RW Studio 2.03

A Network Analysis Application

Table of Contents

Part I

Introduction

1 Introduction

RW Studio 2.03

The Studio application is a standalone application, which can be used in combination with your favourite desktop GIS to do routing calculations.
It is built on top of our RW Net 4 SDK and makes a lot of the functionality available from an easy to use interface.

The file format is the same as RW Net 4, so it is also possible to use the application for importing data into FleetEngine etc.

Main features

- Import from SHP and TAB files
- Import from MS SQL, Oracle and PostgreSQL
- Export to SHP, TAB, MIF, GeoJSON, GML, CSV
- Vehicle type editor
- Network statistics
- Distance matrices
- Isochrones
- Service Area calculation
- Spatial Join
- Topology checking

Routing features

- Shortest / fastest route
- Dynamic segmentation
- One ways
- Turn restrictions
- Scalar restrictions (weight, width etc.)
- Bitwise restrictions (avoid bridges, toll roads etc)

1.1 Feature Matrix

Versions

Product	Network size, links	Point data size	Import from file	Import from DBMS	Topology checking	Route calculations	Advanced routing
Free	3,000	5	Yes	No	Yes	Yes	No
Standard	500,000	300	Yes	No	Yes	Yes	No
Pro	1,000,000,000	∞	Yes	Yes	Yes	Yes	Yes
Academic	50,000	∞	Yes	Yes	Yes	Yes	Yes
Topology Checker	1,000,000,000	0	Yes	No	Yes	No	No
Import	1,000,000,000	0	Yes	No	No	No	No

An *Academic* license is only for research activities at university level. Strictly no commercial activities. Proof required.

The *Topology Checker* license is only supplied to previous owners of the Topology Checker application. We do not sell it anymore.

An *Import* license is supplied to users of FleetEngine, so they can import data into the application.

Currently no routing features in RW Studio 2 are tagged as "advanced", but they will be added in later releases.

1.2 Licensing

A license for RW Studio 2 includes up to 5 activations of the software.
Once all 5 has been used, you can not activate it anymore.
It is not possible to de-activate.

A license includes updates for 12 months.
Software releases after that period can not be activated.

A license includes support for 30 days.

Support & updates can be bought for one year at a time.
This gives you access to a new serial number, allowing you to register again.

The license is personal unless anything else has been agreed.

1.3 Registration

You can register the application online by entering the serial number, that has been sent to you by e-mail.
This will open up for additional features.

There is a maximum number of registrations available per serial number, so don't share it with colleagues etc.

1.4 System Requirements

Operating System
64 bit Windows.

Hardware
A fast CPU and hard disk is a good idea, but there are no specific requirements.
Networks require app. 55 bytes per link. An example: A network with 1 million links uses 55 MB.
This is when opening spatial index, but not caching coord3 file.

Additional software
You will need a desktop GIS to prepare input and view output:
ArcGIS, MapInfo, QGIS and many more can be used.

Database drivers for MS SQL, Oracle and PostGreSQL are built into the application, so no further client libraries are required.

Internet
You need an internet connection to register the software online.

1.5 Settings Folder

The settings folder is where these files are stored:

- Configuration file (INI file style)
- Log file
- License file (after successful registration 31)

Use the menu item to open the folder.

The configuration file can be edited manually, but please do so with great care and only do it while RW Studio is not running.
It keeps a list of all connections and networks and also the global settings.
Decimal numbers may occur in a few places - here . is always used as decimal point.

Information about each network is found in the network folders.

1.6 User Folder

The user folder is where these files are stored:

- 2 folders with sample data
- Networks after import from DBMS (default location)

Use the menu item to open the folder.

Part II

Options

2 Options

2.1 Output Format

This allows you to choose a file format for export 29, calculations 24 and more:

- SHP (default)
- TAB
- MIF
- MIF - UTF8 (for 64-bit MapInfo)
- CSV
- GML 2.1.2
- GeoJSON

SHP and TAB files are both restricted in size (2 GB), which means they may not be able to hold all kind of output, such as large distance matrices.

The other fileformats are written directly to disk and can be as large as your disk allows.

Depending on the output formats you plan to use, this information is needed for generating output:

SHP	PRJ file
TAB / MIF	Coordsys clause
GML / GeoJSON	EPSG code

When importing from a DBMS, the generated network automatically contains the required information for all formats.
When importing from SHP, only the PRJ is present.
When importing from TAB, only the Coordsys is present.

As a help we have added two buttons for the most commonly used coordinate systems, when importing from SHP or TAB.

If you need to generate a network from a SHP with an unusual coordinate system and export to non-SHP, then you will need to find the EPSG and/or coordsys clause on your own.

2.2 Unit

This allows you to choose a distance format for calculations:

- Km
- Miles

This is used for output files containing distances and/or speed fields.

Time is always calculated in minutes.

Part III

Connections

3 Connections

This is the main tree view of:

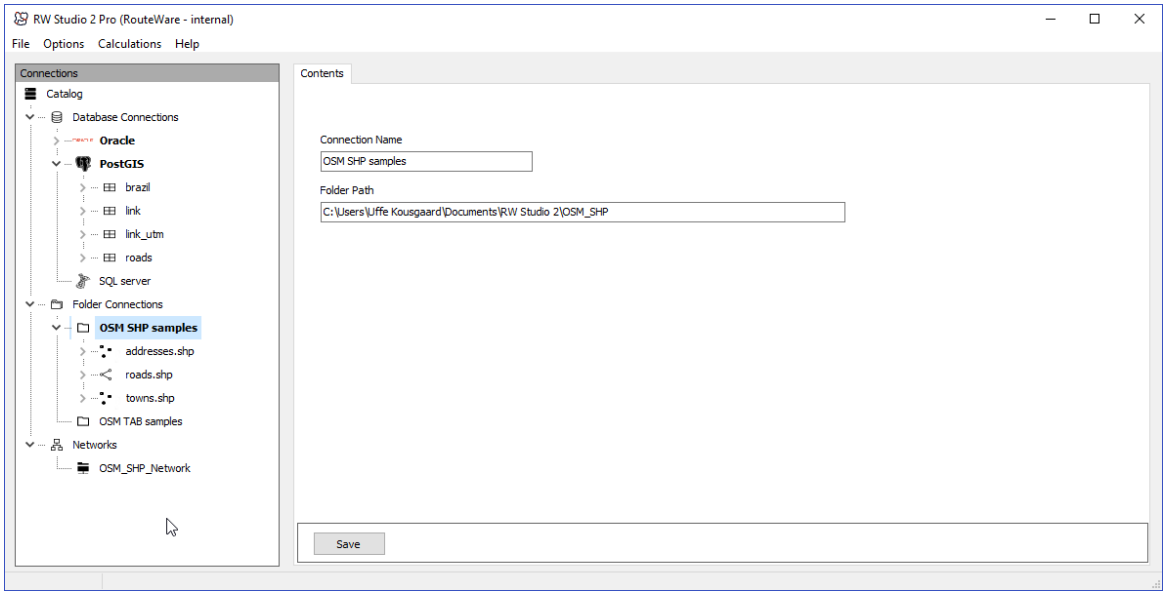
- Database connections (MS SQL, Oracle, PostgreSQL)
- Folder connections (SHP, TAB)
- Networks

A database connection is a single database, which may hold multiple tables inside. Right-click to open and view available tables.

Similar a folder connection points to a folder on disk, which may contain a mix of SHP and TAB files.

Networks contains data that has already been imported.

Generally right-clicking gives access to adding, copying, deleting etc.



3.1 DBMS

These DBMS are supported:

- MS SQL
- Oracle (locator / spatial)
- PostgreSQL (with PostGIS extension)

All connections are defined by their

- Server (IP-address / servername)
- Database
- Username
- Password

For Oracle XE, use:
server = <server>:<port>:XE
database = XE

For SQL Server Express, use:
server = <server>\SQLExpress

Passwords are stored in the configuration file, in an encrypted form.

If a table has been uploaded to the database from MapInfo, it will automatically look for a mapcatalog and get the correct coordsys string from there, during import.
Without a mapcatalog it will translate the EPSG code into a coordsys string, from an internal lookup list.

Part IV

Datasets

4 Datasets

A street network is the basic source for the routing. Generally the requirements are quite simple:

Polylines drawn from one one intersection to the next, with optional information about oneways, name of the road, speed limits etc.

The rules are:

- They need to snap
- They need to split at intersections
- The network should be plane unless there is an overpass (or use Z-levels, see below)
- You should avoid subnets 29 (islands)
- You should avoid very long links, which have a negative impact on speed of certain calculations

Z-levels

These can be used to express the Z-level of start and end of a link, to make sure vehicles do not "jump" off bridges.
It is a value in the range -9 to 9, with 0 being used for ground level.
Many datasets use these: TomTom, HERE, ITN etc.

OpenStreetMap do not use Z-levels, the exception being our sample dataset, where it has been added for illustration.

Coordinates

Two kind of coordinate units are supported:

- Spheric / Latitude-longitude (global / local)
- Cartesian / Projected (local)

When working with spheric coordinates, all distance calculations are performed using great circle distances and the Earth is considered a perfect sphere with radius 6378.13 km.

When working with Cartesian coordinates, all distance calculations are performed using straight Pythagoras formula. Several different Cartesian units are supported.

Global projections such as Web Mercator and more are not supported. They use meters as coordinate unit, but the scale factor depends upon the latitude.
If you use it, you will not get any error messages, but the lengths will come out wrong, when you are not on Equator.

4.1 Attribute field

The attribute field is key to defining one-way restrictions etc. It is an integer in the range 0 to 65535.

If you don't want to set it up properly at first, just create a field with all 0's and use that as attribute field.

- Road class, 0-31 (5 bits)

These have no predefined meaning, but the speed for each class can be defined through speed files 21.

- Hierarchy, 1-5, (3 bits, 32-64-128, bit 5-7)

Not used in RW Studio, but can be reported in statistics 22.

- No-drive through, true/false (1 bit, 256, bit 8)

This can be used to define areas, where you are not allowed to drive through to get to the target.

- One-way, To-From direction is not allowed, (1 bit, 512, bit 9)

Similar to "FT" in some datasets prepared for Network Analyst.

- One-way, From-To direction is not allowed, (1 bit, 1024, bit 10)

Similar to "TF" in some datasets prepared for Network Analyst.

If both bit 9 and 10 are set, the link is closed for driving.
Similar to "N" in some datasets prepared for Network Analyst.

- Roundabout, true/false (1 bit, 2048, bit 11)

Can be used in creating driving directions.

- Non-driving link, such as a ferry or car-train, true/false (1 bit, 4096, bit 12)

Can be used in creating driving directions.

- True if not allowed to make U-turns at the From-end of the link. (1 bit, 8192, bit 13)

- True if not allowed to make U-turns at the To-end of the link. (1 bit, 16384, bit 14)

- SkipInSearch, true/false (1 bit, 32768, bit 15)

Set it true for links, where you don't want routes to start or end.

An example:

A motorway of class 1, which can only be traveled in the direction of digitization and should be skipped in searches:
1 + 512 + 32768 = 33281.

4.2 Roadname field

A roadname field is mostly used in the creation of driving directions (such as in FleetEngine), but can also be used in a spatial join 27.

Names gets stored in unicode format, once imported.

When importing from a SHP/DBF file, you will have to specify the codepage, if both CPG file and languageID of the DBF (byte 29 in the header) is missing.
For all other data sources it happens automatically.

4.3 Limit fields

It is possible to define 2 kinds of route restrictions for links in the network:

- A scalar quantity such as a maximum weight, height, width etc. If the limit for a certain link in the network is 100 and you calculate a route for a vehicle with a value >100, that link will be avoided in the route. It is mandatory to scale your limits into the 1-255 range.

- A ***bit pattern*** for defining special links such as ferries, toll roads etc. which you may want to avoid in your routing. If the limit for a link is 3 = 00000011 it may mean it is both a ferry and a toll "road". If your value has either bit 1 or 2 set, that link will be avoided in the route.

You can have 3 scalar limits and 1 bitpattern with 8 bits in RW Studio.

For both types, a link value of 0 means no limitations at all.

4.4 External ID field

This is a globally unique field, which is typically used as a reference for turn restrictions.
All major datasets includes one and it can be a string or large integer.

4.5 Turn restrictions

There are 2 types of restricted turns:

- Banned turns
- Delayed turns

Normally you will be using banned turns only, since for most normal routing purposes, setting different road speeds for road classes are sufficient for giving a realistic route choice.

Generally you can change choice of route A LOT by using wrong values for delays, so take care.

If you add a turn restriction, where one of the links making up the restriction is already marked as one-way, it is skipped.
You will not notice this unless you export the turns.

Fileformat

Turn restrictions are stored in a text file, space-delimited.

The format is one or more lines, where each line stores one restriction with parameters.

7 different types of restrictions are possible:

0: Simple Turn restriction, 2 external link ID's + 1 cost value
1: Simple Turn restriction, 2 link ID's + 1 cost value
2: Turn Standard, coordinates for node (rarely used, see RW Net 4 documentation for details)
3: Mandatory turn, 2 external link ID's
4: Mandatory turn, 2 link ID's
5: Complex Turn restriction, >2 external link ID's + 1 cost value
6: Complex Turn restriction, >2 link ID's + 1 cost value

Example:

// Comment
0 A4003234 A4003127 -1
1 456 230 -1
2 -77.024098 38.902711
3 A4003234 A4003127
4 456 230
5 A4003279 A4003234 A4003127 -1
6 89 456 230 -1

Lines starting with // are ignored as comments.

If you use turn restrictions with external ID's (type 0, 3 and 5), make sure you have opened the network with an external ID.

Type 0, 2, 3 and 4 gets translated into one or more type 1 during import and type 5 gets translated into type 6 during import.

If you export the turn restrictions, you will only get type 1 and 6.

Part V

Cleaning Data

5 Cleaning Data

Before importing data, it is a good idea to clean them from the most obvious issues.

We have created two applications, which does this for MapInfo and ArcGIS. It performs these steps:

1. Test for ungeocoded records and delete them
2. Delete objects which are not LINE or PLINE (mapinfo script only)
3. Disaggregate multi-part objects
4. Remove duplicate and consecutive nodes
5. Delete polylines with 0 or 1 node
6. Split loops / circular links in two

You can adapt the scripts if you want to - full source code is included.

Find the mapbasic and arcpy script in the installation folder.

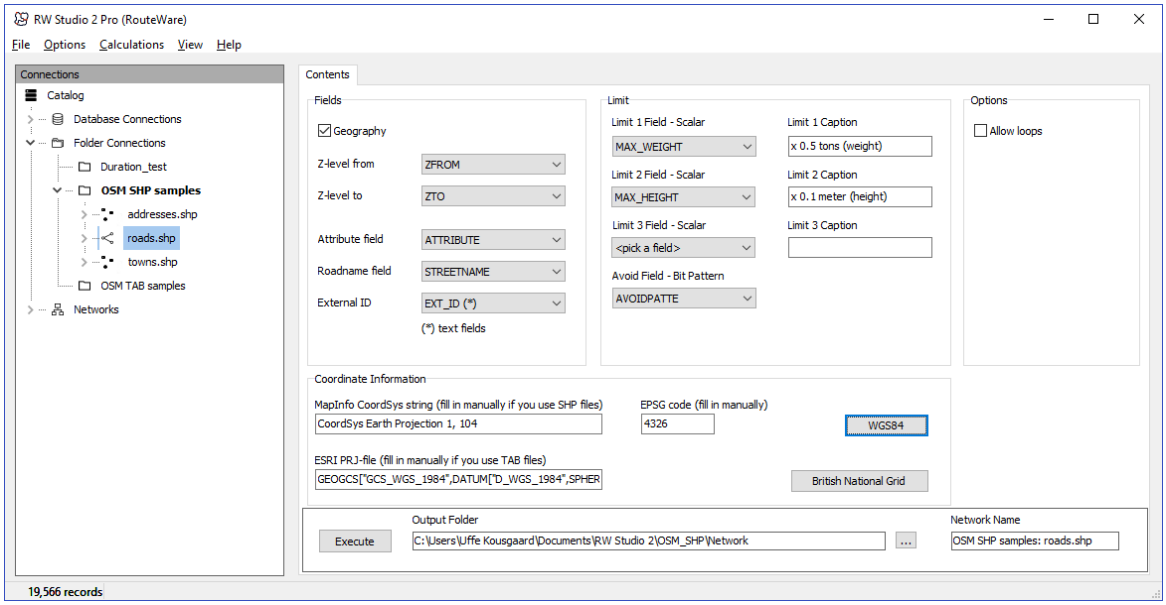
Part VI

Import

6 Import

Right click a connection to open it. Now select a poly-line table and right-click to import from it.

You should now specify which fields in your dataset hold each piece of information:



Allow loops should normally be left unchecked since loops are not allowed when doing routing.
- If unchecked and there are loop links in data: They will be reported as errors in network_report.txt
- If checked and there are loop links in data: They can be exported in one of the topology checking 29 functions.

Finally set output folder, before:

Click the Execute-button

After you have imported a network, it gets automatically added to the list of networks 20.

If you started by cleaning the data 16, there should be no issues listed in the file "network_report.txt", which has been generated in the output folder.

Specific notes about importing from Folder connections

SHP files should have a PRJ file.
Some SHP files may require that you specfiy the codepage, if the autodetection isn't possible.
SHP files should be 2D ("PolyLine"), rather than 3D ("PolyLineZ / PolyLineM")

Part VII

Networks

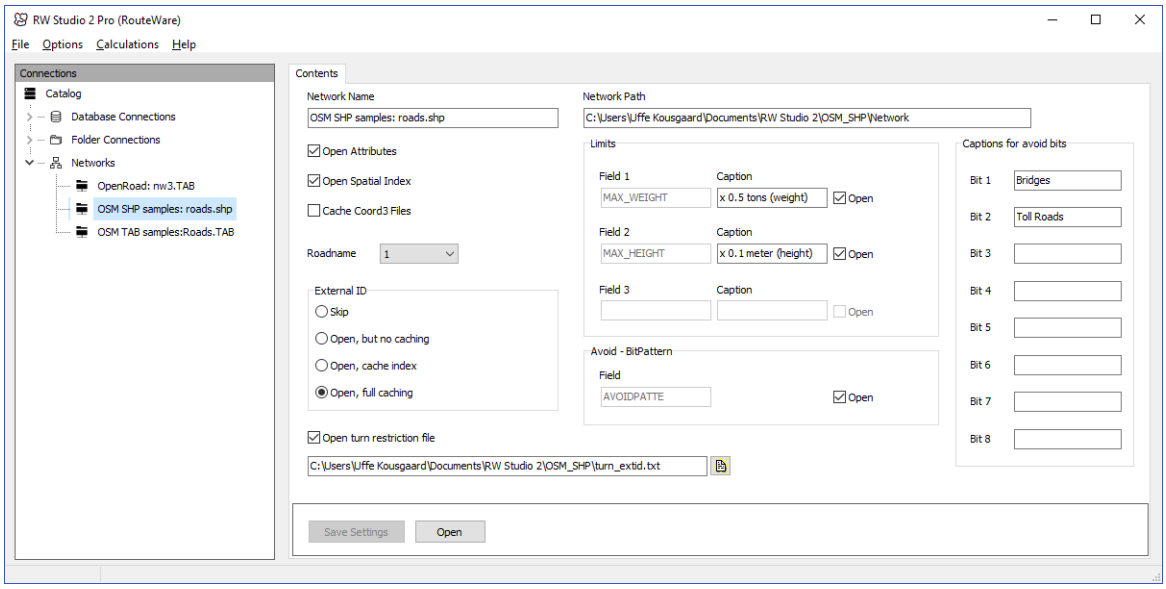
7 Networks

This is a list of networks and the settings you have defined.

Right-click to open. Only one network can be open at a time.

Settings for a network can only be edited, while it is closed.

Networks "remember" from which connection 8 they were imported, so you can choose to re-import and it will show a dialog setup, ready to import again, if your data source has changed.



The configuration file is *rwstudio2_network.ini* and is stored in the same folder as the network. Be careful if editing on your own and only do so, while RW Studio is not running.

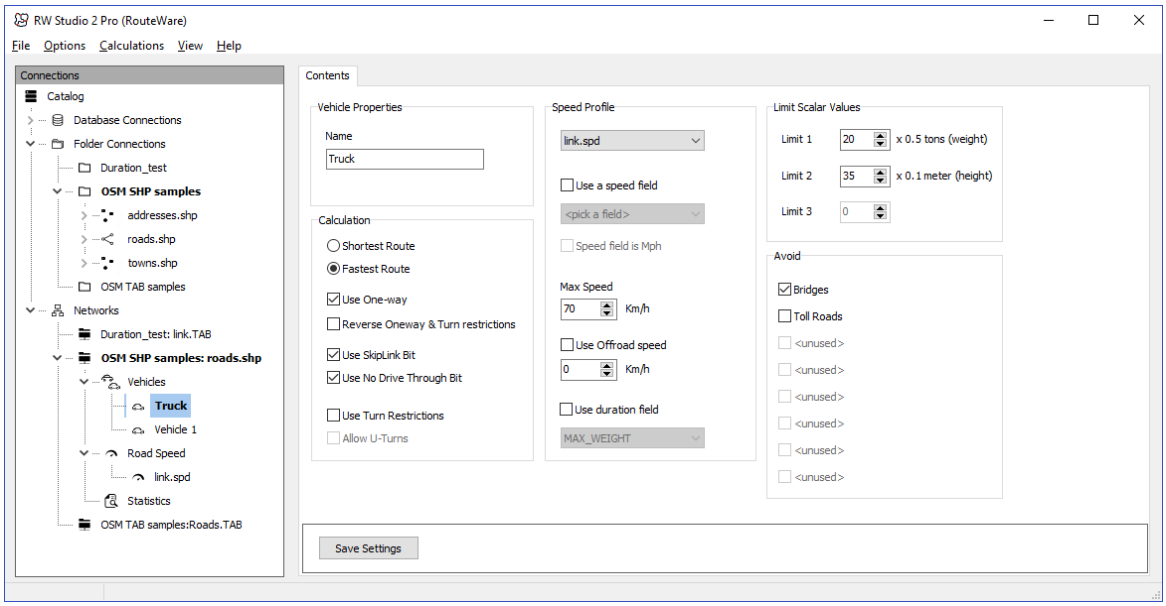
7.1 Vehicles

A vehicle is a set of properties, which together define possible routing on the network. Such as if oneway information should be used, turn restrictions, weight etc.

Multiple "vehicles" can be defined: Car, truck, pedestrian, bicycle etc.

You should choose one of the vehicles to be the active vehicle, the one used during calculations 24 (right-click to choose active vehicle).

In the example below we have a truck, which uses fastest route and acts according to one-way and turn restrictions. Speed profile is the generic, but with an upper limit of 70 km/h.
Weight is 20 x 0.5 = 10 tons.
Height is 35 x 0.1 = 3.5 meters.



Vehicle data is stored inside the configuration file for the network 20.

If you use a speed field, the presence of the original street network is mandatory. Otherwise the network can be used on its own.

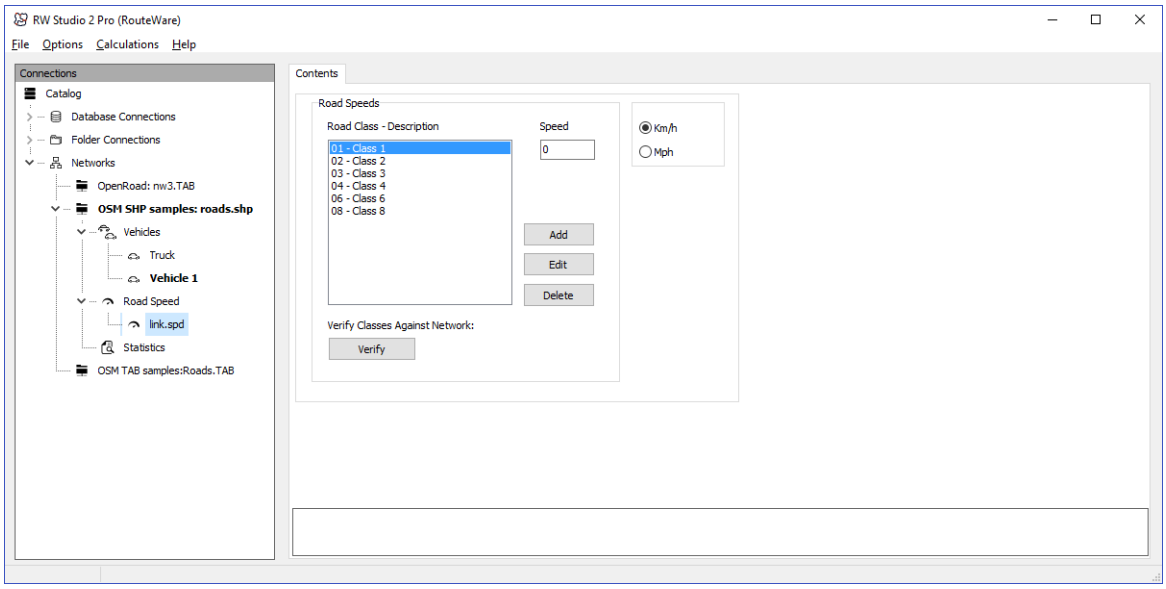
Similar if you use a duration field (in minutes). All values >0 in that field will be used, rather than the original speed for the class.

7.2 Road Speed

Road speeds are generally stored as a speed per road class, where road class is a number from 0 to 31 and part of the attribute 11 field.

The dialog allows you to add, modify or delete road classes and the corresponding speed.

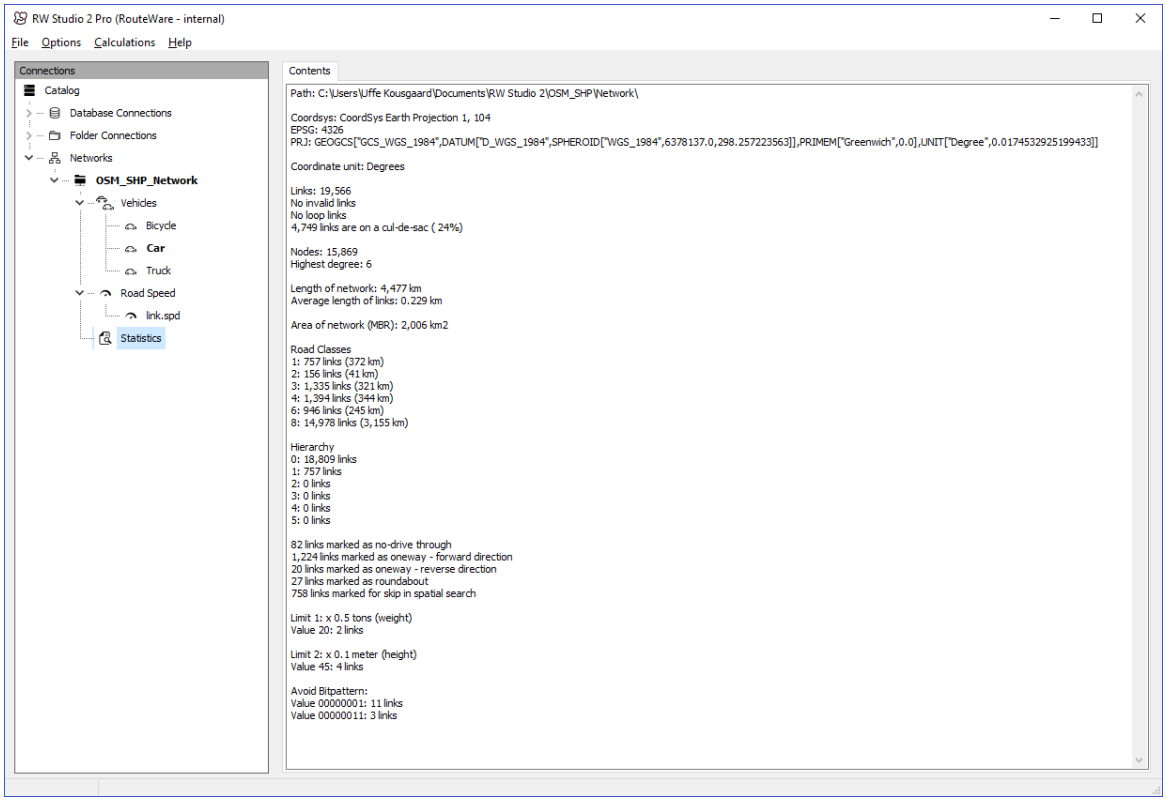
You can also define if the speeds are in km/h or mph. Changing this do NOT recalculate the values.



Information is stored in Spd-files, which have the same format as in RouteFinder 5 and RW Net 4, with the exception decimal numbers are not allowed in the editor above.

7.3 Statistics

After you have opened a network, this page shows a lot of useful statistics about the data. Use it for checking if something looks wrong.



Part VIII

Calculations

8 Calculations

Once a network has been opened, you can do various calculations.

At first we do distance matrices only. Later on these gets added:

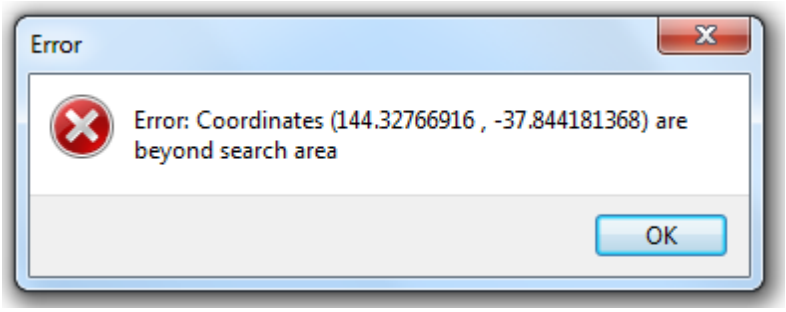
Nearest N centres
Travelling Salesman

Coordinates

For all calculations involving a point table as input, your table should be in the same coordinate system as the street network.
I.e. no coordinate transformations are done.

If you try to mix them anyway, you will get an error that the coordinates of your point data are outside the search area.

In this example the point table is in Lat/long data, while the street network is a local projection:



8.1 Distance Matrix

This function allows you to calculate a distance matrix between one or two tables, for all combinations.

Input:
SHP / TAB files with point objects.
A unique ID field should be present.
Projection should be the same as the network.

You can swap the tables, since it is faster, if the from-table has less records than the to-table.

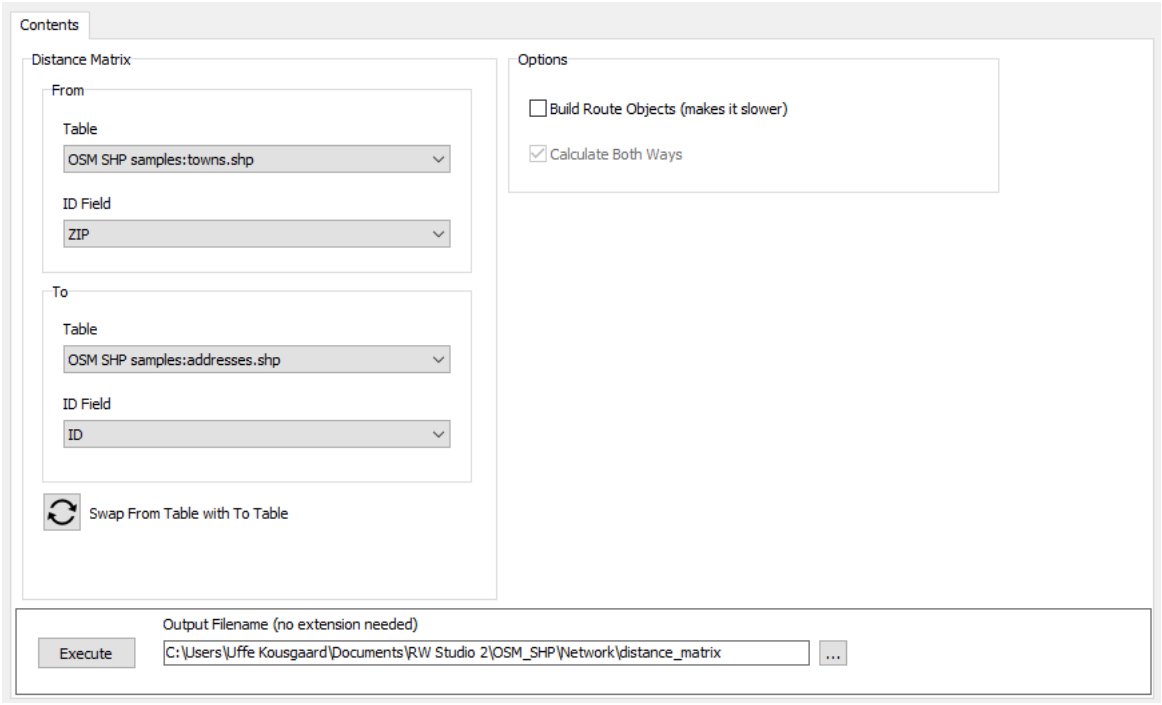
Options
"Build route objects" control if you are getting just the time and distance or also the actual route chosen.
This takes more time and generates much bigger files in large networks.

"Calculate both ways" is only relevant where the same table is used for both TO and FROM table. When disabled, you only get the distance from A to B (not from B to A), giving you a matrix of half the size in shorter calculation time.

Use it when there are no one-way streets or turn restrictions.

Output:
Distances are stored with 3 decimals, while time (minutes) have 2 decimals.

A progress bar estimates when the process finishes.



8.2 Isochrones

This menu item covers a wide selection of actual isochrone calculations and then the service area method, many of them sharing the same list of input parameters.

Methods:

- Linked-based
- Simple
- Voronoi
- Alpha shapes
- Service area

Input:

A point based table in either TAB or SHP format.
For most of the methods it is also required to select an ID field.

Steps can be either minutes or km, according to choosing either fastest or shortest route. This applies to all methods, except the service area.

"Add nodes" determines how many artificial nodes are created along the link, between the real nodes of a link.

For instance a 2 km link with addnodes = 0.5 would add nodes at 0.5, 1.0 and 1.5 km.
Low values may add a huge number of nodes and slow down calculations a lot and require more RAM.
Normally don't go below 0.1 km or set it to 0 for no additional nodes at all (large networks).

"Angle" is used for simple isochrones. Either set it to 0 or some low value such as 3.

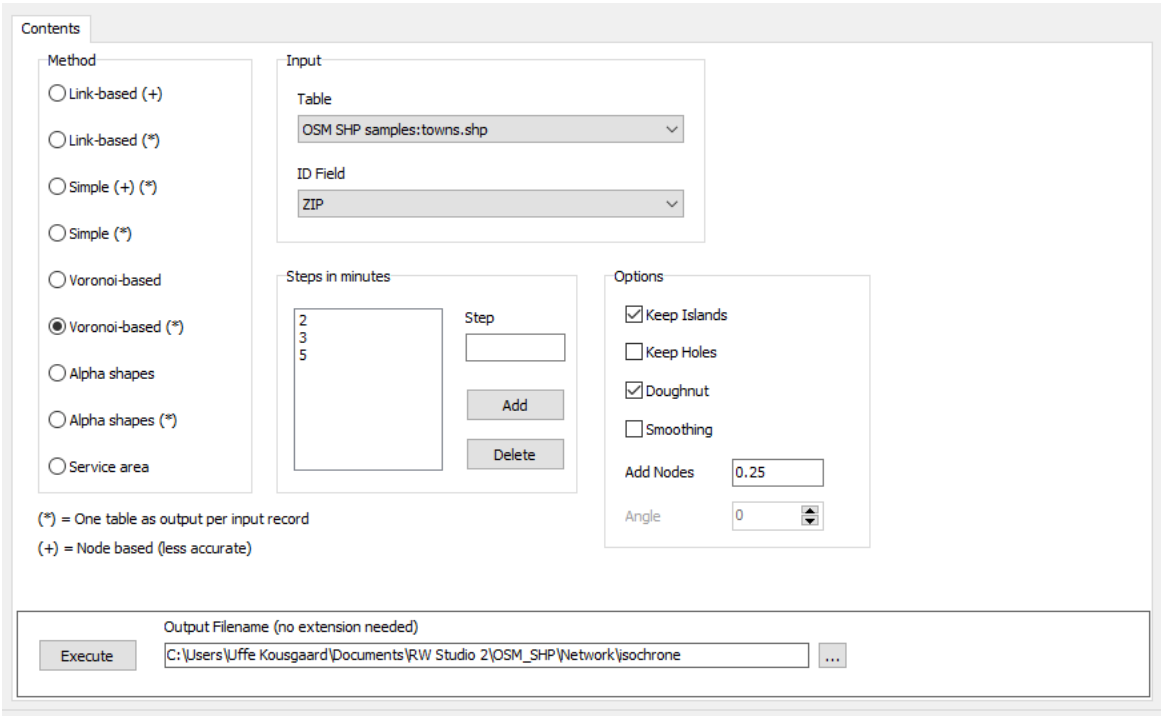
Smoothing can be enabled, but may give degenerate results in some situations.

Table of methods and possible parameters:

	Steps	Keep Islands	Keep Holes	Doughnut	Smoothing	Add Nodes	Angle
Link-based	X						
Simple	X			X	X		X
Voronoi	X	X	X	X	X	X	
Alpha shape	X	X				X	
Service area		X	X			X	

Methods marked with a (*) generate one table as output for each record in the input table. The output tables are all named with _ and the content of the ID Field.

The 2 methods marked with a (+) use nodes as starting point in the calculations, while the rest all use the more accurate dynamic segmentation.



Note: The 3rd method above was removed in build 2.2.0.5.

8.3 Spatial Join

The purpose of this function is to show how point data (addresses for instance) snaps to the road network.
This is so you can fix possible issues in data, but also for other purposes (such as snapping GPS data to roads).

Input:

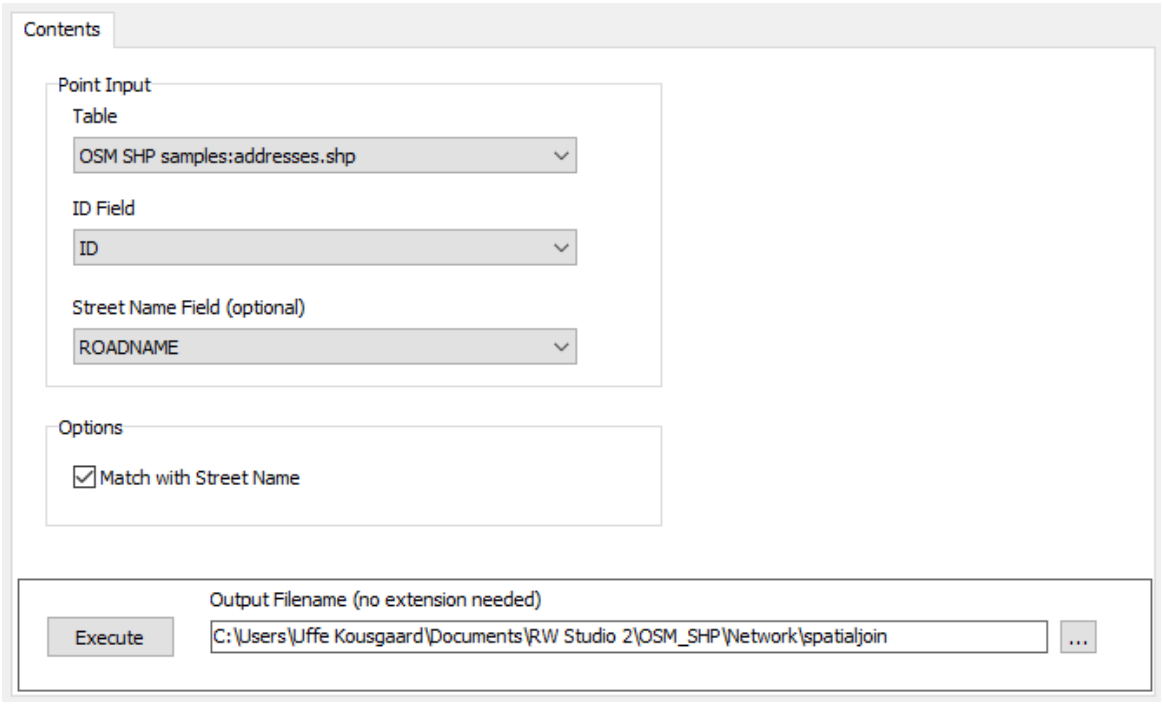
A point based table in either TAB or SHP format.
An ID field is also required to match the output with the input.

If choosing both a street name field and selecting "match with street name", it will search for the nearest link with the correct name.
If the name doesn't occur in the network at all, it will simply look for the nearest street.

If choosing both a street name field and **not** selecting "match with street name", it will search for the nearest link and report if the street name is correct.

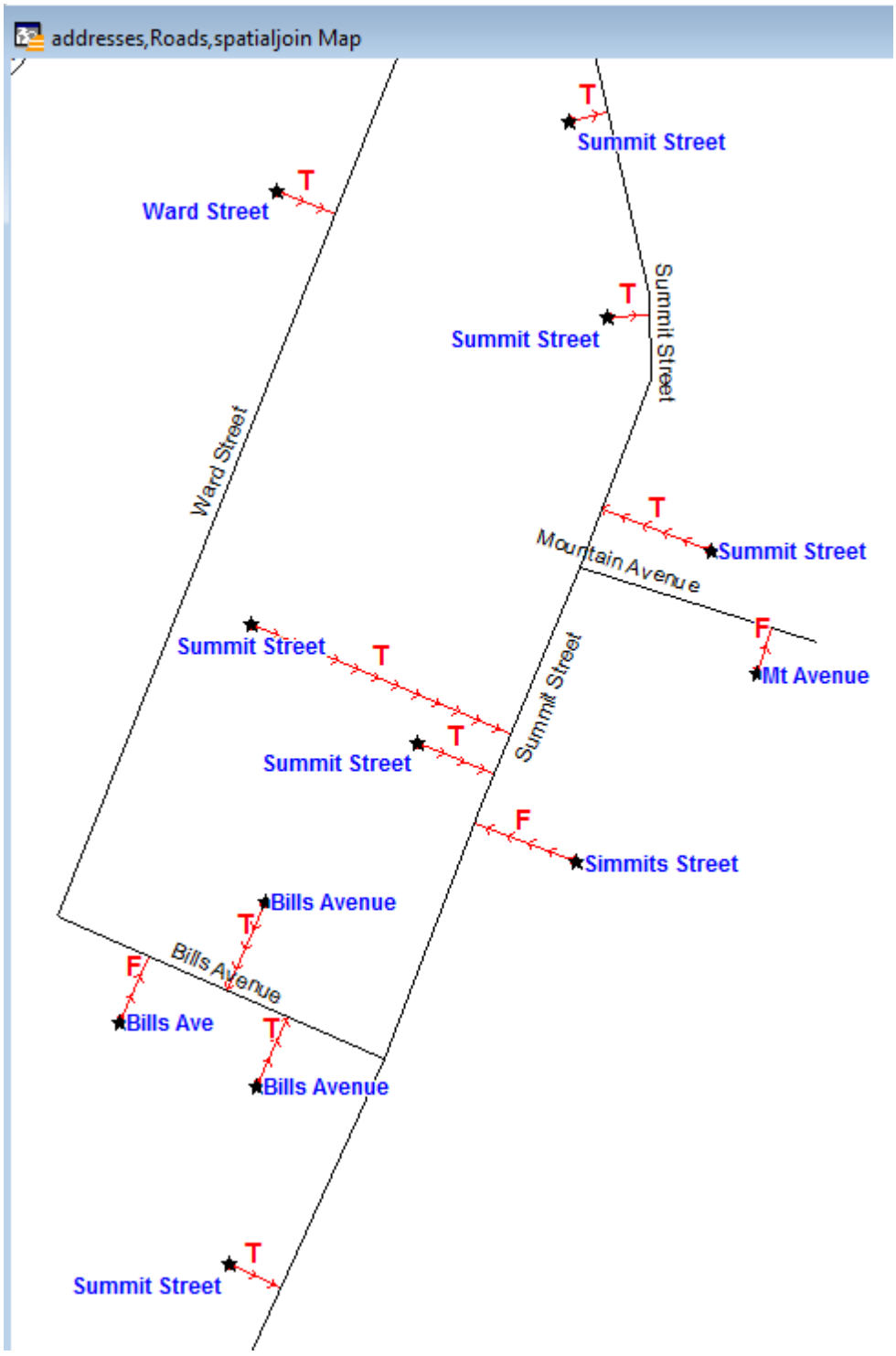
If a street name field has not been chosen, it will simply search for the nearest link.

In all cases, it will also report the location, the distance in km and side of road (1: Right side, -1: Left side).



Here is a map of the sample data and the result of a calculation (red).

T/F shows if the correct street name was found:



8.4 Network Export

This function will export the street network back to the GIS format 6 chosen.

- Links

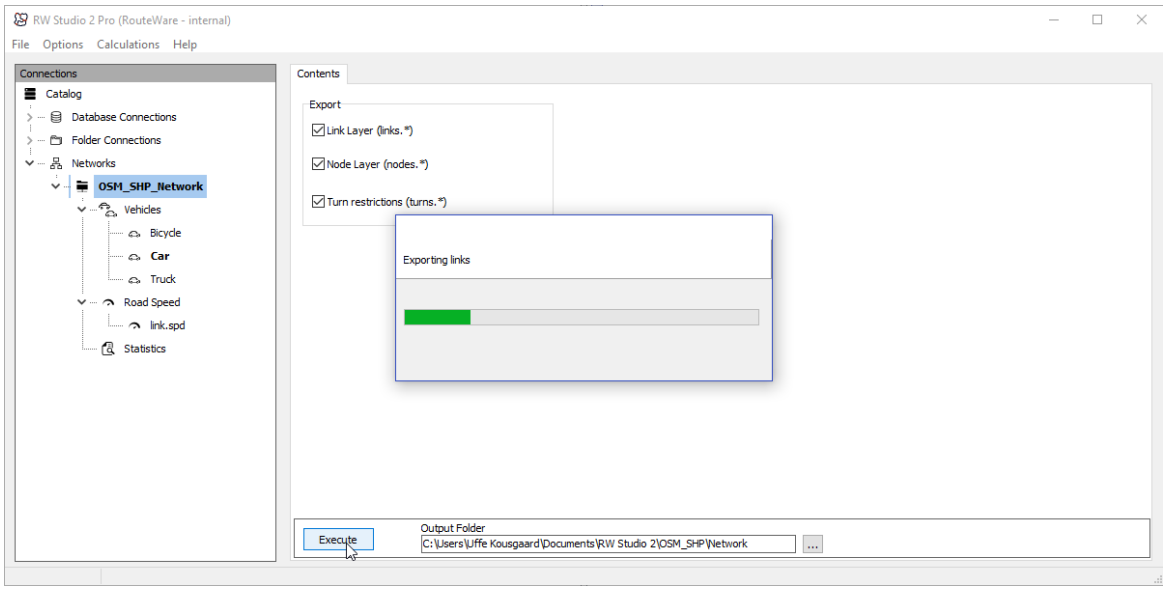
This closely matches the original input data, except information from the attribute field is also stored as individual pieces
of information and is easier to use for thematic mapping.
Information about Cul-De-Sac is also included as true / false.

- Nodes

The node layer simply returns a new layer with all nodes in the network and their degree.
Degree is the number of links connected to the node. Creating a thematic map on the degree can be helpful in locating problems.
Degree = 1 means a dead-end. Make sure they are where you expect them.

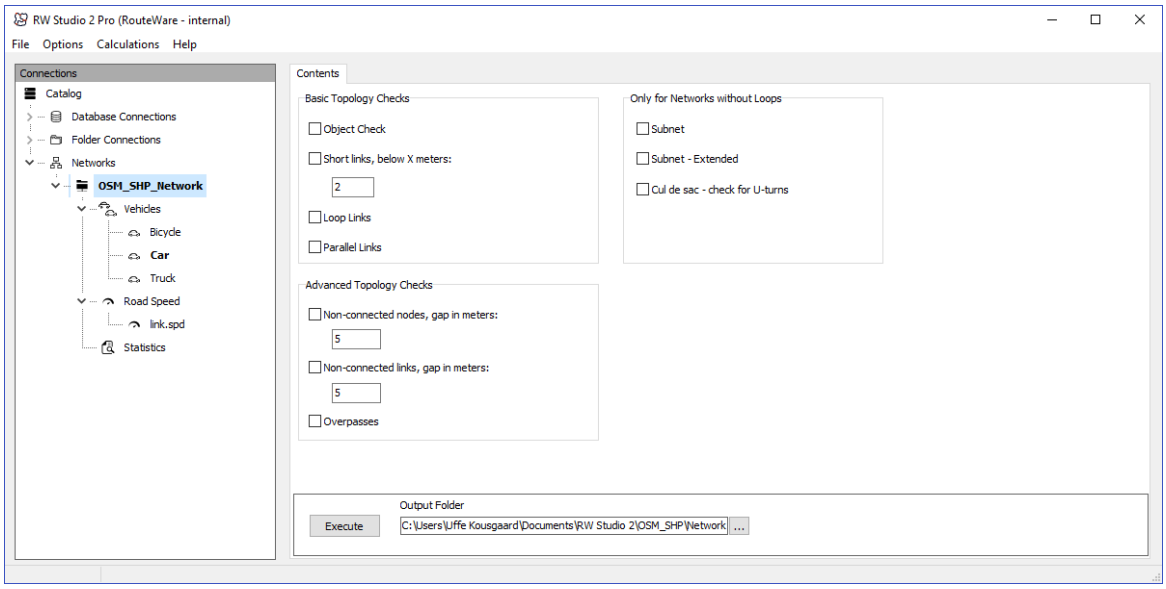
- Turn restrictions

If you have any turn restrictions, this allows you to export them in a format, which makes it easier to check them visually.



8.5 Topology Checking

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.
The functions here shall help you pointing out most of these, but will NOT change your data.



- Object check

These are possible issues related to single objects.
Sharp turns > 45° and self-intersections.

- Short links

Very short links (< 5 meter) is typically an indication of errors.

- Loop links

If checked and the network has loops, these shall be output as a separate layer.

- Parallel links

Parallel links are two or more links that all start and end at the same two nodes.
Especially when samelength = true it indicates an error.
Typically due to double digitalization of the same link.

- Non-connected 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.

- Non-connected links

This locates links within short distance of nodes, but without being connected.
Here the link typically need to be split in two and then snapped together.

Use a value of 1-10 meters depending on the level of detail in the data.

*** Don't use Z-levels during import when checking for non-connected. ***

- 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 / tunnel.

- SubNet

These 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 a ferry for instance.

- SubNet - Extended

These shouldn't occur at all since it means it is only possible to go from A to B, not from B to A.
It is usually due to errors in one-way settings. It can be a bit tricky to locate exactly where the problem is,
since a single error may highlight several links.

- Cul-de-sac - check for U-turns

When driving down a cul-de-sac you have to make a U-turn somewhere to get back again.
This function identifies additional links with the same problem as normal cul-de-sacs (already marked in layer "link").
Not errors as such, but may point to errors with oneway restrictions.

Strategy

Choose a different folder for output and then click "Go !".

With large datasets it may take a while. Especially "non-connected" and "overpasses" take a long time to run,
so do the first runs without these, fix as much as possible and then include them at some point.

Once you have made a significant amount of edits due to some of the first elements on the list, re-import the network and start over.
As you move along, there should be less and less issues left.

It is quite normal that the same error can trigger several of the checks above.

Part IX

TAB files

9 TAB files

MapInfo 15.2 introduced extended TAB files allowing you to let TAB files grow beyond 2 GB. They can be recognized by having the type NATIVEX in the TAB header.

At the same time both classic and extended TAB files started by being created with a default blocksize different from 512 ($0200) which was otherwise the standard.
They are not so easy to recognize, but the size is stored in byte 262-263 ($0106) of the MAP file. Inside MapInfo you can query tableinfo(tablename,49).

We support all variants when importing networks, but you can only use classic TAB files with blocksize 512 as input for calculations.

Part X

History

10 History

Version 2.03 (31. Oct 2018)
Build 2.3.0.6
Updated libraries.

Version 2.02 (1. Feb 2018)
Build 2.2.0.5
Case-insensitive check of user names etc. in license files.
Updated libraries.

Version 2.01 (24. Aug 2017)
Build 2.1.0.4
Fix for license file checking. You may need to re-activate.

Version 2.00 (26. May 2017)
Build 2.0.0.3

Version 2.00 BETA 3 (22. May 2017)
Added: Isochrones, Service Area calculation, Spatial join.
Added duration field for vehicles.
Several other fixes.

Version 2.00 BETA 2 (9. Feb 2017)

Version 2.00 BETA 1 (17. Jan 2017)
New major release.

Version 1.06 (18. Jan 2016)
Last 1.x release.

Version 1.00 (24. Oct 2012)