name to **Input Feature**. This variable will be used for taking input dataset.
- Right-click the **Input Feature** element and choose Model Parameter. A letter **P** is showing at upper right corner of the element. This allows user to choose input when running the mode.
- Repeat the steps to create a variable with the same variable type and name it Clip Feature. This variable will be used for holding clip feature (cookie cutter)

- Create another variable and name it to Output Coordinate System and set the variable type to Coordinate System. Make it as Model Parameter. This parameter allows user to set the output coordinate system.
- Make the Output Feature Class at right to CLIP box as a Mode Parameter and rename it to Output Dataset
- Make a connection from the Input Dataset to PROJECT box as input feature and make another connection from the Coordinate System variable to PROJECT box (see the image below) as output coordinate system
- Make a connection from the Clip Feature variable to CLIP box as clip feature. You should have tool diagram like the following image



- Double click **Project** tool to open the tool window. Set *temp.shp* for the Output Dataset under your local workspace and click OK. This is just a temporary dataset and will be deleted once the model finished. So it doesn't matter what name is.
- Now save the tool by clicking on the save button (the first button left on the toolbar)
- Right-click the model and choose Properties. Click Parameters tab, make sure the order of parameters is correct(Input Feature, Output Coordinate System, Clip Feature, Output Feature). Use the arrow buttons at right to change the order if necessary
- Save the model as **reproject-clip-general**

This tool is more general and can be used on any dataset at any time. Give a try to run this tool.

You cannot run this general tool in Model Window. You will need to go to ArcCatalog tab to run it,

- Double click the tool **reproject-clip-general**. You should have a window popped up
- Click the Environment button to set the workspace to your local folder
- Now set the parameters as the following by clicking the button at right for each parameter

Input Feature:	L:\labs\geog413\data\management\landuse.shp
Output Coordinate System:	NAD_1983_UTM_Zone_10N
Clip Feature:	L:\labs\geog413\data\management\bdy.shp
Output Feature:	K:\geog413\management\landuse_utm.shp

You will have the final output dataset *landuse\_utm.shp* in your local folder. Check it.

This tool can be also used to run a loop to complete the task for a list of datasets.

- Now generate a point dataset from **viewpoints.csv** with Create a Feature Class->From XT table. Save the result as *viewpoints.shp*

By now you should have all spatial data ready for GIS analysis. You can use the batch process and model builder to automatic your GIS analysis task

~~~~~ The End ~~~~~