Instructions for installing and using Transport Link Scanner

This file describes how to install and how to use the Transport Link Scanner (TLS) model. It refers to the model version made available on 16 december 2015. This is a first draft of the instructions file.

Important note: the 16th December 2015 version of TLS is the culmination of many years of development to support various experiments. Because of its past the naming and structure of the model is not thoroughly consistent. Future updates will tackle this.

Date of this manual: 18 December 2015

Contents

1 Installation

System requirements are minimally a Intel Pentium (or compatible) PC with at least 1 Gb RAM. A 64 bit machine with at least 8Gb RAM is recommended. OS requirements are Windows Vista, 7 or 8. Windows 10 is presumably supported but currently (16/12/2015) not yet tested.

1.1 Required for installation

- Transport Link Scanner data (in TLS_data_16dec15.zip)
- Transport Link Scanner script files (in TLS_cfg_16dec15.zip)

- GeoDMS software. The open source GeoDMS software (see http://objectvision.nl/geodms/software). TLS will function with many recent versions of GeoDMS: the model has recently been tested successfully with versions 7.114 and 7.122.

1.2 Script editing software

Although the GeoDMS scripts may be altered using any text editor, we recommend using Notepad++ for editing (https://notepad-plus-plus.org/). Dms language definitions for Notepad++ are available in the file GeoDMS_npp_def.xml.

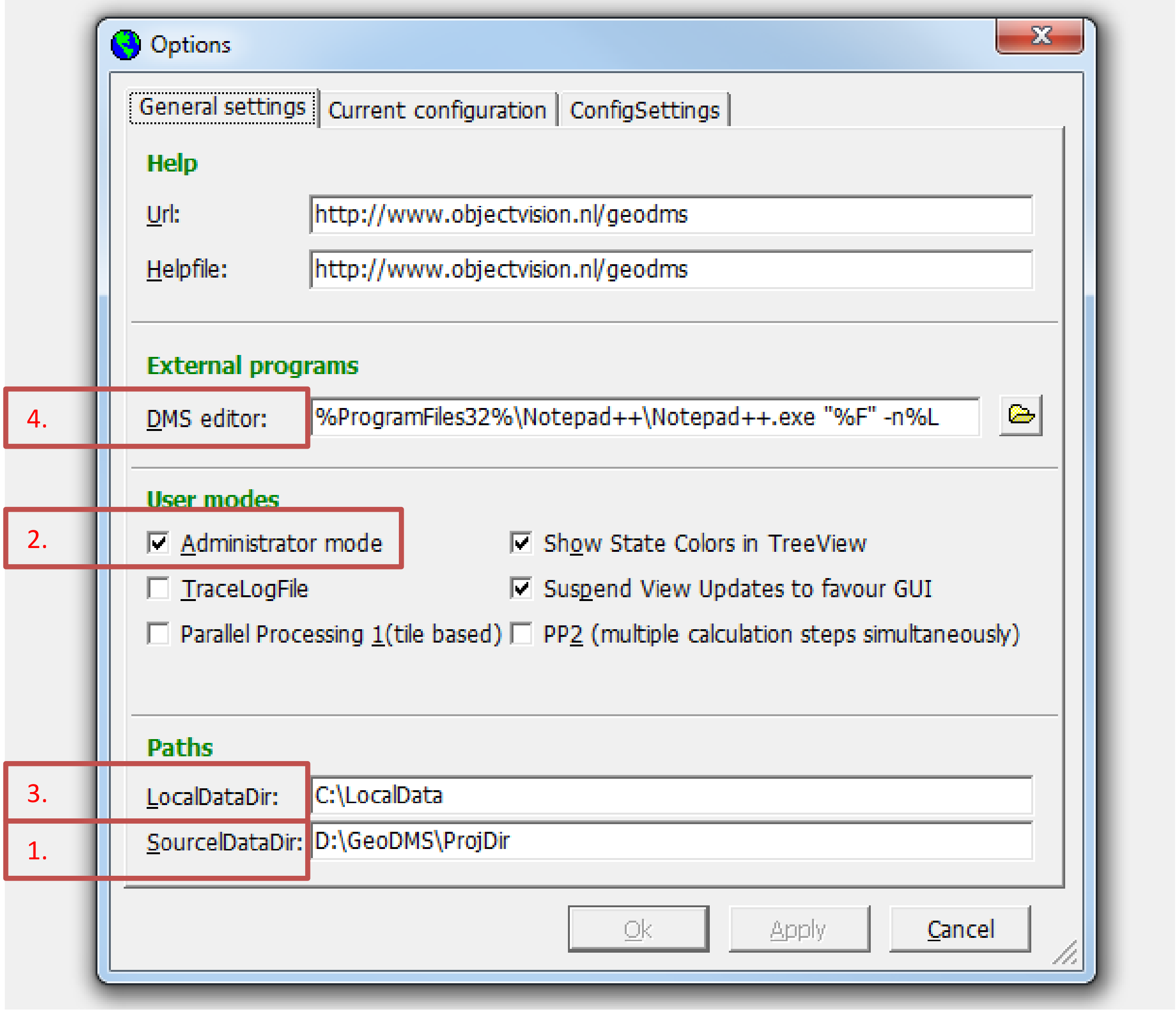
For information on using and working with GeoDMS we kindly refer to the ObjectVision website: http://objectvision.nl/geodms

1.3 Install instructions

To install GeoDMS, please follow the instructions on the ObjectVision website to install GeoDMS. Note that the installer does not automatically add GeoDMS to the Start menu, that will have to be done manually.

To install Transport Link Scanner, please unpack both zip files to a local drive. The simplest way is making one folder (e.g. D:\TLS) and unpacking the zipped folders directly into that folder so that, in the example, one obtains the following folders: [D:\TLS\data]; [D:\TLS\batch] and [D:\TLS\cfg].

Subsequently, start GeoDMS. If a dialog box asks to open a configuration file, point it to the main.dms file in the TLS cfg folder (e.g. D:\TLS\cfg\main.dms]. For now ignore any errors that may be noted. Open the Options menu (through CTRL+Alt + O or by mouse Tools -> Options). Go to the General Settings tab on the left. This menu is shown in Figure 1.



Help

Url: http://www.objectvision.nl/geodms

Helpfile: http://www.objectvision.nl/geodms

External programs

DMS editor: %ProgramFiles32%\Notepad++\Notepad++.exe "%F" -n%L

User modes

☑ Administrator mode ☑ Show State Colors in TreeView

□ TraceLogFile ☑ Suspend View Updates to favour GUI

□ Parallel Processing 1(tile based) □ PP2 (multiple calculation steps simultaneously)

Paths

LocalDataDir: C:\LocalData

SourceDataDir: D:\GeoDMS\ProjDir

Figure 1: GeoDMS general settings.

The following NEEDS to be adapted for the current installation:

- Change the path noted in the SourceDataDir box to the root path where Transport Link Scanner is placed (indicated by 1 in the figure). So if the TLS data path is D:\TLS\data, set the SourceDataDir as [D:\]. If this is done incorrectly the model will not run. In the example above, the TLS data is placed in D:\GeoDMS\ProjDir\TransportLinkScanner\data.
- Check the box in front of Administrator mode to ensure that all relevant items are visible in the model (indicated by 2 in the figure).

It is advised to change the following:

- GeoDMS stores intermediate calculation results in a so-called ‘CalcCache’ folder. This folder is stored in the so-called ‘LocalData’ path. It is advisable to point the LocalDataDir path (indicated by 3) to a physically separate hard drive (from the drive that stores the data and scripts). There has to be enough space on that drive (TLS can use 50Gb hard drive space in the LocalDataDir, or even more, while running).
- If Notepad++ is used as dms editor, change the “DMS editor” box (indicated by 4) with this text: [%ProgramFiles32%\Notepad++\Notepad++.exe "%F" -n%L] (without the brackets). This helps

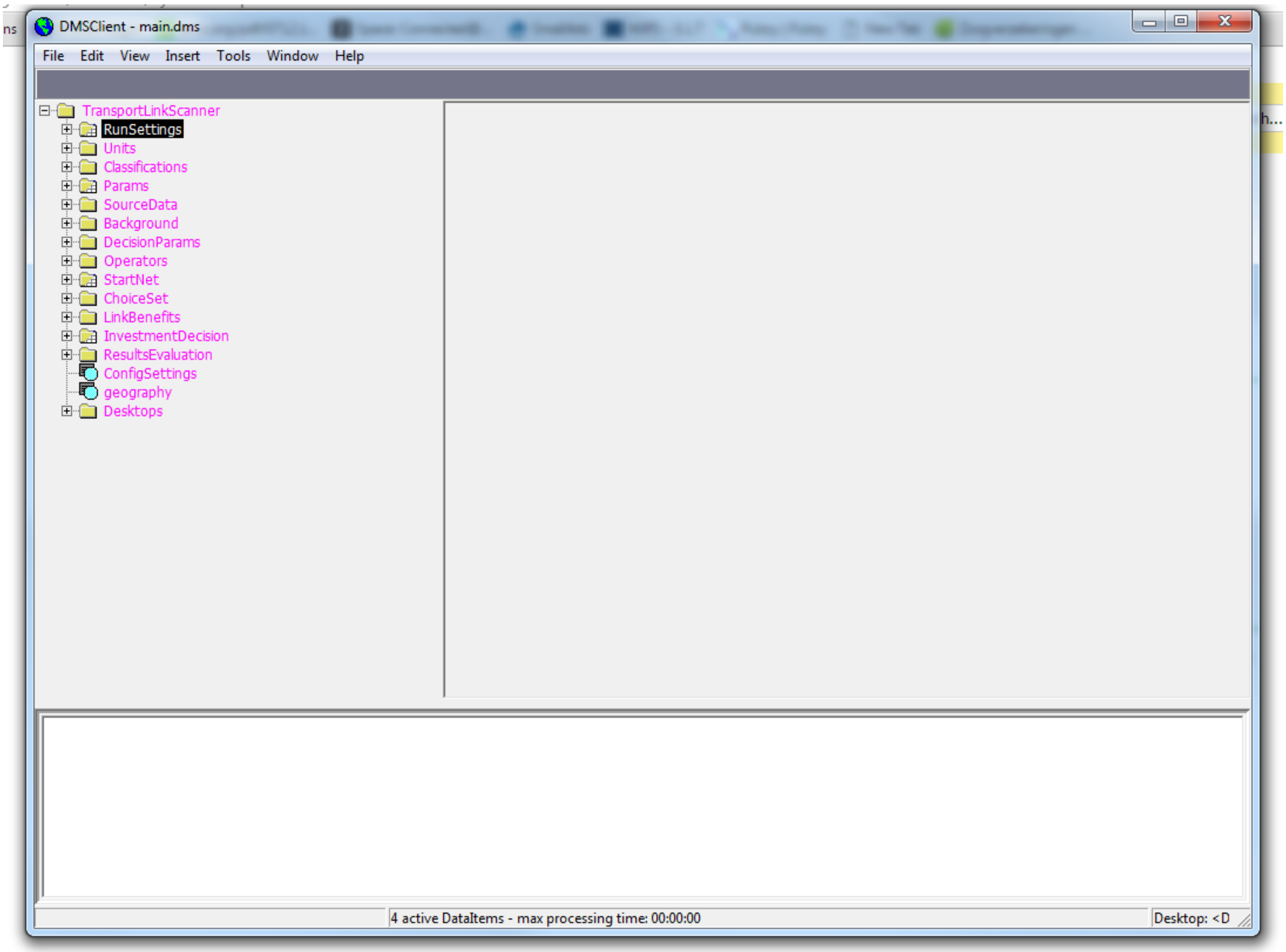
GeoDMS open the script files automatically when the user double clicks an error message or a ‘go-to script’ command, and point the user to the right line in the relevant script file.

- Check the boxes to ensure Parallel processing (Parallel Processing 1 and PP2). Those are located slightly beneath the administrator mode checkbox (2).

To ensure that the settings are stored and active, GeoDMS will have to be restarted.

After closing - but before restarting - GeoDMS, it is advisable to compress the contents of the LocalDataDir. This can be done through Windows Explorer. Compressing this folder will help obtain intermediate results (hard disk speed usually is a bigger limitation here than processer power).

After these steps all the main software settings should be in order. Make sure that the main.dms configuration file is opened in GeoDMS, if need be through File -> Open Configuration File, then point to the main.dms file (e.g. D:\TLS\cfg\main.dms). The screen should look more or less as in Figure 2.

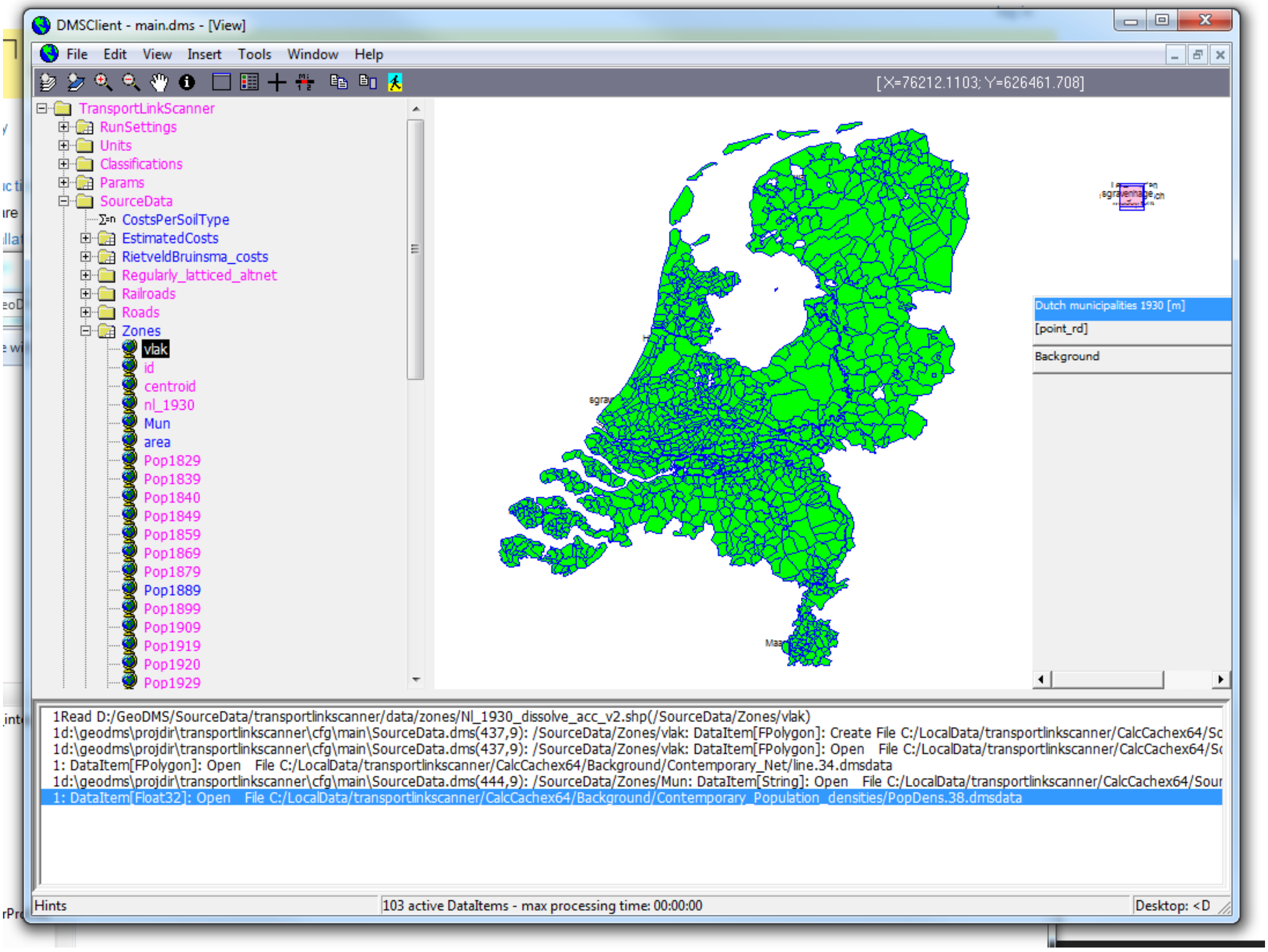


4 active DataItems - max processing time: 00:00:00

Figure 2: TLS opening screen.

Before anything else a scenario-specific investment tracking file will have to be generated. To do so, go to RunSettings and doubleclick the StoreNewInvestmentTracker item. This generates a file that will be stored in a new folder in the TLS install path, typically [D:\TLS\runs\SCENARIONAME\runtrack.dbf], and is necessary to let the otherwise iterative model keep track of the current investment number and current year. Remember that this item will have to be generated every time a new scenario is started. After generation this item will not have to be started again; if the file is regenerated for an existing scenario this will have the effect of a reset button, starting the scenario runs from new. Also note that this does not set the model to run; a more elaborate procedure is necessary to do so. Instructions for model running will be given later. After generation of the investment tracker, restart GeoDMS.

To ensure that the installation is executed correctly, go to SourceData -> Zones, open it, and double click the 'vlak' element in the unfolded list. If the settings are correct a map of historical Dutch municipalities should be visible as in Figure 3.



```
1Read D:/GeoDMS/SourceData/transportlinkscanner/data/zones/Nl_1930_dissolve_acc_v2.shp(/SourceData/Zones/vlak)
1d:\geodms\projdir\transportlinkscanner\cfg\main\SourceData.dms(437,9): /SourceData/Zones/vlak: DataItem[FPolygon]: Create File C:/LocalData/transportlinkscanner/CalcCachex64/Sc
1d:\geodms\projdir\transportlinkscanner\cfg\main\SourceData.dms(437,9): /SourceData/Zones/vlak: DataItem[FPolygon]: Open   File C:/LocalData/transportlinkscanner/CalcCachex64/Sc
1: DataItem[FPolygon]: Open   File C:/LocalData/transportlinkscanner/CalcCachex64/Background/Contemporary_Net/line.34.dmsdata
1d:\geodms\projdir\transportlinkscanner\cfg\main\SourceData.dms(444,9): /SourceData/Zones/Mun: DataItem[String]: Open   File C:/LocalData/transportlinkscanner/CalcCachex64/Sour
1: DataItem[Float32]: Open   File C:/LocalData/transportlinkscanner/CalcCachex64/Background/Contemporary_Population_densities/PopDens.38.dmsdata
```

Hints | 103 active DataItems - max processing time: 00:00:00 | Desktop: <D

Figure 3: map to be shown if TLS is set up correctly.

If GeoDMS shows the map as shown above, installation can be considered successful.

If the model issues an error message like the following, GeoDMS will most likely have to be restarted to ensure old hooks on datafiles are released for reading:

```
d:\geodms\projdir\transportlinkscanner\cfg\main\Background.dms(31,9):
/Background/Contemporary_Net/built: DataItem[UInt32]

Generating operation failed to complete, last ErrorMsg: ReadSequence Error: stream size 1112
conflicts with internal size 4294967294

: DataItem[UInt16] Failed.
```

```
d:\geodms\projdir\transportlinkscanner\cfg\main\Background.dms(31,9):
/Background/Contemporary_Net/built: DataItem[UInt32] Failed.
```

In the case of the above error message, restart GeoDMS and try again.

If the settings are incorrect an error message like the following will most likely be shown:

```
d:\geodms\projdir\transportlinkscanner\cfg\main\SourceData.dms(446,9):
/SourceData/Zones/Pop1829: DataItem[UInt32]

Cannot read data from
D:/GeoDMS/ProjDir/transportlinkscanner/data/zones/Nl_1930_dissolve_acc_v2.dbf because

d:\geodms\projdir\transportlinkscanner\cfg\main\SourceData.dms(448,9):
/SourceData/Zones/Pop1840: DataItem[UInt32]

Storage D:/GeoDMS/ProjDir/transportlinkscanner/data/zones/Nl_1930_dissolve_acc_v2.dbf does
not exist
```

In that case, control if the path mentioned in the 4th line exists; if not revise the previously outlined settings.

2 Structure of TLS

The model is technically structured in a number of interlinked modules. Their main purpose is explained here:

RunSettings: manages all the necessary settings for the model run at hand. Includes selected scenario, investment at hand, year of the current investment and year of available population data. Furthermore manages many Scenario-specific options in RunSettings/Scenarios.

Units: manages base settings of model such as geographical representation.

Classifications: stores visualization schemes for data.

Params: gathers model parameters for the scenario at hand.

SourceData: loads and prepares all data necessary for the model, including built railroads, zones, roads and regularly latticed network that is the basis for alternative paths.

Background: manages the background visualization that is mapped underneath any other mappable data.

DecisionParams: manages the preferences of investor types in the last parts of the model.

Operators: manages the modelled investors/operators and their characteristics.

StartNet: manages the modelled railway lines at the start of the model run.

ChoiceSet: modules in which the choice set of a model iteration is generated. The element tasked with the actual collection is [ChoiceSet/SelZones/Alternatives/ResultsCollection/allLinks]. All other code leads up to the generation of that item. The modules are tasked with the following:

NetworksPreparation: generates the current network

SelZones: selection of most attractive zone pairs and generation of plausible paths [to be renamed]

AltLinks: Definition of functions to generate plausible paths [to be renamed]

LinkBenefits: modules tasked with the computation of various benefits of alternatives in the choice set.

getInbetweens: tasked with loading separate alternatives in a generic manner

calcBenefits: tasked with computing changes in passenger mileage and accessibility levels

Schemes: definition of functions to compute changes in accessibility

AdditionFlowSchemes: definition of functions to compute transport flows

InvestmentDecision: gathers alternatives' benefits and picks one most attractive link based on the preferences defined in DecisionParams.

ResultsEvaluation: provides methods to compare the accuracy of various run scenarios.

3 Running a scenario

Before running a scenario, the proper scenario has to be selected by adapting the TLS code. To do so open the main.dms file and search for the 'ScenarioId' parameter or start editing it from the GeoDMS GUI. The latter is done by selecting RunSettings/ScenarioId and then either start the editor by right clicking and selecting 'edit config source' or press CTRL+E.

In the ScenarioId parameter, the scenario is selected by means of the name of the scenario. This can be changed by changing the typed name. Thus, if scenario A3 is selected, the attribute will look like this:

```
parameter<Scenarios>   ScenarioId:                    expr = "rlookup('A3', Scenarios/Name)";
```

If you want to run B5 instead, change the parameter into:

```
parameter<Scenarios>   ScenarioId:                    expr = "rlookup('B5', Scenarios/Name)";
```

In all cases the given scenario name will have to be identical to any of the scenario names in RunSettings/Scenarios/Name.

After adjusting the parameter, restart the GeoDMS GUI, and do not forget to store a new investment tracker through RunSettings/ StoreNewInvestmentTracker.

3.1 Using batch files

The easiest way to run a scenario is by using the batch files that come along with the scripts. Those batch files start the command-line execution of GeoDMS scripts and in that way manage the iterative running of the various modelling steps. All batch files can be found in the batch folder in the model's root folder. Which batch files to use depends on the chosen scenario. Some scenarios require full computation of benefits (e.g. A3, B5 and others); while currently two scenarios (the Random and RietveldBruinsma scenarios) only require the generation of a plausible path and the explicit addition of that path to the modelled network. The batch files for running a scenario always exist of two files: one steering the settings and taking care of the model iterations; the other file running the model.

The Random and RietveldBruinsma scenarios require **repeat_simple_gen.cmd** (steering) and **run_simplemodel.cmd** (running)**. To execute one of these scenarios** adjust the ScenarioId parameter accordingly, store a new investment tracker and **run repeat_simple_gen.cmd**

All other currently defined scenarios require **repeat_analysis.cmd** (steering) and **allocate_links.cmd** (running) to fully run the model. **To execute one of those scenarios** adjust the ScenarioId parameter accordingly, store a new investment tracker and **run repeat_analysis.cmd**

In all cases the batch files will have to be adjusted to match local computer settings. Both can be opened with common text editors.

3.2 Adjusting the steering files

In **repeat_analysis.cmd** or **repeat_simple_gen.cmd**, the following lines will have to be adjusted:

```
set pf="Program Files"
```

The pf parameter will need to reflect the root folder where GeoDMS is installed; typically this is the Program Files folder. As it is the batch files assume GeoDMS is installed on the C:\ drive.

```
set config=D:\TLS\cfg
```

The config parameter will need to reflect the folder where GeoDMS can find the main.dms file; in our common example this is the folder mentioned above.

```
set geodmsversion=GeoDMS7114
```

The geodmsversion reflects which version of GeoDMS is being used. Please note that the script will look into [*pf*]\ObjectVision\[*geodmsversion*] to find the command-line version of GeoDMS.

3.3 Adjusting the running files

In **allocate_links.cmd** and **run_simplemodel.cmd** the following lines may have to be adjusted.

In lines such as the underneath, the drive name where GeoDMS is installed may need to be changed:

```
C:\%pf%\ObjectVision\%geodmsversion%\GeoDmsRun.exe %config%\main.dms
AltRailways/ChoiceSet/SelZones/Alternatives/ResultsCollection/allLinks
```

To reduce unnecessary hard drive usage, the CalcCache is repeatedly deleted. The line below might therefore have to be changed to reflect the relevant LocalDataDir:

```
rmdir /s /q c:\LocalData\TransportLinkScanner\CalcCachex64
```

3.4 Computing process

After adjusting all the necessary files, run the batch files relevant for the scenario at hand. Depending on system specifications and the number of iterations set in the steering file, a model run may take 1 – 15 full days.

The results of the scenario are stored in [*model folder*]\runs\[*scenario name*].

In the **altgeneration** folder the choice sets are stored.

In the **decade_net** folder the network outcomes are stored. The outputs here consist of so-called ESRI shapefiles. The shapefiles are all called modellednet_[*Investment number*], with the shapefile with the highest investment number being the last model outcome.

In the **flows** folder the transport modelling results are stored as shapefiles per alternative.

In the **linkselection** folder relevant variables pertaining to the final investment decisions are stored.

4 Evaluating outputs

To evaluate the accuracy of ran scenarios a number of options to generate the necessary accuracy estimates are available. All options to evaluate model accuracy are in the ResultsEvaluation module. It is important to note here that for all evaluated scenarios, the results will have to be computed beforehand (see 'Running a scenario'). The method assumes that the modelled results are where the model stores them, i.e. in [*model folder*]\runs\[*scenario name*]. In all cases scenario accuracy evaluation is based on the historically built network.

4.1 Setting up the model results evaluation

Before evaluation, some particular inputs will have to be adjusted before evaluation. All necessary adjustments pertain to the same data element, namely ResultsEvaluation/ScenariosRan. First of all the number of ran scenarios that have to be evaluated will have to be adjusted. In the example below, three scenarios are included. To set the proper number of scenarios considered, change the NrOfRows propery of the ScenariosRan unit:

```
unit<uint32> ScenariosRan: NrOfRows = 3
```

or, in the case of for example 8 scenarios, set:

```
unit<uint32> ScenariosRan: NrOfRows = 8
```

Subsequently, two attributes will have to be adjusted to ensure that the scenarios of interest are evaluated properly. In the Name attribute the name of relevant scenarios will have to added or edited. In the example below, the B5, A3 and RietveldBruinsma scenarios are set to be evaluated:

```
attribute<string> Name:  ['B5', 'A3', 'RietveldBruinsma'];
```

Furthermore, the number of allocated investments for each of the scenarios will have to be added. All values are comma separated, and scenario names (indeed all string values) need to be between single quotes. The method needs this attribute for loading the required data. This is done in the LastInvestment attribute. In the example below, the model is set to read model results for the B5, A3 and RietveldBruinsma scenarios until the 56th, 63rd and 41st modelled net respectively.

```
attribute<uint32> LastInvestment:   [56, 63, 41];
```

If the numbers set in this attribute exceed the number of modelled investments in a scenario the method will not work. This means that if in the example a modellednet_57 does not exist in the folder of A3 results, this method will not work.

4.2 Generating model accuracy results

Accuracy measures are computed by calculating, for each investment in a scenario, those measures compared with the historically built network from when it had a roughly equivalent length. The method provides two ways to generate comprehensive model accuracy results, either for the entire network or by investor type.

To compile accuracy for the entire network, use the **CompileEvalResults** item. In the GeoDMS GUI, double-click that item or order to generate the table (CTRL+D). The method will subsequently start generating the accuracy results. Depending on system specifications and number of scenarios, this may take a considerable amount of time. When the computations are finished, right click the element for exporting the primary data.

To compile accuracy separated by investor type, do repeat the steps outlined above but select the **CompileEvalResultsPerOperatorType** data item.