## Civilian Topographic Map (CTM) Generalization

Out-of-the-box, Esri provides geoprocessing tools, python scripts and python site packages with the building blocks an organization needs to automate generalization of their data. Esri provides these components rather than a completed set of tools because the schema of the large scale data, the desired output scales and the expected results are unique. With these components, anyone can combine the generalization tools using Model Builder and\or Python until the desired output is achieved.

With so many generalization tools it can be confusing and difficult to get started with generalization. The CTM generalization samples illustrate how to generalize data in the CTM schema. While these models are designed specifically for the CTM schema they can be adapted to other data schemas. The CTM generalization models work on 5 themes of data: transportation, buildings, hydrography, land cover and elevation.

Contents

[Civilian Topographic Map (CTM) Generalization 1](#_Toc427247569)

[Generalization Approach 1](#_Toc427247570)

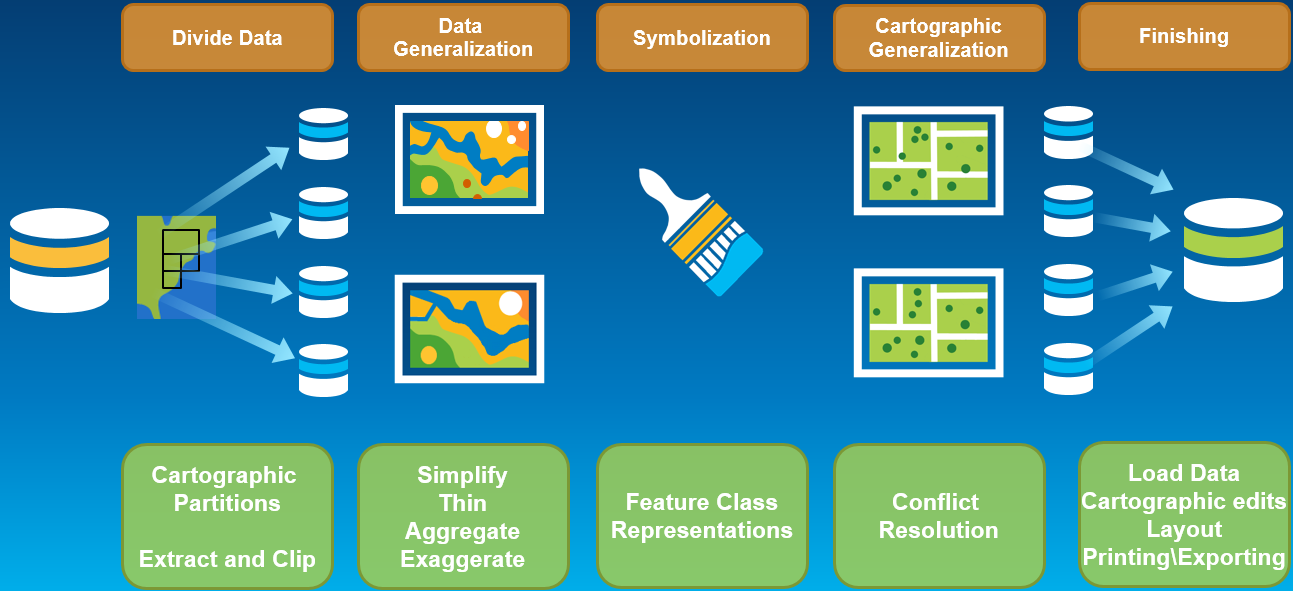
[Running the CTM 50K Generalization Models Individually 2](#_Toc427247571)

[Running the CTM 50K Generalization Models automatically 4](#_Toc427247572)

[Tips and Tricks for Using the CTM 50K Generalization Models 5](#_Toc427247573)

## Generalization Approach

The first step in producing a map at a specific scale is to produce data at that scale. Traditionally, organizations engage in data collection activities at that scale to produce the data. However, this can be a time consuming and expensive process. Cartographic generalization focuses on taking large scale data and simplifying it though automated methods so that it is appropriate for smaller scales.



The next step in producing a map at a specific scale is to symbolize the data with symbology appropriate for that scale and output product. After the data is symbolized, cartographic generalization and conflict resolution occurs. This ensures that there is clarity and separation between the symbols on the output map product. Because feature class representations are typically used for symbology, cartographic generalization is specific to that symbology and map product.

## Running the CTM 50K Generalization Models Individually

The CTM\_50K\_Generalization.tbx contains the models used to generalize Civilian Topographic Map data so it is appropriate for the CTM 50K products. The CTM samples include data collected at 25K over Salt Lake City. This data can be used when testing these models.

1. Make sure you have unzipped the CTM\_ProductLibrary.gdb.zip
2. Make sure you have unzipped the SaltLakeCity.gdb.zip.
3. Make a copy of the SaltLakeCity.gdb and name it SaltLakeCity\_50K.gdb.

*The generalization models are designed to alter the input data. For this reason it is important to make a copy of the input database so you can retain a copy of the large scale data as well as having a database appropriate for the 50K output scale.*

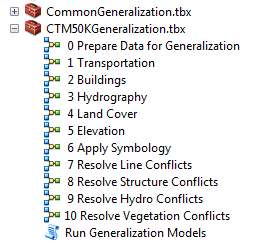
*If using data other than the sample Salt Lake City data that covers a larger geographical extent, you may need to partition the data into smaller sections which allows for data to be processed on multiple machines and to ensure that the generalization can be completed within the hardware constraints of your machine.*

1. Create a new scratch database that will store temporary output from generalization. Name the database something like Scratch.gdb.

*If you do not wish to create a new scratch database, you can use the Default.gdb or in\_memory.*

*You need to use the same scratch workspace for all models as some of the temporary data is used for multiple models.*

1. Open the CTM\_50K\_Generalization.tbx



*Inside the toolbox you will see toolsets and a series of numbered models. You will run the numbered models. The models included in the CommonGeneralization.tbx do not need to be run as they will be called from the numbered models. The CommonGeneralization.tbx and CTM50KGeneralization.tbx must remain in the same directory on your machine.*

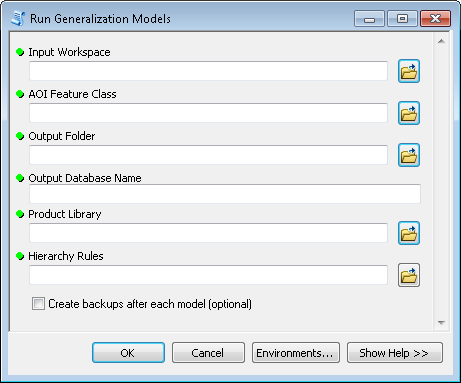
1. Execute the models in order starting from 0.
   1. The Prepare Data for Generalization model must be run first. This model ensures that fields and feature classes used by the other models are added to the database. This model also performs some basic data cleanup to fix errors that could cause failures with tools used in other models.
   2. All the models have two common fields. Use the following values.
      1. Input workspace: SaltLakeCity\_50K.gdb you copied in step 3.
      2. Scratch workspace: Scratch.gdb you created in step 4.
   3. For any models requiring an Area of Interest, use the AOI\_50K\_Demo feature class from the SaltLakeCity.gdb\Reference\_Layer feature dataset.
   4. For any models requiring a Hierarchy Rules file, use the Generalization\_Hierachy.vvs file in the generalization directory.
   5. For any models requiring a Product Library, use the CTM\_Product\_Library.gdb you unzipped in step 1.
   6. Each model has other parameters like minimum area or generalization tolerance that are used by the generalization tools in the model to determine how much generalization to perform on the features. The default values are starting points that work well for the Salt Lake City CTM data. These values can be increased for more dramatic generalization or decreased to retain more features or feature details.
   7. If desired you can make a copy of the data after running each model to review the changes.
2. Once all the models are run, open the CTM50KTemplate.mxd from the CTM > Fixed50K >Cartography > MapTemplates directory to view the results.
   1. Update the broken links to point to the SaltLakeCity\_50K.gdb geodatabase that contains the results of the generalization.

*Some of the generalization tools use the IS\_VISIBLE field on the feature classes to determine whether or not individual features should be visible in the output map. If you manually add the generalized data to the map, add a definition query to all layers with the SQL statement “IS\_VISIBLE = 0 or IS\_VISIBLE IS NULL”.*

## Running the CTM 50K Generalization Models automatically

If you don’t want to run the generalization models one at a time, you can use the Run Generalization Model python script tool to automatically run all the models in order.

1. Make sure you have unzipped the CTM\_ProductLibrary.gdb.zip
2. Make sure you have unzipped the SaltLakeCity.gdb.
3. Open the CTM\_50K\_Generalization.tbx.
4. Open the Run Generalization Models tool.



1. Populate values in the Run Generalization Models tool.
   1. The Input Workspace should be the unzipped SaltLakeCity.gdb.
   2. The AOI Feature Class should be the SLC\_AOIs feature class from the SaltLakeCity.gdb\Reference\_Layer feature dataset.
   3. For the Output Folder, choose a directory on your machine where the generalized database will be created.
   4. For the Output Database Name, enter a name for the generalized database that will be created.
   5. For the Product Library, point to the CTM\_ProductLibrary.gdb.
   6. For the Hierarchy Rules, point to the Generalization\_Hierarchy.vvs file in the generalization directory.
   7. Optionally, enable the Create backups after each model option. When enabled, a database will be created with the results after each model is run.
2. To view the results open the CTM50KTemplate.mxd from the CTM > Fixed50K > Cartography > MapTemplates directory.
   1. Update the broken links to point to the SaltLakeCity\_50K.gdb geodatabase that contains the results of the generalization.

*Some of the generalization tools use the IS\_VISIBLE field on the feature classes to determine whether or not individual features should be visible in the output map. If you manually add the generalized data to the map, add a definition query to all layers with the SQL statement “IS\_VISIBLE = 0 or IS\_VISIBLE IS NULL”.*

## Tips and Tricks for Using the CTM 50K Generalization Models

* Executing these models requires an ArcGIS Advanced license along with Spatial Analyst and Production Mapping extensions.
* Make sure that the location of all databases used in these models do not have a space in the path name.
* The scratch workspace required as input to the models is a database where intermediate data will be written. You must have write access to this database. It is recommended that you create a new file geodatabase named “Scratch” in the same location as your input database.
* You can use in\_memory as the scratch workspace. Writing geoprocessing output to the in-memory workspace is an alternative to writing output to a location on disk or a network location. The in\_memory workspace works well for the Salt Lake City data but may cause performance issues or the inability for models to complete if used with larger datasets. When data is written to the in-memory workspace, the computer's physical memory (RAM) is consumed. If too much data is written to this workspace, all the computer's memory may be used up and additional data cannot be written to memory.
* These models can be reused with any data in the CTM schema. However, if the data was collected differently than the Salt Lake City data then additional pre-processing or tiling of the data may be necessary to ensure that the models run without memory issues. For example if the data was originally collected at 10K instead of 25K like the Salt Lake data:
  + There may be too many features in a 50K output map to process in one run of the models. You may need to split the data out into a few databases, run the models on each database and then bring them back together for the final map creation.
  + Some of the features, like roads, may be been collected as different feature types. For example in the Salt Lake City data, the roads were only collected as lines. If you collected your data at 10K you may also have road polygons in your data. You may need to add additional pre-processing steps to remove the road polygons and repair any gaps in vegetation or other surrounding features.
* As part of the elevation generalization model, a raster is created from the existing contours and elevation points. If you have a DEM for your data it is recommend that you use the DEM to simplify the contours rather than the raster created as part of the model.