Topology Checker 1.00

A Tool for Batch Topology Checking

Table of Contents

1 Introduction

Topology Checker 1.00

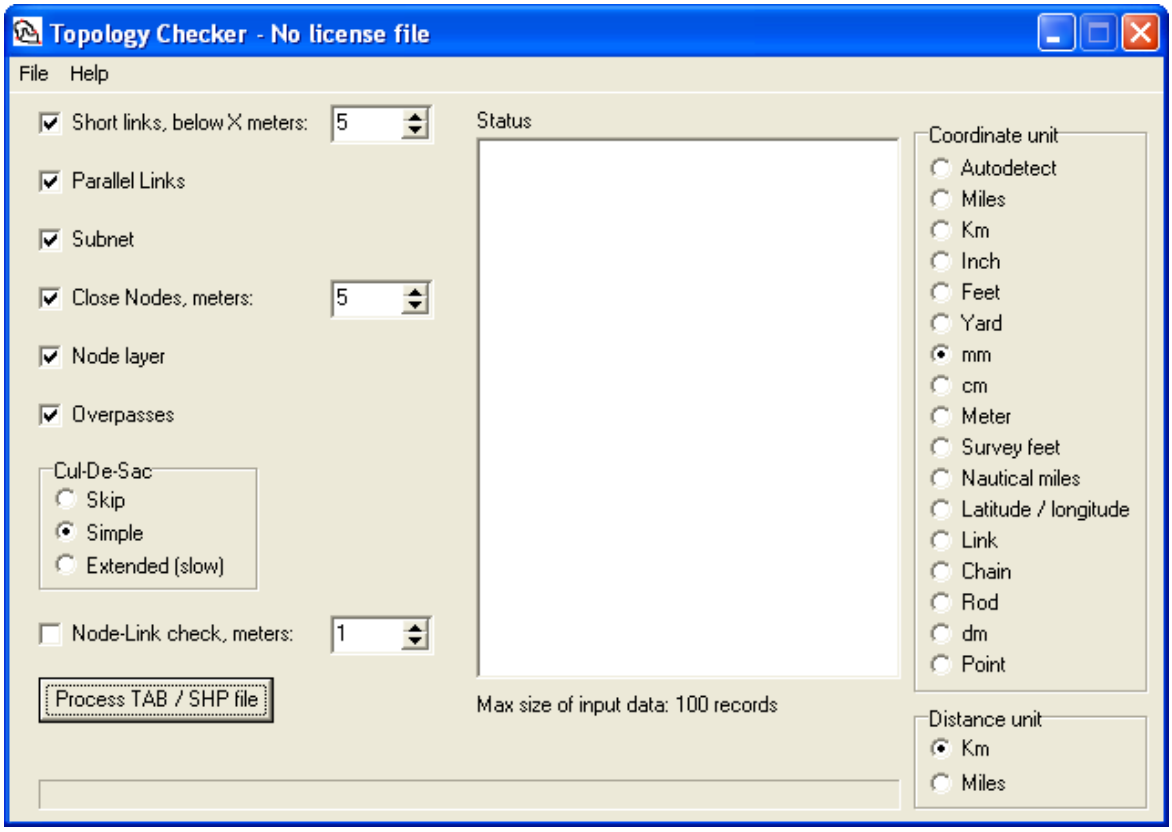
This application will check a street network for various topological problems. Normally a street network is connected by having exact snap between links, but the data is typically entered through a manual digitization process and this leads to many errors. Topology Checker helps pointing out most of these, but will NOT change your data.

When you run the application, start by setting the options and the select a TAB or SHP file. Processing will start immediately.

If you choose autodetection of coordinate unit, this will be read from either a PRJ file (SHP format) or directly from the MAP file (TAB format).

Distance unit (lower right corner) determines, if the 3 parameters entered are meters or thousandths of a mile. When distances are a part of the output files, these will also be either km or miles.

All options are stored in an INI file between each run.



Without a license file, the application is restricted to testing datasets with up to 100 records.

2 Problem detection

Several different kind of possible problems can be detected. A key term is "nodes", which in this context means the start or end of link. A node is typically shared by several links. A link is the same as a record in a SHP or TAB file.

- **Short Links**

Short links will typically result in many other kind of possible problems being marked. Use a value of 5 meters or less for detailed street networks.

- **Parallel Links**

Parellel links are 2 or more links that both start and end at the same 2 nodes. This may be the result of the same street having been digitized twice.

- **Subnet**

Subnets are isolated links, not connected with the rest of the network. Route calculations between different subnets will fail, so subnets should be avoided. In case of islands, add ferries for instance.

- **Close Nodes**

This will locate pairs of nodes within a short distance of each other, that are not directly connected with a link. This is typically a result of missing snap. Use a value of 1-10 meters depending on the level of detail in the data.

- **Node Layer**

The node layer simply returns a new layer with all nodes in the network and their valency. Valency is the number of links connected to the node. Creating a thematic map on the valency can be helpful in locating problems.

- **Overpasses**

This will generate a list of pairs of links that intersect at other places than the start / end. This should normally only occur where there is a bridge.

- **Cul-de-Sac**

This will detect Cul-de-Sac situations in your data. Either in simple mode (type 1-2) or extended mode (type 1-4):

1 = Cul-de-Sac
2 = Loops
3 = Bridge (in graph terms)
4 = Subnets

Type 4 are subnets that would be left, if bridges were removed, so this is a little different from "Subnets" above.

Type 1 and 3 which are also one-way streets will be a problem in any street network used for routing, since it prevents routing from either A > B or B > A, if A and B are on each side of the link in question.

- **Node-Link**

This will perform buffer operations around all nodes and test if the links within the buffer are also connected to the node. Use a small buffer such as a few meters for this check.

All results from the 8 analysis' above are stored as individual layers (TAB or SHP format), that is easy to bring up in your GIS together with the input file.
Distances / length reported as part of "Close Nodes"

- **Network Report**

This is a text file ("network_report.txt") generated during the import and besides various information about the process, it also reports on (possible) errors on individual links:

1: Object is marked for deletion (TAB format only) / not geocoded
2: Object is not a polyline
3: Object has >1 parts
4: Object has zero length
5: Object is a loop link (same as Cul-de-Sac, type 2)
6: Object has duplicate vertices (may mean self-intersecting)
7: Object is self-intersecting
8: Object has a very sharp turn, almost U-turn like
9: Object has only 0 or 1 vertex

3 Performance

We have executed a test on a street network with 191,000 links:

	Time, sec
Import (always required)	10
Cul-de-Sac (simple)	0.03
Parallel links	0.06
Subnets	0.2
Short links	0.7
Close nodes	1.4
Nodes (valency)	2.0
Overpasses	19
Node-link	696
Cul-de-Sac (extended)	699
Total	24 min

The test was executed with default parameters on an AMD64 3.0 GHz computer.

4 Strategy

As can be seen from the performance 5 page there is a big difference between running time of the different options. So we suggest that as many errors as possible are corrected while only testing for those problems, that are fastest to detect.

Run the application with only the "short links" option checked.

1) Start with the network report and see, if there are any problems located here. Edit the network.

2) Short links in the network below a threshold value, such as 5 meters, give reason to many false problems being located, so start with the short links and if possible remove such (merge with other links).

3) Now test for all problems except node-link and extended Cul-de-Sac. After you have edited the network you can go back and test again.

4) Now test for all problems. After you have edited the network you can go back and test again.

5 Sample data

The sample data supplied (SHP and TAB format) has been edited to illustrate most of problems, that TopologyChecker can detect.

They have <100 records to allow testing without a license file.

Run the application and then open any of the workspaces: "Demo.wor" for MapInfo Professional and "Demo.mxd" for ArcGIS.

6 Command line mode

It is possible to run the application from the command line by calling it with the name of the input file:

```
topologychecker sample_shp\network.shp
```

The application will terminate once finished.

7 History

23. Feb 2006: Version 1.00