

## **tGAP Wrapper User guide**

**RGI-233**

**Usable (and well scaled) maps for consumers**

**Version 1.0**

**Project Code: 33042**

**24/02/2009**

# Populate TGap structure / Run Generalization

## Overview

Creates a TGap or constrained TGAP topology dataset (as defined by TU Delft) and can also generalize the dataset in order to create a vario-scale generalized dataset.

TGap structure has been designed by:

- Arta Dilo
- Martijn Meijers
- Peter van Oosterom
- Theo Tijssen

The TGap Wrapper works in two modes:

- Populate TGap dataset: Takes a planar Radius Topology manifold, and a theme (a set of class importance and compatibility rules) and generates a tgap/ctgap dataset.
- Generalize TGap Topology: Takes a TGap topology dataset and performs either TGap or Constrained TGap Generalization using the code from Arta Dilo and Martijn Meijers

## Using the Wrapper

These steps describe how to create a generalized TGap dataset starting from a set of spatial data. If you only need to perform certain steps, then you can simply start following the instructions at the appropriate place. The summary of the steps required is:

1. Find a dataset
2. Create a topology manifold for this dataset using the Radius Topology Quickstart wrapper
3. Create a TGap Theme (weights and compatibility tables)
4. Create a TGap dataset from this Radius Topology manifold using the TGap Wrapper
5. For constrained TGap enter correct region values in the face table
6. Generalize the TGap dataset using the TGap Wrapper

### Start with a dataset of polygons which form a planar partition

Any set of polygons stored in Oracle as SDO\_GEOMETRY can be used with this process. These should form a clean coverage of the data:

**Holes** Holes in this coverage are tolerated because they will be represented as features in a class called 'Hole' within the TGap dataset. When you create the TGap dataset, these holes will be flagged as warnings with a view showing you the geometries of these holes.

**Overlaps** Overlaps in the coverage are will halt the TGap dataset generation process. When you create the TGap dataset, these overlaps will be flagged as errors with a view showing you the geometries of these overlaps. Creating the Topology using Radius Topology will snap any small overlaps away, and then larger overlaps can be found and identified in the TGap dataset creation

process.

**Islands** The data must also form a single continuous 'cluster' with no islands. If there are any islands within the data then the dataset creation process will be halted and the location of all the islands reported to the user.

**Multiple tables** This data is expected to normally be stored in a single table (although multiple tables are supported).

### Identifying a feature's class

To identify the class of a feature, there should be one of the following defined on the table:

- Either a column on this table which defines the name of the class. If there are multiple tables then the column should have the same name in each table.
- Or a stored procedure which takes one or more feature columns and returns the name of the class. E.g if the class name can be calculated by using columns 'TYPE' and 'THEME' then in the GUI the function should be named as MY\_FUNCTION(TYPE, THEME)

### No spatial index needed

There is no spatial index needed on the table. In fact the topology creation process will be faster if the spatial index is removed first. It is recommended however to have Oracle spatial metadata in place as this means that the Radius Topology quickstart wizard can read the spatial parameters from this metadata and the user does not have to specify it.

## Create Topology using Radius Topology Quickstart wizard

Once you have the table(s) of spatial data, you need to run the Radius Topology Quickstart Wizard and Connect as the user who owns the data.

**Hint: If you cannot connect to Oracle using the standard tab (this uses the OCI driver), try clicking on the advanced tab (this uses the thin driver) and enter the hostname as the machine name where the database is running.**

Choose 'Create new topology for a set of tables'. Select the tables containing your spatial data to be topologically structured, give the manifold a name (the name is not important, but we recommend you enter the name of the table containing the data) and then you are on the manifold settings page. There are two tabs on this page. The manifold metadata tab should be populated with values and you should ensure that the 'Manifold is Planar' box is ticked. On the topological rules tab, you should enter a value in the 'Default minimum snapping tolerance' box. This is a tolerance used to snap small overlaps and gaps together when forming the topology. It is important that:

1. This value must be chosen carefully. The value should be a positive non-zero value. A typical value for data in a projection in metres is '0.01' (1cm) or '0.001' (1mm). You must make sure that the snapping tolerance is not too large otherwise features will not all structure because some might otherwise collapse to a point or have sections where there is more than one point that it could snap to.
2. This value should be at least ten times larger than the Geometric Tolerance value from the

'Manifold Metadata' tab. If the geometric tolerance is larger than 1/10th of the snapping tolerance, then simply make the geometric tolerance smaller.

When you run the wizard the topology will be created. Any features which couldn't be structured will be counted and these can be identified as having a null value in the LSL\_TOPO\_OBJ\_ID column. Features do not structure because of one of these reasons:

1. A real geometric error. A problem such as a duplicate vertex or a self-intersecting polygon
2. A virtual geometric error which is only a problem because of the snapping tolerances. For example a polygon which has a narrow section that is narrower than the snapping tolerance will not structure because the snapping would cause the section to collapse and cause a self-intersection. To fix these either reduce the tolerance or edit the feature to make the section narrower.
3. A complex situation where a point on a feature has more than one location within tolerance that it could snap to. Either reduce the snapping tolerance and try again, or edit the data to make the features line up more accurately.

## Create a TGap Theme

Before you create a TGap dataset, you need to create a TGap Theme. A TGap Theme has a name and is identified by two tables called <name>\_weights and <name>\_compatib. The contents of the weights table is needed to convert the class name calculated from the features into a class id to be stored in the dataset face table. There is a utility procedure provided with the wrapper which creates the weights and compatibility tables and populates the tables with entries for each class based on finding distinct values of class names from a table of data. This enters the weights and compatibility values for all classes as 0 and so need to be updated with real values before proceeding. Note that this process also creates a pseudo-class called 'Hole' which is used to represent any holes in the coverage and is what allows the process to continue even if holes exist. For example using the following SQL to automatically create class and weights tables for the theme called 'MY\_THEME' based on a table called MY\_AREAS with a column CLASS\_NAME:

```
call LSL_RTW_TGAP.GENERATE_THEME_TABLES('MY_AREAS', 'CLASS_NAME', 'MY_THEME');
```

## Create the TGap dataset using the TGap Wrapper

Start the TGap Wrapper and connect as the user who owns the data.

**Hint: If you cannot connect to Oracle using the standard tab (this uses the OCI driver), try clicking on the advanced tab (this uses the thin driver) and enter the hostname as the machine name where the database is running.**

Specify that you want to create the TGap dataset, then select the Radius Topology manifold that you want to convert to a TGap dataset. Only planar topology manifolds are listed here.

Next, enter a dataset name. This name is used to prefix the node, edge, edgegeo and face tables. The tables will be created but if they already exist then they will be emptied. As it is used to prefix a table then the name must contain only alphanumeric characters or underscores and must begin with a letter.

Next, select an existing TGap Theme. A theme is defined by two tables <theme\_name>\_compatib and <theme\_name>\_weights. Names are only listed here if the two relevant tables exist. The

classes in these tables should match the classes contained within the data that was used to create the Radius Topology manifold. If the theme tables do not exist then you must first create the theme tables (see above). If you have created the tables while the wizard is open, then hit Back then Next to see the theme listed.

Lastly, you need to specify how to find the class name for each feature to be created. This is usually specified using a column on the original table(s) of data and should contain the class name to use for each row of data. If more than one table was used, then the column name should be the same.

A more advanced option is to create a PL/SQL function which takes as input columns from the feature table and returns a class name. You must create this function yourself and specify it on this page, passing the column names as parameters. For example, if the class name can be calculated using a function called MY\_FUNCTION by using columns 'TYPE' and 'THEME' then in the function should be entered here as: MY\_FUNCTION(TYPE, THEME)

When you select 'Run' on the summary page then the process will create the TGap dataset.

### **For constrained TGap, enter region values in the face table**

The face table is created with the region value set to 0. In order to perform constrained TGap generalization then you must set the region values to meaningful numbers.

### **Generalize a TGap dataset using the TGap Wrapper**

Start the TGap Wrapper and connect as the user who owns the data.

**Hint: If you cannot connect to Oracle using the standard tab (this uses the OCI driver), try clicking on the advanced tab (this uses the thin driver) and enter the hostname as the machine name where the database is running.**

Specify that you want to generalize a TGap dataset, then select the correct dataset. Datasets are listed by looking for sets of face, edge, edgegeo, and node tables with the same prefix.

Next, select whether you want to perform normal or constrained TGap generalization. Constrained TGap can only be used if there are correct region values set in the face table.

Next, select an existing TGap Theme. A theme is defined by two tables <theme\_name>\_compatib and <theme\_name>\_weights. Names are only listed here if the two relevant tables exist. The classes in these tables should match the classes contained within the face table. If the theme tables do not exist then you must first create the theme tables (see above). If you have created the tables while the wizard is open, then hit Back then Next to see the theme listed.

When you press 'Run' on the summary page, the dataset will be generalized. Note that this is not a process that can be aborted: You must wait until it has completed. If the dataset has already been generalized, then the process will stop with a suitable error message.