Buildout Analysis Tool Documentation

Code Author: Connor Hornibrook
Language: Python 2.7
Git Repository: https://github.com/cfh294/BuildoutAnalysis

Tool Overview

This tool was developed as a Python-scripted version of the NJ DEP Buildout Analysis Model Builder Model. Various methods were developed, mainly those that calculated minimum lot size for zoning data and buildout numbers for both pre- and post- environmental constraint areas.

Imported Libraries

- sets
- math
- arcpy

Method Overview

minimumLotSize(zoningData, outputWorkspace)

- Takes zoning data and appends a "MINLOT" field. Using the zoning schedules of Oldmans, Hopewell, and Carney's Point, python dictionaries were created that assigned a minimum lot value to each zone. The input zoning data (shapefile or geodatabase feature class) is then updated. For each new town that would want to use this model, this method would have to be updated.
- Only works for Hopewell, Oldmans, and Carney's Point Townships right now.

uglyFieldManagement(fc)

- After a long series of "Identity" tool calculations, there are a lot of left over FID fields that are unnecessary once the model is finished. This method eliminates these fields.

nitrate_BO(minlot, septicDensity, shapeArea, isSeptic, CZ_BO_number)

- Calculates and returns buildout numbers under Nitrate Dilution standards.

currentZoning_BO(minLot, shapeArea)

- Calculates and returns the buildout numbers under current zoning.

canSplit(NO3_val, CZ_val)

- Returns a Boolean value that pertains to whether or not a parcel can be split, based on Nitrate dilution buildout and current zoning buildout numbers

buildoutCalculations(featureClass, isPost)

- Using the previous two methods, appends the GIS data with buildout numbers.
- Run twice, for pre- and post-constraint erase

Input Parameters

- **Zoning** feature class or shapefile
- The **town name** that the user would like the output file to have attached to it (type: String). For instance, if the user types "oldmans" in this area, the final file will be named "oldmans_final_result".
- **Additional constraint data** in the form of shapefiles or feature classes (multi-value). This is an optional parameter, that is only used if a town has additional GIS data that it would like to use as part of the constraint erase.
- **Output workspace** that the user would like to save the new file. Note: the location of this file must be on a drive/server that isn't almost full; this is due to the many temp files that will exist for a short time during the duration of the model.
- **Constraint Files Workspace** the workspace containing the constraint files

Setup

1. Within an ArcToolbox, add a new script and call it "Buildout Analysis"
2. Under the Source tab, set the script file to wherever the python file is saved on the computer
3. Under the Parameters tab, do the following (in this order):
    a. Create a parameter called "Zoning" of type Feature Class
    b. Create a parameter called "Municipality" of type String
    c. Create a parameter called "Additional Constraints" of type Feature Class. Set MultiValue to "Yes" and Type to "Optional"
    d. Create a parameter called "Output Workspace" of type Workspace or Feature Dataset
    e. Create a parameter called "Constraint Files Workspace" of type Workspace
4. Click "OK"
5. Before use, the Constraint Files Workspace needs to be downloaded from the git repository.

Data

Zoning

Not all zoning GIS data is available publicly, and currently this script only works with Carney's Point Township, Hopewell Township, and Oldmans Township (*as of 4 August 2015*). If one would like to make this tool compatible with new zoning data, they will have to edit the minimumLotSize() function to include the new town's minimum lot size laws.

Model Input Files

This model utilizes several state-wide files (open space, preserved farms, etc.). The workspace containing these files is available, but the file size was too large for Git. It can be obtained by contacting the members of this project.

Process Flow Chart (generalized)