Omnitrans International

How to convert from SATURN (10.3) to OmniTRANS (5.1)

Document and Contact Information

Document

Date:	01/03/2011
OmniTRANS version:	5.1 or later
Author(s):	Martijn Ruijgers

How to Contact Omnitrans International

🖃	Snipperlingsdijk 4 7417 BJ Deventer The Netherlands	Address
☎	+31-(0)570-666111 +31-(0)570-666888	Phone Fax
🖂	P.O. Box 161 7400 AD Deventer The Netherlands	Mail
@	info@omnitrans.nl	Email
@	techsupport@omnitrans.nl	Technical support
💻	www.omnitrans-international.com www.omnitrans.nl	Website

Contents

1 Introduction

This document is written to assist people who want to convert an existing SATURN project to OmniTRANS. The document assumes that the reader has a basic knowledge of OmniTRANS and a solid understanding of the SATURN data/model.

The process described in this document will result in an OmniTRANS project which can be used to analyse the quality of an existing traffic network, under varying levels of demand. Traffic can be assigned to the network both statically and dynamically. The project can be used as a base for further studies comprising trip generation, distribution, mode or departure time choice behaviour.

1.1 From SATURN to OmniTRANS

The conversion process consists of three steps:

I. Import the SATURN data (i.e. network, matrices) into OmniTRANS;
II. Convert and run the modelling process (assignment);
III. Interpret the resulting output data.

1.1.1 Importing the SATURN data

Using the OtSaturnImport and the OtMatrixCube classes, OmniTRANS can import both network and matrix data from SATURN to OmniTRANS. The starting point for the network import is the SATURN ASCII network file, and for the Trip Matrices matrices in ASCII format are required.

The procedures described in this document, and the functionality of both classes is based on the specifications of SATURN version 10.3.

OmniTRANS supports the import of:

- Geographical network layout (zones, nodes, links);
- Network attributes; e.g. link types, speeds, capacities and junction types;
- Results from an assignment in SATURN;
- Counts (link based only);
- Junction characteristics like signal settings and approach lanes;
- Trip Matrices.

Bus routes and frequencies can be modelled in OmniTRANS, however, the current OtSaturnImport class does not support the conversion of SATURN PT data to OmniTRANS, so this must be coded manually in the OmniTRANS project.

1.1.2 Converting the model structure

The OmniTRANS Job Language offers a wide variety of transport modelling classes and programming facilities to reproduce and/or extend the modelling process of SATURN. The OtTraffic class facilitates static traffic assignment modelling and offers multi-user-class, equilibrium assignment with junction modelling. The OtMadam class offers similar functionalities but then for macroscopic dynamic traffic assignment modelling. Both processes can be extended easily to incorporate demand modelling.

1.1.3 Interpreting the results

The OmniTRANS Graphical User Interface provides a wide range of user-friendly facilities to visualise and analyse the resulting output data. Examples are visualisation of flows combined with volume/capacity ratios, comparisons between variants, reports including visualisation of junction data.

The assignment modelling classes (OtTraffic and OtMadam) offer selected-link/route, selected-centroid, cordon, internal/external/through traffic analysis and many more. Turn data can be visualised for each junction separately.

1.2 The SATURN Template

In order to facilitate the conversion process, OmniTRANS offers a readymade 'SATURN template'. The template already incorporates project setup structures and various import and modelling jobs that are useful for conversion of the SATURN model to OmniTRANS. Most examples and assumptions in this document are based on this template.

1.3 Document Structure

The structure of this document is based on the three steps described in paragraph 1.1, hence, it covers the entire process of importing and validating of a SATURN project into OmniTRANS, running assignments in the model and analysing the results.
Chapter 2 describes import of data.
Chapter 3 describes the translation of the modelling process.
Chapter 4 describes the possibilities to visualise and analyse the output data.
Chapter 5 describes some theoretical/principal differences between OmniTRANS and SATURN.

2 The Import Process

2.1 Introduction

This chapter gives detailed information on the process of converting SATURN data (network and matrices) into OmniTRANS. There are differences between the two packages in the way data is used to describe the networks; these are reviewed in this section.

To import the SATURN data into OmniTRANS the following structure is recommended:

1. Prepare the SATURN data file;
2. Start a new project in OmniTRANS, using the SATURN Template;
3. Adjust the project settings of this template to the specific needs of the new project;
4. Import the Saturn network using the OtSaturnImport class;
5. Check the results of the import (warnings and visual);
6. Import the Trip Matrices;

2.2 Preparing the SATURN data

2.2.1 The ASCII network file

The starting point for the import is an ASCII file containing a single SATURN network for a given time period. The OtSaturnImport class is then used to import this data into an OmniTRANS project.

Warnings and Errors

OtSaturnImport generates warnings if the data is not specified as expected or if the data is modelled differently in OmniTRANS. However, it does not check whether the data is logical from a modelling point of view.

The OtSaturnImport may generate warnings and or error messages. If these relate to conditions in the data that can be corrected in the input ASCII file (and so may have either generated warnings in a Saturn run, or relate to data conditions that SATURN is more tolerant of) then this should be done and the import process re-run. If this cannot be done then any corrections will have to made within OmniTRANS to make the data compliant with the OmniTRANS data structure.

An example of a such a warning is where green times and inter greens of a signalised junction are greater than the cycle time. SATURN scales all green times ‘to fit’, whereas OmniTRANS does not. As a result, data may get lost. In Appendix A a list is given of the most common warnings produced during the import by OtSaturnImport.

The $INCLUDE command

SATURN allows the inclusion of associated data files within the main ASCII network files using the $INCLUDE command:

```
$INCLUDE SATURN.111
```

The OtSaturnImport class cannot process these $INCLUDE files, so all ASCII network files must be combined into one.

Data Sections not processed

OtSaturnImport does not support the import of:

- the fixed routes section (66666)
- the generalised costs section (88888)

Consequently, it is not necessary to include them.

2.2.2 Trip Matrices

SATURN's trip matrices are stored in a .ufm format (possibly as stacked matrices), which cannot be read directly by OmniTRANS. It is necessary to convert these within SATURN to unstacked matrices in ASCII format.

The procedure to do this is as follows:.

Unstack a matrix

Module Run → SATURN → MX
Matrix ➔ specify the .ufm file which contains the trip matrix
Choose:
  15 – stacking and unstacking .ufm files
  1 – unstack the internal matrix into x output square .ufm matrices
specify the filename of the first trip matrix e.g. 89LGV.txt
specify the filename of the next trip matrix e.g. 89HGV.txt
Choose:
  0 – return
  0 – exit
Now the stacked matrix is split into square matrices. However, they are still in the .ufm format and have to be converted into ASCII format.

Convert a matrix from .ufm to ASCII format

Module Run → SATURN → MX
Matrix ➔ specify the .ufm file which contains the OD-data → run
Choose:
  13 – dump matrix data to a .dat file
  5 – comma separated spreadsheet-style format (all cells included)
specify the filename of the trip matrix e.g. 89LGV.txt
Choose:
  0 – do it!

0 – return
0 – exit

Now a stacked matrix in .txt format is created.

```
RUN DUMPED MATRIX FROM MX
 &PARAM
 MPNEXT  =    T
 LONG    =    T
 NROWS   =  137
 NCOLS   =  137
 &END
                 0
TITLE UNSET                                            LEVEL -  1
53852    0.000   0.009   0.005   0.003   0.002   0.009   0.009
         0.000   0.004   0.001   0.000   0.009   0.001   0.000
         0.000   0.000   0.000   0.087   0.001   0.000   0.009
         0.000   0.009   0.009   0.000   0.000   0.001   0.000
```

Repeat these steps until all matrices are converted. The import of the matrices into OmniTRANS is performed by the OtMatrixCube class. This is explained in paragraph 2.9.

2.2.3 Results of a SATURN assignment

For control purposes it can be useful to analyse SATURN assignment results using OmniTRANS; primarily to compare SATURN and OmniTRANS results, but it could be to compare various SATURN assignment results.

To do this an ASCII file must be created in SATURN in which the flows per link are stored. The procedure to do this is as follows:

Module Run → SATURN → SATDB
SATURN UF file ➔ specify the .ufs file which was created by SATALL

The choices for exporting loads (to be imported into OmniTRANS) are following:
2 – enter the link selection procedure
6 – based on link type
2 – simulation turns; no
3 – simulation centroid connectors; no
5 – buffer centroid connectors; no
0 – no more changes
0 – no more tests
4 – read link based data from the .uf file(s)
1493 – flow up, demand flow upstream
4503 – d-flows, demand or assigned flows
1423 – down flow, total demand downstream flow
0 – return
13 – dump the full data base to an ASCII file
2 – node formats based on 5 or 10 columns, compatible with SATURN inputs
10 – output list sorted as per screen listings

0 – continue and accept the following:
specify the filename of the results e.g. peakva.txt
run
0 – terminate

The result is a file containing five columns:

- Anode
- Bnode
- upstr. act. flow
- mid. act. flow
- downstr. act. flow

Because OmniTRANS requires that all zone centroids are connected to a link using a zone centroid connector, SATURN links may be split into two new to accommodate the node for the generated connector link links (see paragraph 2.6.3). The upstream and downstream flows are used to calculate the flows on the split links, the middle flow is put directly on the original links. Furthermore, due to the modelling difference it is not possible to show results on connector links.

2.3 Starting a new OmniTRANS project

2.3.1 SATURN template

To enable a quick start of a new project, OmniTRANS contains several different templates to facilitate the set up of new projects. These templates contain different combinations of database fields, jobs and predefined graphics, etc.

For the import of SATURN projects a special template is provided containing the basic requirements which can (and probably should) be adjusted or extended to fulfil the specific needs of your project. It is recommended that you use this template as your starting point.

This template is be found in the 'Quick Imports' tab when starting a new project (Ctrl+n, or the '*create a new project based on a template*' reference in the Start Up window).

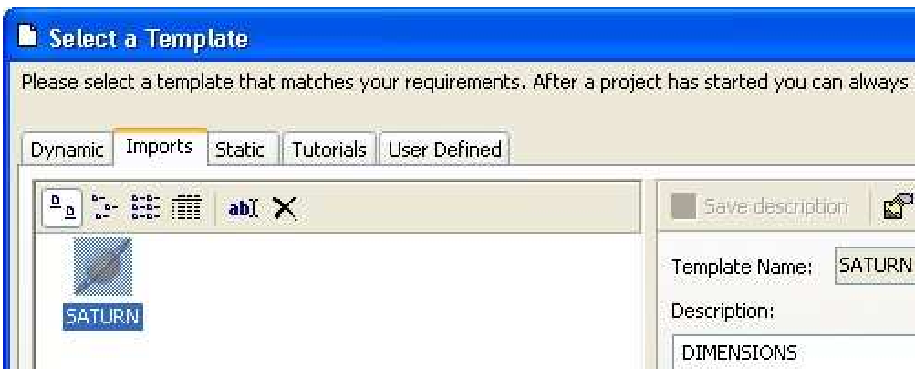


Figure 1- Template window

Most of the examples in this document are based on the parameters and dimensions of this template.

2.4 Customise the project settings

To adjust the new project to your needs the original project requirements as specified in the SATURN data file should be specified in OmniTRANS. Most of these should be set in the Project Setup window, which contains seven tabs. Four of these tabs are relevant for the import of SATURN projects. The tabs: *Zonal Data*, *Transit Modes* and *Transit Transfers* are not considered here.

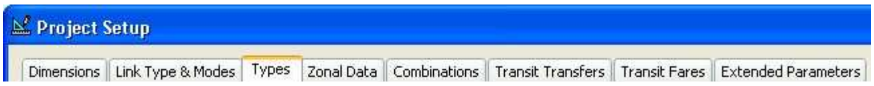


Figure 2- Project Setup tabs

The four relevant tabs will be discussed below.

2.4.1 Dimensions

One of the characteristic OmniTRANS features is the usage of a six-dimensional structure for data management. This PMTURI-structure contains the following dimensions: Purpose, Mode, Time, User, Result, Iteration.

The number of required dimensions varies per object type; for example, for a network only the Mode and Time dimensions are relevant and for a trip matrix only the Purpose, Mode, Time and User dimensions. A thorough review of the usage of the PMTURI-structure is given in the Online Manual.

All dimensions have at least one entry in the SATURN template; to add or edit entries use the yellow buttons (see Figure 3) or the right mouse button.

Figure 3 – Dimensions tab

Below a brief review of the six dimensions is given.

Purpose

The dimension purpose is used to distinguish different trip and these should be defined to reflect the journey purpose structure (if any) in the imported SATURN model. Only one purpose, ‘total’, is defined in the template, but this can be extended easily to accommodate your model structure. See Figure 4 for an example.

Figure 4 – Purpose dimension

Mode

Together with Time, Mode is one of the two dimensions which are relevant for the network specification. This implies that links can be specified which are only accessible for certain modes, for instance bus lanes.

A SATURN network is primarily associated with private vehicles and consequently the 'default' mode defined in the Saturn template is 'car'. However, it is possible that public transport and walk will be a component, so the Mode dimension should be defined to reflect the modal structure of the imported SATURN model. This might include a network distinction between cars and lorries so that different free flow speeds and capacities can be specified.

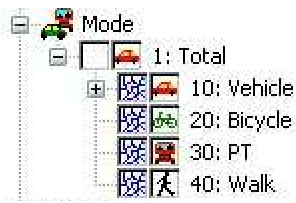


Figure 5 – Mode dimension

Time

The time dimension is also associated with the network. This is particular useful for modelling time dependent traffic measures such as reduced maximum speed during the peak periods or usage of peak lanes. Further, the time dimension can be used for data management purposes, i.e. to store different trip matrices and the resulting loads per time period.

Figure 6 – Time dimension

User

The user dimension can be used to specify user based segmentation in the data. This can include segmentation such as trips originating 'locally' or 'regionally' (different network access; 44444 data field in SATURN) and business drivers or non-business drivers (different values of time and distance; 88888 data field in

SATURN). The user dimension is very important in the multi user class assignment (see paragraph 3.5).

Figure 7 – User dimension

Results

The previous dimensions were all required to define the location of the network and trip matrices. The results (and iteration) dimension is used to store the results of different assignment methodologies, such as All or Nothing, Equilibrium, or loads generated by SATURN. Below the default result dimensions of the SATURN template are shown.

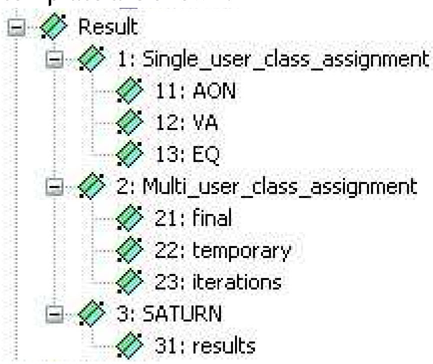


Figure 8 – Result dimension

Iterations

In here, the results per iteration of an iterative assignment can be stored. When the model also incorporates a demand model it can be used to store different cycles of the feedback process.

Figure 9 – Iteration dimension

2.4.2 Review of the dimensions

Given these facts, the user should base the OmniTRANS project dimensions on the following considerations:

- If there are trip matrices for different time periods it is recommended to distinguish these by the time dimension. If capacities or speed limits differ between these time periods, this can be specified in the Link Types & Mode tab (see paragraph 2.4.4).

- If there are multiple user groups with identical values of time and network access, they can either be stored in the purpose dimension or in the user dimension.
- If there are user groups with different values of time and network access, these should be stored in the user dimension and assigned to the network using a multi user class assignment.
- Although the transit data is not imported automatically into OmniTRANS, it can be specified manually. To do this, the mode dimension should contain car, transit and walk modes.
- Results of different assignments should be stored in the result dimension. Consider which assignments should be stored. For example, it might be useful to store the results of a volume averaging assignment with junction modelling and one without junction modelling.

2.4.3 Types

A type is associated with an object and contains two or more type items. For example, the type 'County' could be associated with the object Centroids and contain the type items Surrey, Kent and Sussex, etc. Types are managed in the *Project Setup* window and can be useful to distinguish or aggregate characteristics associated with certain objects in reports and visualisations on screen.

All projects contain the type: LinkType. Since this mandatory type has many features per type item, it has its own tab (called Link Type & Modes), which will be discussed in paragraph 2.2.3. Next to LinkType, three types are defined in the SATURN template as can be seen in Figure 10; NetworkType, NodeType and TurnType.

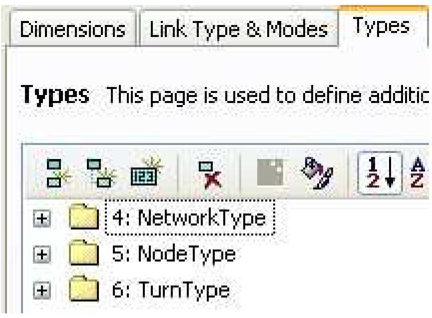


Figure 10 – Types tab

Types can be visualised in the network using the setting-buttons (e.g. node settings) on the graphics toolbar.

Figure 11 – Node settings

NetworkType

NetworkType contains two type items: Simulation and Buffer. These relate to the SATURN link definitions and are preserved during the import in this type. This type can be used to visualise the buffer and simulation network or to detect inconsistencies in the project (e.g. nodes which are modelled as junctions but are connected to buffer links).

NodeType

NodeType provides information about the nature of the nodes in the network. Some of them correspond to the junction types in SATURN, some of them correspond to the junction modelling of OmniTRANS, and some have been added to draw attention to specific situations due to the conversion.

It is important to realise that node type is not associated with the junction modelling. Hence, changing the node type does not influence the junction modelling and vice versa. To keep the visualisation of your network correct, it is recommended to change the node type as well, if the junction type is changed.

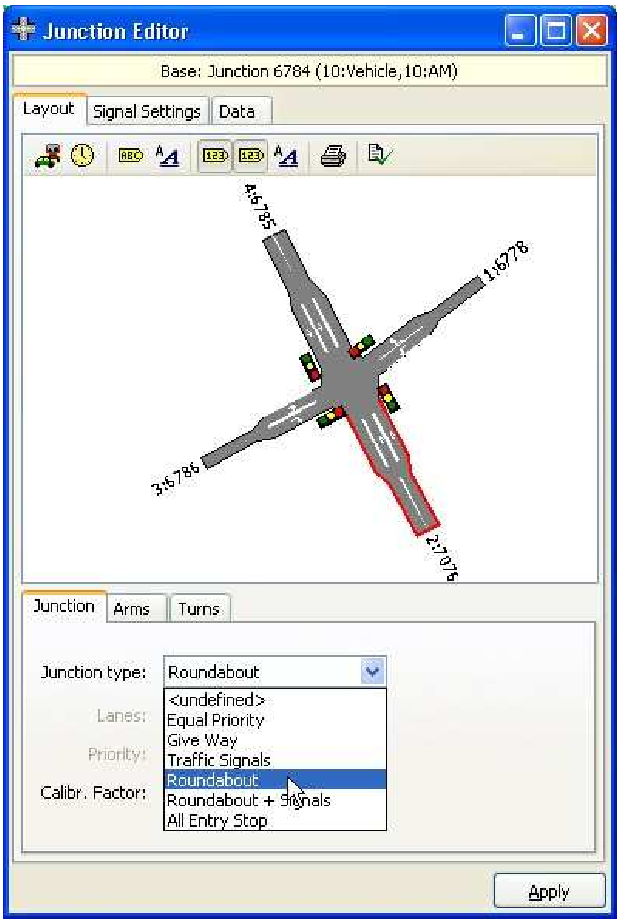


Figure 12 – Changing the junction in the junction editor

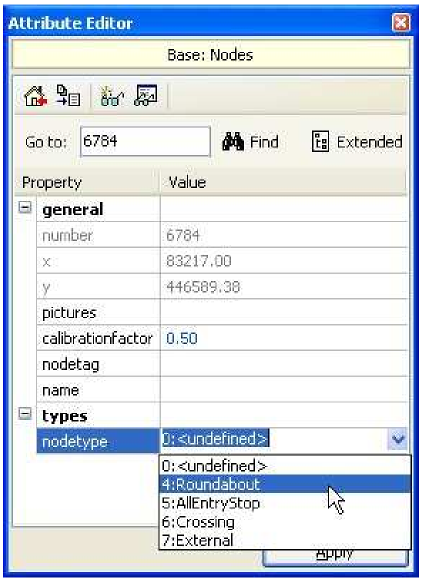


Figure 13 – Changing the nodeType in the node window

In the SATURN template 11 predefined type items are specified for the NodeType.

Figure 14 – Type items of nodeType

Below the main characteristics of every type item are summarised.

<undefined>
This type item is generated during the import if the junction data was not valid or if it was removed because the specification in the ASCII network file contains:

- an undefined junction type;
- merging or weaving turns;
- simulation links AND buffer links;
- junctions with less than two links (except for crossings).

For more information about these issues see paragraph 2.6.4.

EqualPriority
Not possible in SATURN. Hence, not imported, but can be created/changed manually.

GiveWay
Applied if SATURN junction type is 1, priority junction.

TrafficSignals
Applied if SATURN junction type is 3, traffic signals.

Roundabout
Applied if SATURN junction type is 2, roundabout or 5, roundabout with U-turn.

Roundabout+Signals
Not supported in SATURN.

AllEntryStop
Not supported in SATURN.

Crossing
In SATURN signalised junctions with two links are used to model crossings of pedestrians or railways by defining signal stages with large intergreens. In OmniTRANS it is not possible to apply junction modelling to junctions with only two links. The effect of a crossing should be modelled manually in OmniTRANS by specifying turn delays for the node (note that turn delays must be specified in hours).

External
Applied if SATURN junction type is 0, external node.

Dummy
Applied if SATURN junction type is 4, dummy node.

Inserted
SATURN connects centroids to links instead of nodes. This is not recommended in OmniTRANS if junction modelling is used, and therefore a new node will be inserted on a link. The centroid is connected to this node by means of a connector link. For more information about this procedure see paragraph 2.6.3.

TurnType

The turn priority markers from the SATURN data file are for visualisation purposes stored as TurnType.

Figure 15 – TurnType

TurnType is only used for visualisation and does not influence the junction modelling or the other way around.

2.4.4 Link Type & Modes

As mentioned in the previous paragraph, LinkType has its own tab (called Link Type & Modes) where the properties of all the link types can be specified. The predefined link types of the SATURN template are based on the link types stated in the SATURN manual (for more details about these link types see paragraph 3.2.2).

In OmniTRANS for each link type different values for speed and capacity can be specified per mode and per time. Hence, a network is mode and time dependent. This will be explained based on Figure 16 and Figure 17. In the example the dimensions of the SATURN template (left) are extended with the mode ‘lorries’, and a network is associated to the off peak period (right).

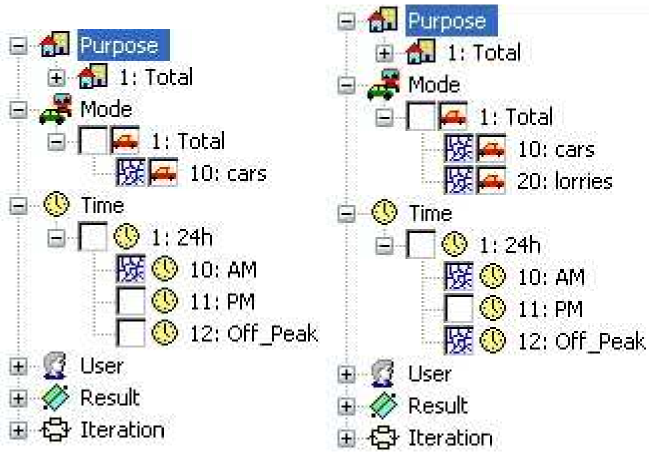


Figure 16 – SATURN template dimensions vs. extended dimensions

A project setup such as this could be used to model a highway network. The link characteristics are specified in the Link Types & Modes tab, which is shown in Figure 17.

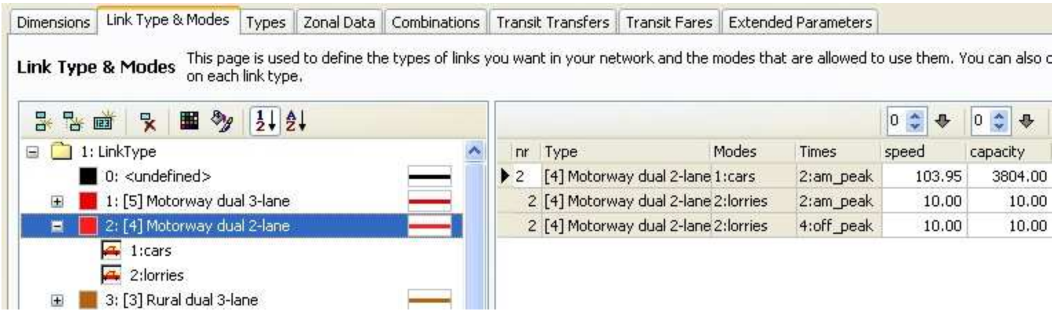


Figure 17 – Properties of link types

In this example lorries are only allowed to use one lane (capacity is 1902). Cars can use two lanes (capacity is 2 * 1902) during the off-peak period, and three lanes during the peak period (capacity is 3 * 1902). Frthermore, lorries have a maximum speed of 80 km/h and cars 103.95 km/h. Thus, an assignment in the peak period will lead to different results than an assignment to the off_peak period.

A consequence of using different capacities and speeds for both categories is that it is not possible to assign both simultaneously in the multi user class assignment as described in paragraph 3.5. However, usually lorries form only a small percentage of the total number of vehicles and therefore the lorries can be assigned first to their own network, and then be used as fixed flows in the assignment of the cars. If the proportion of lorries is large, it might be better to assign cars and lorries simultaneously using the multi user class assignment (see paragraph 3.5), using the User dimension. In this case it is not possible to model different capacities per user class.

In SATURN it is not possible to specify different speeds and capacities per mode or per time period. Therefore this is not directly relevant for the import. However, it may be used to refine your model after the import.

After loading the SATURN template, the project should be modified, to match the original SATURN project. Therefore it must be determined which capacity indices are specified in the ASCII network file. They can be found in the:

- 11111 data field, to specify individual simulation links;
- 33333 data field, to specify default values for capacity indices;
- 33333 data field, to specify individual buffer links.

All the different capacity indices found in the data fields above must be specified as a link type item in the 'Link Type & Modes' tab. Make sure the all capacity indices of your SATURN project are defined as linktype, otherwise they will be set as undefined, which means they will not be used in the path building and assignment processes.

Two additional link type items should be specified to enable the definition of:

- simulation connectors;
- undefined links in the ASCII network file.

2.4.5 Combinations (count data)

In SATURN count data is stored in the 77777 data field of the ASCII network file. It may contain counts on links and counts on turns. OmniTRANS does not support the storage of count combinations per turn, but only per link (per direction). The OtSaturnImport class automatically skips counts in the 77777 data field which are based on turns.

```
77777
*                              LGVs      HGVs     TOTAL
1604      1603                  766       316      1082
1603      1604      1605        623       199       822
99999
```

In the example above, the counts on the turn 1603 – 1604 – 1605 will be skipped during the import since it is not a link count but a turn count.
In OmniTRANS counts are stored as 'Combinations'. Combinations are used for the definition of productions and attractions (which are not considered in this document) and the definition of counts. In the SATURN template, three combinations are specified for the storage of counts; car, lorries and total.

2.4.6 Miscellaneous

Left hand drive

If the OtSaturnImport class is used, the settings of the data file are applied only to the way the data is imported; the project settings are not automatically adjusted. It is very important to specify the correct rule of road manually, otherwise the assignment will lead to unreliable results. The settings can be changed in the Preferences window.

Figure 18 – Preferences window

This window can be accessed via the Tools menu or the key combination *Ctrl + F*.

2.5 Importing the Network and Junction Data; OtSaturnImport

2.5.1 Basic Import Job

The SATURN template contains an example job for importing a SATURN network; this uses the OtSaturnImport class that has been referred to earlier. This job, described below, contains the most commonly used properties and provides an easy starting point for importing a SATURN network. You are referred to the Online Manual for a complete description of the OtSaturnImport class properties.

```
require 'OtConversion.rb'

imp = OtSaturnImport.new
  #specify the pathname of the ASCII data file
  imp.filename = "#{$Ot.dirPrj}External User Data\\network.dat"
  #specify the mode and time dimension of the network
  imp.network  = [1,2]

  imp.countCombinations = [1,2,3]

  #define the types as specified in the project setup
  imp.linkType = 1
  imp.networkType = 2
  imp.nodeType = 3
  imp.turnType = 4

  #define the type items as specified in the project setup
  imp.defaultLinkTypeItem   = 100
  imp.connectorLinkTypeItem = 101

  #what to do with differences between OmniTRANS and SATURN
  imp.removeJunctions = ONE_IN_TWO_OUT_JUNCTIONS |
      MERGE_AND_WEAVE_JUNCTIONS

imp.execute
```

The content of this job is described as follows:

The first line, *require OtConversion*, is necessary to load the Conversion class into OmniTRANS, of which OtSaturnImport is a member.

After creating an instance of the OtSaturnImport class called ‘imp’, the location of the SATURN data file is specified, as are the mode and time dimensions into which the network is imported.

Next, the types and type items are specified. These must correspond with the values defined in the Project Setup.

OmniTRANS does not model junctions with one inbound link and two outbound links, therefore junction information of these junctions is not imported. Furthermore, merging and weaving in OmniTRANS is not modelled as a junction, hence, this information is removed as well.

Finally the method *OtSaturnImport.execute* performs the actual import of the data, based on the specified parameters.

2.6 Differences between SATURN and OmniTRANS

2.6.1 Introduction

Before the properties of the OtSaturnImport class are discussed in detail it is important to understand the differences between the way SATURN and OmniTRANS handles its data structures. These are reviewed below.

2.6.2 Centroid numbering

SATURN allows zones to be numbered in any way the user desires; in OmniTRANS zones must be numbered starting at 1 and increasing monotonically until the maximum zone number is reached.

Although OmniTRANS can support a zone numbering system that has gaps in the numbering (which would accommodate a SATURN 'free' zone numbering system) the penalty in OmniTRANS would be 'dummy' zone centroids in the model which will increase computation time and the size of the database.

The default operation during import is for zone centroids to be automatically renumbered using the OmniTRANS 'rules', with the original zone number being preserved in the 'name' field (see Figure 19). However, the property OtImportSaturn.renumberCentroids (= false) can be used to preserve the original zone centroid.

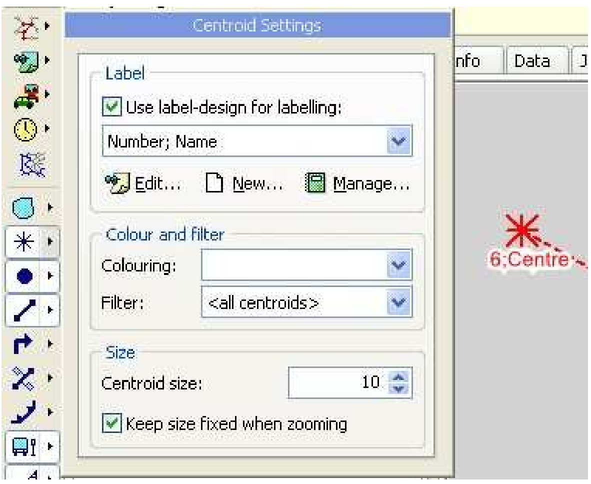


Figure 19 – Centroid number and name

In Figure 19, the original SATURN centroid number was 509, and this was renumbered during the import to 332.

2.6.3 Zone Centroid Connector links

An important difference between SATURN and OmniTRANS is the way vehicles enter the network from the centroids. SATURN allocates traffic to links, while OmniTRANS allocates traffic to nodes. In Figure 20 the traffic enters and leaves the network on link 2847 _ 2841 (in both directions). If this would be converted into OmniTRANS by allocating the traffic to node 2847 and 2841 wrong results would be obtained, since OmniTRANS uses a different junction definition.

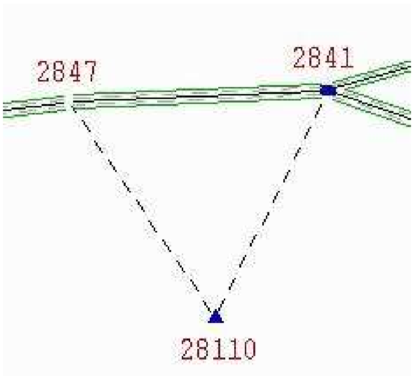


Figure 20 – Centroid connector in SATURN

To overcome this problem additional nodes are inserted during the import of the network. In Figure 21 the result of this process is shown.

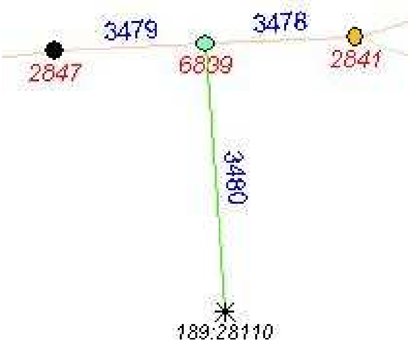


Figure 21 – Converted centroid connector in OmniTRANS

The two connectors to the nodes 2847 and 2841 are replaced by one single connector which is attached to the network by a new node 6839. The node type 'inserted node' is applied to these kind of nodes. In this way it can easily be seen in the network which nodes are new.

The result of this conversion is that links are divided into two new links. Consequently, after the import the original correspondence of zone and node numbers between SATURN and OmniTRANS is no longer valid.

Finally, it can be seen that SATURN uses a different way of specifying links than OmniTRANS. In SATURN a link and the direction are specified by a-node and b-node. In OmniTrans every link has a linknumber, and the direction is specified by 1 or 2.

The insertion of nodes is not performed if the property .addConnectorNode is set to false. This might be done for SATURN projects in which the centroids are connected to nodes instead of links, like in OmniTRANS (see below). Another reason might be that no junction modelling is applied. In that case no problems will occur from junction modelling, and the original node numbers can be used.

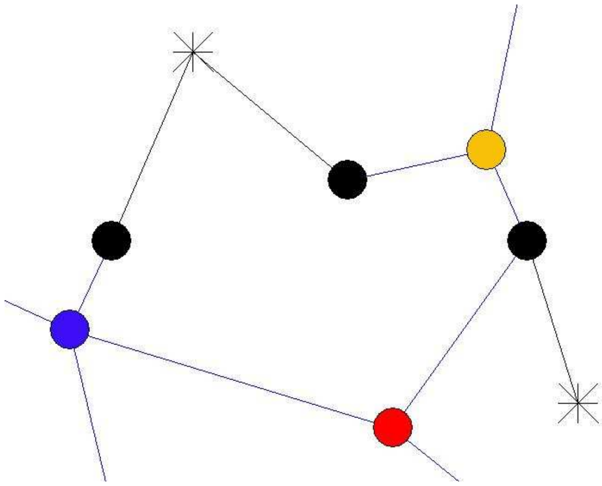


Figure 22 – Centroids attached to nodes instead of links

2.6.4 Junctions

The objective of junction modelling in OmniTRANS is to create a realistic model of a network whilst spending minimal effort on coding individual junctions. This results in the use of default junction types, which can individually be manipulated to reflect reality (e.g. signal settings). The result is a time efficient data input process, without losing reliability of the results of the assignments.

Sometimes it might be beneficial to change the default settings of the junction modelling (parameters such as the maximum cycle time and the average vehicle length). Changing the default settings is performed in an ASCII-file and is recommended for expert users only. For normal situations the default values are appropriate. For more information see the document 'Junction Modelling in OmniTRANS'.

Junctions can be created or edited in the Junction Editor (Ctrl+j). The two tabs Layout and Signal Settings are used for input, while the Loads tab can be used to visualise results of an assignment (see paragraph 4.2.3).

Below the differences in specification and modelling of some junction types will be discussed.

Junction with buffer links

It sometimes occurs that junctions are specified for nodes of buffer links (mostly due to the different approach of connecting centroids to the network). Since no turns are defined for buffer links, it is not possible to determine the configuration of this junction properly. To avoid unreliable results, all junction data for this junction is removed, and a warning is generated. In this case the user should specify the junction manually using the junction editor.

Signalised intersections

In Figure 23 it is shown how the code 'snippet' from the ASCII network file shown below is visualised in OmniTRANS.

```
1390    4    3    4   21  120    0    0    0
  45 1391*   2   35  233    0   0 0 3800   1 2 1200X 2 2
             37   15  740          1.83    1
  42 1372*   3   35  120 1300   1 1 1900   2 2 1600  3 3
             37   15 1480          1.83    1
  35 1389*   2   35  172 1300   1 1 3800   1 2    0  0 0
             37   15  740          1.83    1
   0 1406    0    0  140    0   0 0    0   0 0    0  0 0
             61    4    4 1389    0 1391    0
             12    2    2 1391    0
             22    0    4 1372 1406 1372 1391
             13    5    2 1372    0
```

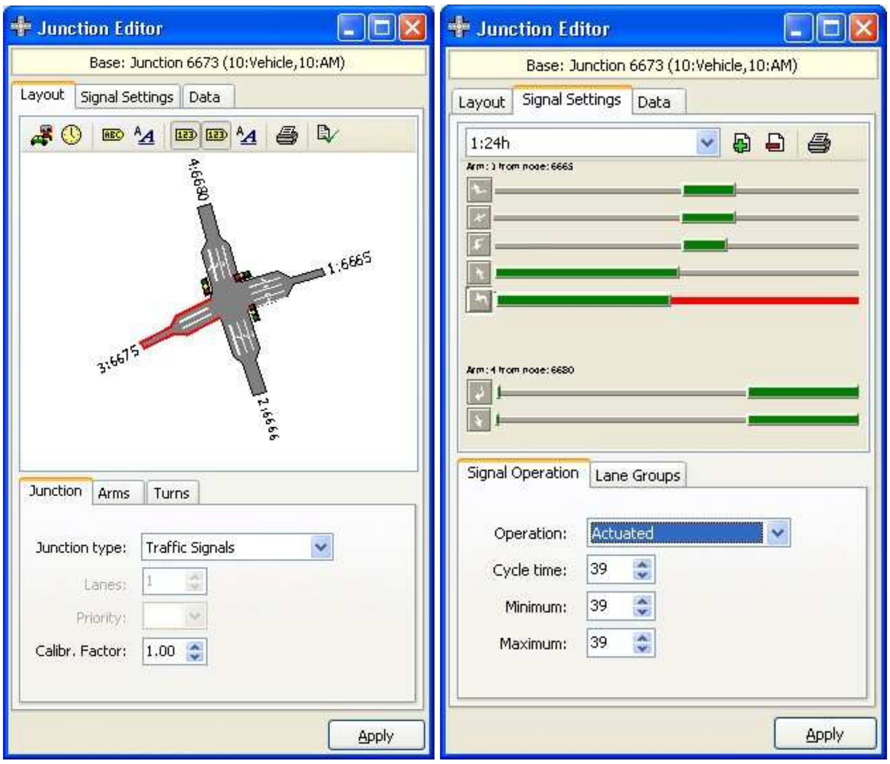


Figure 23 – Junction Editor; signalised intersection

As can be seen, the data which is specified in the ASCII network file is used to define the local characteristics of the junction.

Note that the inter green between stage one and two for turns from node 1391 to all nodes, is in SATURN modelled as green, while it is modelled as red in OmniTRANS. In most common occasions this will have not have a significant influence on the results. Furthermore, OmniTRANS currently does not support synchronisation of signals of multiple junctions. Therefore, the offset of the cycle which is specified in SATURN is ignored.

Roundabouts

In SATURN U-turns are generally banned, but they are automatically generated for junction type 5 roundabouts. In OmniTRANS U-turns are not possible. If it is desired to enable U-turns on a certain roundabout, this should be simulated by manually specifying a series of three way intersections. It is not important whether a roundabout is specified as type 2 or type 5, the results after importing them into OmniTRANS are the same.

Consider the piece of code from an ASCII network file below.

```
7506    4    2   15 2800             25
     2338    2   80  475 2400  1 2 2400  1 2 2400  1 2
  87 2301    2   48  500 1500  1 2 1500  1 2 1500  1 2
 211 2243    2   80  805 1750  1 2 1750  1 2 1750  1 2
     7507    2   40  100 2287  1 2 2287  1 2 2287  1 2
```

After importing this into OmniTRANS, this will result in the junction of Figure 24.

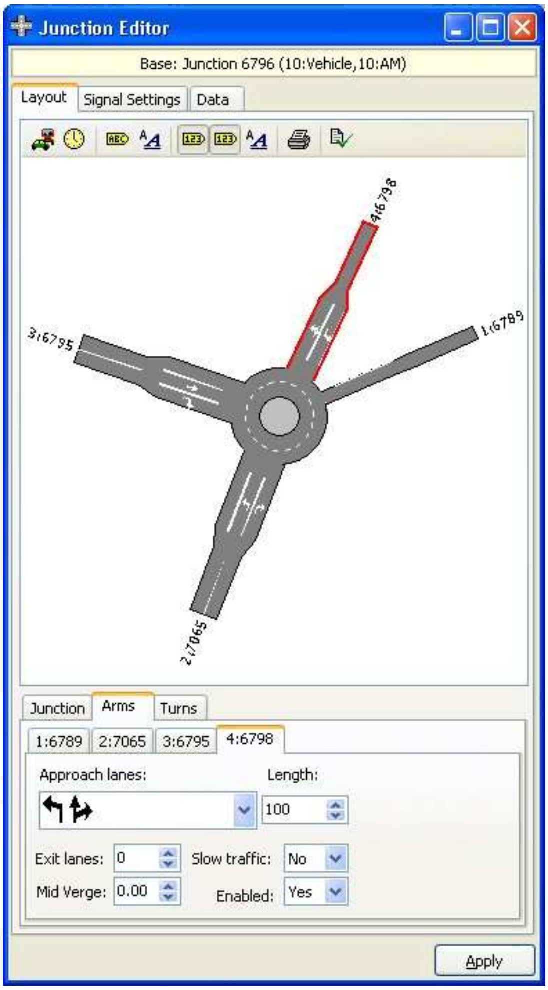


Figure 24 – Junction Editor; roundabout

Note that the arrows on the approach lanes are only used for visualisation. In fact, this roundabout is modelled as a two lane roundabout.

Merges and weaves

OmniTRANS does not take merging or weaving on motorways into account for static assignments. For dynamic assignments both are taken into account, however, not based on the static junction modelling of OtTraffic.
Although theoretically merging and weaving are priority situations, the priority junction of the junction modelling should not be used. The junction modelling is based on theories which apply to the surface street network, not to motorways, and therefore coding merges of freeway on-ramps as a priority junction will lead to incorrect results. Therefore, in most situations it is better to remove the junction modelling data from the nodes with merge or weave turns. Because OmniTRANS does not recognise whether a junction is on a motorway or on the surface street network, there are two possibilities, either remove the junction modelling data from all nodes containing merge or weave turns of from none at all. This can be specified using the property OtTraffic.removeJunctions.

```
imp.removeJunctions = MERGE_AND_WEAVE_JUNCTIONS
```

2.7 OtSaturnImport properties

2.7.1 Network geometry and SATURN results

To import a SATURN network the location of the ASCII network file must be specified using the OtSaturnImport.filename property.

```
imp.filename = "#{$Ot.dirPrj}External User Data\\network.dat"
```

By using #{$Ot.dirPrj}, the specified folders and files are searched in the current project directory. In this way it is not necessary to specify the complete path of the project folder, which can be quite useful if the project is copied to a different location.

It is necessary to specify into which mode and time dimensions the network should be imported. This is done with the OtSaturnImport.network property, which requires an array of two fields, mode and time.

```
imp.network  = [1,2]
```

If results of a SATURN assignment are to be imported, the location of this file must be specified using OtSaturnImport.loadsFilename.

```
imp.loadsFilename = "#{$Ot.dirPrj}External User Data\\AON.txt"
```

The .loadsFilename property is optional and used only if results of a SATURN assignment should be imported. If results of more assignments should be imported this can be done by using the .loadsOnly property. To do this run the import job again, however setting .loadsOnly = true and specifying a different path name for the SATURN results and a different PMTURI to store them. Below an example is given how a network and two assignment results can be imported into OmniTRANS.

Run OtSaturnImport once to import the network and the results of one assignment.

```
require 'OtConversion.rb'
# Import the network including AON-results from SATURN
  imp = OtSaturnImport.new
    imp.network  = [1,2]
    imp.filename = "#{$Ot.dirPrj}External User Data\\Network.dat"
    imp.loadsFilename = "#{$Ot.dirPrj}External User Data\\VA.txt"
    imp.loads = [1,1,2,1,41,1]
    imp.loadsOnly = false
  imp.execute
```

Run OtSaturnImport for the second time to import the results of a second SATURN assignment.

```
require 'OtConversion.rb'
# Import only VA-results from SATURN
  imp = OtSaturnImport.new
    imp.network  = [1,2]
    imp.filename = "#{$Ot.dirPrj}External User Data\\network.dat"
    imp.loadsFilename = "#{$Ot.dirPrj}External User Data\\EQ.txt"
    imp.loads = [1,1,2,1,42,1]
    imp.loadsOnly = true
  imp.execute
```

How to create a suitable file in SATURN with the results of an assignment was discussed in paragraph 2.2.3.

2.7.2 Types and type items

Next to the capacity indices found in the ASCII network file, two more link types should be defined in the project:

- simulation connectors;
- undefined links in the ASCII network file.

In SATURN connectors of the simulation network have no capacity index since it is assumed that no delays will occur on these connectors. To make sure that the connectors are given a linktype in OmniTRANS the property .connectorLinkTypeItem can be used.

Speed and capacity of the simulation connectors are given very high values to adopt the SATURN assumption that no delays occur at the connectors and vehicles are directly distributed to the network. Additionally, the distance of all connectors are overruled during the import and set to 0,1 km, independent of the true distance of the zone to the connected nodes.

The defaultLinkTypeItem is used for links of which no capacity index is found in the SATURN-data file. By applying a default linkTypeItem to these links, it is prevented that they will be ‘undefined’, which can lead to strange results.

Note that links which have a capacity index which is not defined in the Project Setup will be undefined in the network! It is important to check visually after the import that all links are defined.

If the SATURN template is used, the following properties and values must be specified in the job.

```
my_Import.linkType = 1
my_Import.networkType = 2
my_Import.turnType = 3
my_Import.nodeType = 4
my_Import.defaultLinkTypeItem   = 100
my_Import.connectorLinkTypeItem = 101
```

2.8 Reviewing the import

Differences between modelling in OmniTRANS and SATURN, and features in SATURN versions newer than version 10.3 may lead to problems during the import on three levels:

- errors;
- warnings;
- unreported differences.

These levels will be discussed briefly in the sub-paragraphs below.

2.8.1 Errors

Errors are generated if something in the code of your job is wrong. Examples are: syntax errors or invalid path names. If an error occurs, the job engine will abort. To fix the job, look which line causes the error and fix the problem.

2.8.2 Warnings

When the errors are fixed OmniTRANS will import the network, however, it may still generate warnings. It is important to take a look at these warnings. Due to the usage of different formulas and theories or different implementation of similar theories, differences may occur between the modelling in SATURN and the modelling in OmniTRANS. It is up to the user to decide how these differences influence the results and whether they should be overcome.

It is important to realise, that OtSaturnImport assumes a correct ASCII network file from SATURN. This means that the data should not only be correct from a data specification point of view (e.g. which column should be used to specify a specific parameter), but also from a modelling point of view (e.g. it is not correct to apply a green filter to a left turn which shares its approach lane with another turn).

When processing an ASCII network file with SATNET a .LPN file is created which in which all warnings and errors are listed. Some of these warnings will be generated by OtSaturnImport as well, however, most of these warnings are not. Hence, if the warnings of the .LPN file are ignored, it is likely that most of these problems are imported unnoticed into the new OmniTRANS project.

Most of the warnings which are generated during the import are of the data specification type. The most common warnings are summarized in Appendix A. Each warning is explained briefly and a possible solution is provided. For import convenience some warnings mention which line number of the data file generates the warning. The line number is specified at the end of the warning between square brackets. For example, [l3524] means line number 3524 of the ASCII network file.

2.8.3 Unreported differences

Even if all the errors and warnings are fixed, it is still possible that differences occur between the SATURN network and the OmniTRANS network. For example, Link Types in OmniTRANS are based on the Capacity Indices of SATURN. If no Capacity Index is specified for a link, OmniTRANS will apply the default link type to that link (in the SATURN template this is link type 100). If however a Capacity Index is found which is not defined in the Link Type & Modes tab of the Project Setup window, the Link Type will be set to undefined. This implies that this link will not be considered during the assignment.

To make sure that links are not accidentally undefined, it is recommended to set your link width to 4 or 5 after the import of the SATURN data file and switch of all other objects like nodes and centroids. In this way, undefined links can be seen easily.

2.9 Importing Trip Matrices

Trip Matrices in OmniTRANS are specified by four of the six dimensions; i.e. Purpose, Mode, Time, User. A trip matrix in OmniTRANS must be square and 'unique', unlike matrices in SATURN which can be stacked (vertical or horizontal).

In SATURN, trip matrices are stored in a different file than the network specifications, hence, another class called OtMatrixCube has to be used to import the trip matrices. In the job directory of the SATURN template a job is specified which can be used to import the trip matrices.

Note that it is necessary to convert the possibly stacked .UFM matrices to square ASCII matrices first (in SATURN). For example, consider a stacked trip matrix containing two square matrices called OD.ufm. First step is to convert this matrix into two matrices, e.g. LGV_od.ufm and HGV_od.ufm. Next these matrices should be converted into ASCII format, LGV_od.txt and HGV_od.txt. This all performed in SATURN (see paragraph **Fout! Verwijzingsbron niet gevonden.**). Finally, these two matrices can be imported into OmniTRANS using the job below.

```
writeln "Importing LGVs OD-matrix"
  mc=OtMatrixCube.new
  mc.import([1,1,2,11],$Ot.dirPrj+"External User Data/LGV_od.txt")

  writeln "Importing HGVs OD-matrix"
  mc=OtMatrixCube.new
  mc.import([1,1,2,12],$Ot.dirPrj+"External User Data/HGV_od.txt")
```

The results of the import can be found in the Data tab, by double clicking on one of the matrices (see below).

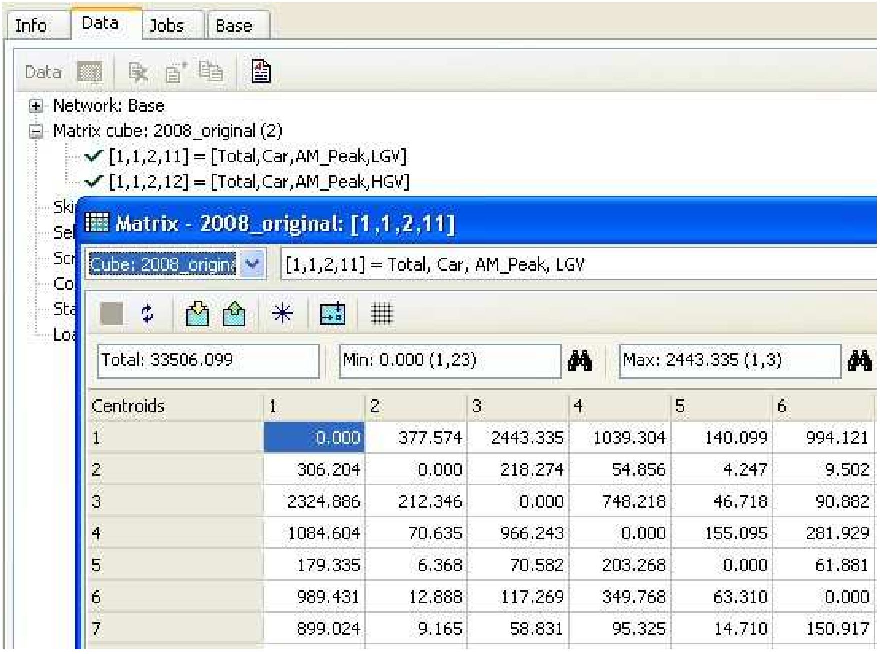


Figure 25 –Imported trip matrix

3 Modelling

3.1 Introduction

Having successfully imported the network it is now possible to model a highway assignment. This chapter describes how this can be done using parameters that origin from the original SATURN project.

This stage contains three phases:

1. Run an initial assignment;
2. Calibrate the project and solve possible problems;
3. Run the assignment.

3.1.1 OtTraffic class

In OmniTRANS a class called OtTraffic is available which is used for the static assignment of traffic to the network. The structure of this class is similar to the OtSaturnImport class which is discussed in chapter 2; create an instance of the class, specify the properties, perform the actual operations. An example is given below.

```
myAssignment = OtTraffic.new
  # specify the dimensions of the results explicitly
  # the dimensions of the network and OD-matrix
  # use these dimensions as well
  myAssignment.load = [1,1,1,1,1,1]

  # specify the assignment characteristics
  myAssignment.assignMethod = VOLUMEAVERAGING
  myAssignment.iterations = 10
  myAssignmnent.epsilon = 0.001
  myAssignment.saveIterations = true

  # specify the cost charactersistics
  myAssignment.pointsPerType = ArrayPPT
  myAssignment.routeFactors = [0,60,1,0]
  myAssignment.linkCost = [ [[1037],2] ]

  # apply junction modelling
  myAssignment.junctions = true

myAssignment.execute
```

In the following paragraphs issues related to traffic assignment are discussed, and the associated properties are explained. OtTraffic contains about fifty properties; in this chapter only the most important are discussed. For information about the other properties the see the description of the OtTraffic class in the Online Manual.

In paragraph 3.4 and 3.5 examples of a single user class assignment and a multi user class assignment are provided.

3.2 Costs

3.2.1 Different approaches for defining speed flow relationships

To run an assignment which takes into account the effect of traffic flow on the travel times the relation between speed and flow has to be specified for the links and/or junctions. There are several ways of doing this.

SATURN defines two different levels of detail in the network:

- a buffer network, which takes into account only the delays on link level;
- a simulation network, which takes into account the delays on turn level (junctions). Furthermore, flow delay relationships on link level can be specified for the simulation network as well.

In the UK, the speed flow relationships determined by COBA should be used. These are based on a different assumption:

- For rural roads speed flow curves are defined based on link geometry, and delays at junctions have to be calculated using junction models.
- For the urban areas the network is considered as a system of links and junctions, which are therefore incorporated in the speed flow relationships on link level.

Each approach has its advantages and disadvantages, noting that the two methods should not be simply mixed or combined. Thus, if junctions are modelled explicitly (e.g. the simulation network) speed flow curves should be based on delays at link level only, so that junction delays are applied twice. In the buffer network, speed flow curves should be used which incorporate the average delay effect of intersections.

3.2.2 SATURN link types

Since the speed flow relationships of COBA are recommended in the UK, the remainder of this paragraph is based on these curves. In the COBA Manual (2002), the following road classes are specified:

1. Rural single carriageway
2. Rural all-purpose dual 2-lane carriageway
3. Rural all-purpose dual 3 or more lane carriageway
4. Motorway, dual 2-lanes
5. Motorway, dual 3-lanes
6. Motorway, dual 4 or more lanes
7. Urban, non-central
8. Urban, central
9. Small town
10. Suburban single carriageway
11. Suburban dual carriageway

These road classes are also used in the SATURN manual, however, values are specified for multiple variable combinations per road class, leading to 24 different link types. For example, road class 9 is divided into 3 link types, based on 35%, 60% or 90% development in the area.

The link types specified in the SATURN manual are widely adopted in the UK, and therefore they are used as predefined link types in the SATURN template.

3.2.3 Predefined flow delay curves

SATURN and COBA work with the speed – flow curve paradigm; OmniTRANS uses the flow – delay curve paradigm. Accordingly, to help users convert speed – flow curves to flow delay curves, a job file is provided in the SATURN Template called PointsPerType.

This job does not have any executable code, but an array called ArrayPPT which contains the flow – delay representation of 24 curves coming from their speed – flow representation. These curves can of course be amended if required. Any job that needs to use this set of curves can do so using the following code example:

```
require 'pointsPerType.rb'
...
myAssignment.pointsPerType = ArrayPPT
...
```

The first line is used to make the file 'visible' to the current job. Thereafter, the OtTraffic property .pointsPerType simply refers to this array rather than have the curves defined explicitly.

The property OtTraffic.pointsPerType specifies coordinates (per link type) between which linear functions are assumed. The x-coordinate is the volume/capacity ratio (0 is freeflow and 1 is at capacity), and the y-coordinate is the travel time/free flow travel time ratio (e.g. 1 means the travel time equals the free flow travel time, 2 means that the travel time is two times the free flow travel time). Since it is based on coordinates, it can represent all kind of flow delay curves, independent whether they are linear, logarithmic or whether the derivative is discontinuous, and the accuracy depends on the number of coordinates in the array. The only requirement is that the values of the y-coordinates should be ascending.

Note that the specified curves (i.e. COBA curves) are not identical to the curves in SATURN. This is due to the fact that curves in SATURN are of the form:

$$t = AC^n + t_0$$

Due to the power n, SATURN can not exactly represent the COBA curve, and therefore tries to fit it as good as possible.

The values of capacity and free flow speed in the SATURN manual are chosen such that the curves optimally reflect the COBA curves. As a result, these values may differ a little bit from the COBA values.

Below the speed flow relationships based on COBA and the curve resulting from the SATURN values are shown (left Y-axis), as well as the associated flow delay curves (right Y-axis).

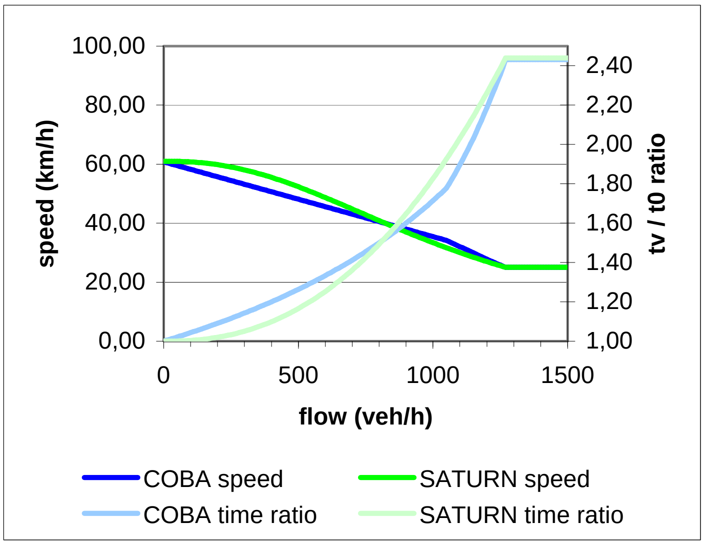


Figure 26 – Flow delay curves for link type 14

3.2.4 User defined flow delay curves

The predefined flow delay curves values form a good starting point for the first assignments to your network in OmniTRANS. Nevertheless, the variables used in the predefined curves are simply copied from the COBA Manual. This has two important drawbacks:

- The values do not take into account local characteristics like for instance the percentage of HGVs, or the hilliness of the area.
- Urban curves do reflect junction delays, while rural curves do not. This can lead to a bias in your assignment.
- For flows greater than capacity or speeds less than the minimum speed, the slope is taken as 'nearly horizontal'.

OmniTRANS provides a user friendly and accurate way to overcome these problems and to get your site specific COBA curves into the project environment. The COBA formulas can be coded into Ruby; in this way, it is possible to use these formulas for multiple projects, enabling the user to use location specific parameter values. The results of these formulas can be saved into an array which can be used by the OtTraffic.pointsPerType property to represent all flow delay curves, provided that travel time increases with flow.

3.2.5 Impedances and delays

As well as flow related costs, additional costs can be applied to model tolls (monetary costs) and delays (temporary costs). The import of additional costs (delays, impedances) on links and turns is not included in the OtSaturnImport class and should therefore be specified manually in the OmniTRANS project. In SATURN these additional link and turn costs are specified in the 44444 section of the ASCII network file.

Turn impedances

It is possible to specify impedances or bans to turns. To do this, add a turn to a node and double click on this link to open the turn window. If the turn should be a banned turn, specify a negative value in the impedance field (e.g. -1). If the turn should only have a delay, specify the value of this penalty in hours (e.g. a delay of 5 minutes is specified as 0.08333). Note that turn impedances are not multiplied by any cost factor (see paragraph 3.2.6). Turn impedances are Mode and Time related.

LinkCost

Next to the appliance of additional costs to turns, it is also possible to define additional costs on a link. These costs can be specified in the network by manipulating speed, distance and the flow delay curve of the link, but easier is to use the OtTraffic.linkCost property. LinkCosts are multiplied by the γ of the routeFactors (see paragraph 3.2.6), and can therefore be either in monetary or time units (however, they should be all of one kind through the network).

3.2.6 Cost perception

In the previous subparagraphs all the possible costs were discussed. In this subparagraph the perception of these costs will be discussed. In SATURN the perception of costs is specified in the 88888 section of the data file (value of distance and value of time). In OmniTRANS these factors should be specified in the assignment jobs using the property OtTraffic.routeFactors.

OtTraffic.routeFactors converts travel times and distances into generalised costs. The input is an array [α,β,γ,δ] which defines the value of:

α. distance (in kilometres);
β. time from the flow delay curves (in hours; if the value per minute is desired, this value should be multiplied by 60);
γ. generalised costs per link (from OtTraffic.linkCost);
δ. fixed costs on the network, based on a user defined cost specification.

3.2.7 Example

Consider the following example. In the SATURN ASCII network file the following specification is given for generalised costs.

```
44444
      1000 1015       $40
      1016 1237        60
      1237 1016        60
99999
...
88888
    1    1 1.00 0.20 0.5
99999
```

The perceived value of time is 0.5 pence per minute (i.e. 30 pence per hour) and the perceived value of distance is 0.2 pence per kilometre. The network contains one toll road (link 15), which costs 40 pence in direction one and one bridge (link 16) which results in an average delay per vehicle of 1 minute.

It is decided to generalise all costs and delays into monetary costs in pence. Thus, the route factor array will be : [0.2 , 0.5 * 30 , 1 , 0].

OtTraffic.linkCost can be used either to specify additional costs or delays, however not both. Since two different penalties types are specified in the SATURN project, it is not possible to use OtTraffic.linkCost for both. Therefore OtTraffic.linkCost will be used to specify the toll, while a turn impedance is used to define the delay for the bridge.

The γ of the route factors is 1, hence, the linkCost should be 40.

Turn delays are always specified in hours, and therefore the impedance should be set to 1/60 = 0.01666. Note that although only two digits will be shown when the attribute editor is reopened, all digits are stored in the database.

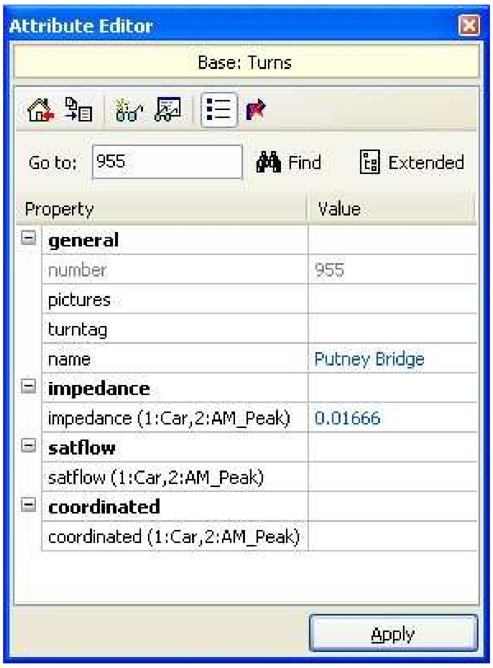


Figure 27 – Turn impedance

To conclude, the data from the ASCII network file as shown at the beginning of this paragraph (3.2.7) should be specified in OmniTRANS in the assignment job as:

```
myAssignment = OtTraffic.new
   ...
   myAssignment.routeFactors = [0.2,30,1,0]
   myAssignment.linkCost = [ [[15,1],40] ]
   ...
myAssignment.execute
```

3.3 Important OtTraffic properties

3.3.1 Dimensions

OtTraffic contains many properties which can be used to adjust the assignment to your needs. All these properties are described in the Online Manual and it is recommended to spend some time reviewing these to get familiar with their scope.

In this section the most important properties are discussed, so after reading this chapter it should be possible to understand the assignment job of the SATURN template, and to run an assignment in OmniTRANS based on the converted SATURN data.

The most elementary assignment is specified below. Of course, more properties can be specified; these will be discussed consecutively.

```
myAssignment = OtTraffic.new
     myAssignment.load = [1,1,1,1,1,1]
myAssignment.execute
```

In this example an instance of the OtTraffic class is created named ‘myAssignment’. The only specified property (OtTraffic.load) defines the pmturi-location at which the resulting flows of the assignment should be stored. The dimensions of the trip matrix (purpose,mode,time,user) and the network (mode,time) are automatically derived from the load dimensions, since they are not specified explicitly. The result of this job is an All-Or-Nothing assignment.

3.3.2 Assignment possibilities

In OmniTRANS there are four possible types of traffic assignment;
- All or nothing;
- Incremental assignment;
- Volume-Averaging assignment;
- Equilibrium assignment.

The (user) equilibrium assignment is in fact a form of volume averaging using the Frank Wolfe algorithm to determine the optimum φ. At the moment a social equilibrium has not been implemented as property of OtTraffic. It is however possible to implement a system equilibrium by combining OtTraffic and ruby routines.

The assignment method is specified using the property assignMethod, which needs either one or two arguments as input. The first argument (mandatory) is the method of assignment, the second (optional) specifies whether and which stochastic effects should be incorporated. Below some examples are shown:

```
myAssignment.assignMethod = ALLORNOTHING
myAssignment.assignMethod = INCREMENTAL | UNIFORM
myAssignment.assignMethod = VOLUMEAVERAGING
myAssignment.assignMethod = EQUILIBRIUM | NORMAL
```

The parameters of the distributions for a stochastic assignment are specified using OtTraffic.spread (deviation of the distributions) and OtTraffic.seed (random seed number).

Convergence can be controlled using the properties OtTraffic.epsilon and OtTraffic.iterations. Epsilon is the OmniTRANS equivalent of the Delta Function in SATURN and iterations is the same as NITA in SATURN. Note that epsilon is only considered if OtTraffic.iterations is specified, if the property .epsilon is specified, but .iterations not, only one iteration will be performed (i.e. all-or-nothing).

3.3.3 Miscellaneous

The OtTraffic.junctions property can be used to specify whether junction modelling should be applied. For further analysis of the results of an assignment it may be useful to store the results of all iterations. By default, only the results of the final iteration are stored, to minimise calculation time and data storage. To save the results of all iterations OtTraffic.saveIterations should be set to true.

3.4 Single user class assignment

Below an example of a single user class assignment by OtTraffic is given, using the discussed properties.

```
require $Ot.dirJob + "pointsPerType.rb"  # refer to

myAssignment = OtTraffic.new

   myAssignment.load = [1,1,1,1,1,1]

   myAssignment.assignMethod = VOLUMEAVERAGING
   myAssignment.iterations = 10
   myAssignmnent.epsilon = 0.001

   myAssignment.pointsPerType = ArrayPPT
   myAssignment.routeFactors = [0,60,1,0]
   myAssignment.linkCost = [ [[1037],2] ]

   myAssignment.junctions = true
   myAssignment.saveIterations = true

myAssignment.execute
```

For a complete review of the used properties see the Online Manual or the paragraphs above. The most important issues of these paragraphs are listed below.

The .load property is used to specify in which PMTURI the results are to be stored. If the locations of the network and trip matrix are not stated explicitly, OtTraffic will derive them from the .load property.

The .pointsPerType property is used to define the flow delay curves and expects an array. This can either be specified directly behind the property, or by specification of the name of an array, which has been defined elsewhere, either in this job, or in a different job. In this example the array ArrayPPT (discussed previously in paragraph 3.2.3) is used.

The property .routeFactors is used to specify convert different quantities into generalised costs. In the example above, distance is not taken into account, travel times are converted into minutes (value of 60 in the second field). An additional link cost has been specified for link 1037 (both directions) of two minutes.

The assignment takes junction modelling into account, and the results of every iteration are stored.

3.5 Multi user class assignment

3.5.1 Why and how are different user classes distinguished

In the previous paragraph an example was given of an assignment for a single user class using the class OtTraffic and some of its properties. By combining OtTraffic with Ruby scripts it is possible to perform a multi user class assignment. An example job is provided in the SATURN template.

Before going into detail of the assignment, it has to be clear why user class are to be distinguished. In SATURN user classes are distinguished considering the following issues:

- modal split influences (e.g. income, car-ownership);
- vehicle types (e.g. cars or trucks);
- route choice criteria (e.g. minimum distance or minimum time);
- network restrictions (e.g. access to city-centre or no access to city-centre).

They will be discussed in the following subparagraphs.

Modal split influences

OmniTRANS contains a special class called OtChoice to model the modal split process. This is not directly relevant for the assignment process and therefore lies outside the scope of this document.

Vehicle types

There are two dimensions (*mode* and *user*) which can be used to distinguish between different vehicle types, it is up to the user to decide which dimension is used. For a true multi user class assignment in which all user classes influence each other, only the User type can be used. This will be explained below.

The most important difference between both dimensions is that networks are *mode* related, but not *user* related (see also paragraph 2.4.4). In OmniTRANS every assignment is based on one single network, i.e. one combination of mode and time. In this way it is possible to specify different speed limits per mode (cars and trucks) or to specify different capacities per time (for instance due to lane restrictions for trucks during the peak period).

If user classes are associated with different networks, it is not possible to assign them all in one assignment. It is possible however to run an assignment for one mode first and to use these results as fixed flows to the network of the second user class. This is comparable to the way SATURN adds transit flows as fixed flows to the network.

The drawback of this method is that the assignment of the first mode is not affected by the assignment of the second. This can be overcome by assigning multiple user classes at once. To do this, the User dimension should be used instead of Mode.

Route choice criteria

The route choice criteria which are used in the path building process can vary among user classes. These differences can be specified by the value of time and the value of distance.

Below, the values of time (respectively 1 and 2 pence per minute) and the value of distance (0.45 and 0.35 pence per kilometre) are specified.

```
88888
    1    1 1.00 1.00 0.45
    2    2 1.00 2.00 0.35
99999
```

In OmniTRANS this should be coded (per user class) as:

```
assignmentUser1.routeFactors = [ 0.45 , 60]
assignmentUser2.routeFactors = [ 0.35 , 120]
```

It is not possible to specify the .routeFactors property more than once to the same instance. However, using multiple instances implies that multiple assignments have to be performed. This is in fact what is done in the multi user class assignment job. It will be discussed below in paragraph 3.5.2.

Network restrictions or penalties

Certain areas within a network (e.g. city centres) or links (e.g. bridges or tunnels) may be banned or penalised to for certain user classes. As with the route choice criteria it should be coded per user class in the assignment job. It is not possible to specify a banned turn in OmniTRANS per user class, since turn penalties are not defined in the assignment job, but in the network. Therefore, OtTraffic.linkCost should be used to define penalties per user class.

3.5.2 Predefined MUCA-Job

In the SATURN template a job is stored to model multi user class assignment. The principle of this job is explained below:

1. determine the costs based on the total flows on the network (i.e. free flow in the first iteration).
2. calculate the shortest paths per user class and assign the auxiliary flows to these routes.
3. take a certain proportion of the loads of the previous load per user class and the auxiliary flow per user class
4. sum up all these resulting flows (i.e. of all user classes).
5. use these resulting flows in step 1.

To use the predefined job, follow the steps below.

First step is to specify the PMTURI-dimensions. This should be done in the PMTURI dimension specification block. The values specified are compatible with the template, but it is possible to change them to your needs, for example by adding a third user class. Make sure that if additional or different parameters are specified, they are changed on the dimensions tab of the Project Setup window as well.

Second, specify an array containing the speed flow relationships of your network, using the .pointsPerType property. In the template an array is available in which flow delay curves are specified for all the link types of the SATURN template. However, this array should be used with only to produce some initial test results. For realistic modelling it is recommended to specify your own curves, taking into account the recommendations of paragraph 3.2.4.

Third, set the number of iterations (default is 10) at line 52.

Fourth, specify the user characteristics. If user classes are added, additional if statements should be specified here for these user classes as well. An example is given below.

```
# first user class
        if u == aUser[0] then
          assign.routeFactors=[0.45 , 60 , 1 , 0]
          assign.linkCost=[ [[1037],2] ]
        end
# second user class
        if u == aUser[1] then
          assign.routeFactors=[0 , 2*60 , 2*1 , 0]
          assign.skipLinks=[1037]
        end
```

Finally, it is possible to store the results of each iteration (per user class). Note however that this will slow down the algorithm, as it requires more database transactions. To save the iterations, simply remove the # at the beginning of line 98.

If all the parameters are set correctly, the multi user class assignment can be performed. Note that possible warnings are only generated for the first assignment, since it will slow down the process, and the warnings will not change between iterations.

4 Output

4.1 Introduction

OmniTRANS provides a range of possibilities to show assignment results graphically and in reports. The way graphical output is visualised is determined by settings in the graphics toolbar. For example, scale of bandwidths and values are set by activating the bandwidth control toolbar (use right mouse button). To specify which results should be shown, the most right button of this toolbar leads to the link bandwidth designer. In the next paragraph some examples of useful visualisation possibilities are given. Reports are defined using the Report Designer. This will be discussed in paragraph 4.4.

Details of all menus are extensively described in the Online Manual. Moreover, the best way to get familiar with these menus is just trying and see what happens. Most important ... is that the PMTURI-dimensions in the designer menus should correspond with the dimensions specified in the jobs.

4.2 Visualisation

4.2.1 Finding objects

To find an object in the network, select the object type, zoom in until only a few objects are visible, open the Find Tool window, fill in the number and press Enter.

Figure 28 – Find Tool

To increase computation speed in most situations the centroids will be renumbered. The original centroid numbers are stored in the name field of the centroids. It is not possible to find a centroid by entering a number in the name field; it will only rename the centroid, and not jump to the centroid. Therefore it is recommended to use the attribute editor of the nodes, because a centroid is always connected to a node.

4.2.2 Link bandwidth designer

V/C ratio

In Figure 29 an example is given of the visualisation of the results of a volume averaging assignment. The width of the bandwidths represents the loads, while the colours represent the volume capacity ratio.

Figure 29 – Visualisation of Volume/Capacity ratio

The used bandwidth style (Association tab) is 'single', the data specification is shown below in

Figure 30 – Specifications of V/C ratio

As can be seen in the colour box, it is possible to specify mathematical equations which use the associated quantities, to create more advanced plots.

Multi User class assignment

In Figure 31 an example is given of the results of a multi user class assignment for two user classes, of which one user class is not allowed to travel across the link in the centre of the picture.

Figure 31 – Visualisation of multi user class assignment

The chosen bandwidth style (Association tab) is 'stacked', the data specification is shown below in Figure 32.

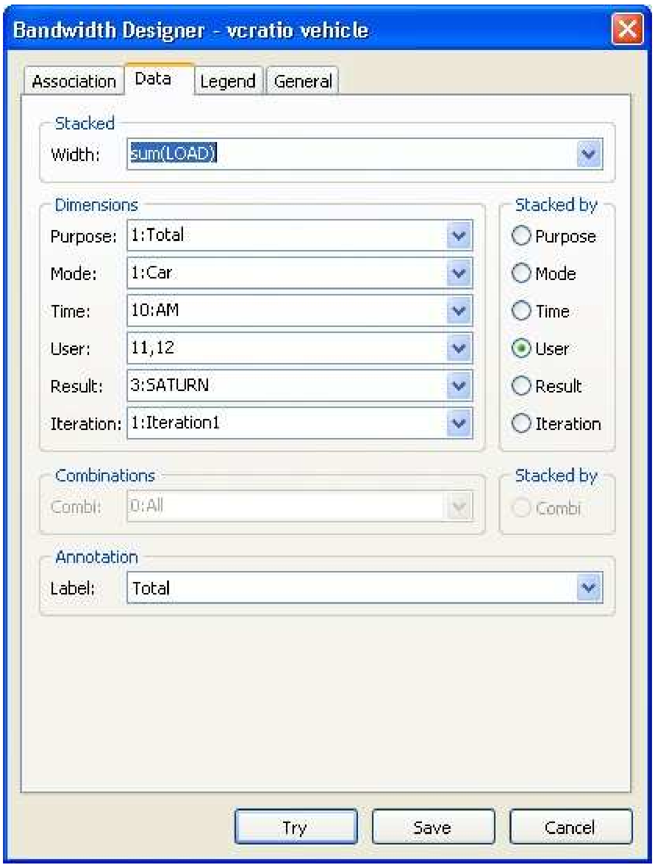


Figure 32 – Specifications of multi user class assignment

Comparing results

The PMTURI-structure is very useful to compare different results. A simple comparison of two assignments is shown in Figure 33. In this case the influence of the value of time and distance (i.e. OtTraffic.routeFactors) was studied.

Figure 33 – Visualisation of comparing two assignments

The chosen bandwidth style (Association tab) is ‘compare attributes, the data specification is shown below in Figure 34.

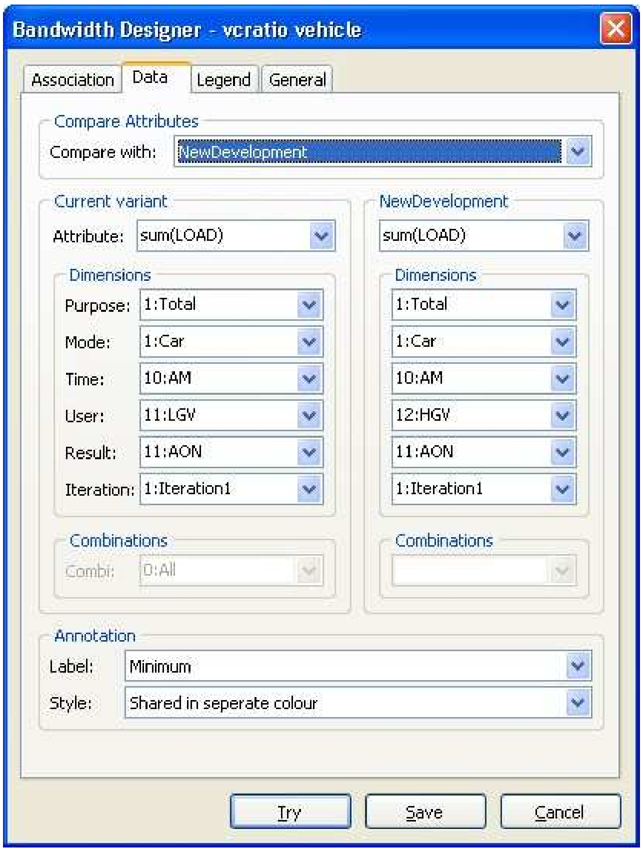


Figure 34 - Specifications of a comparison of two assignments

4.2.3 Junction editor

As well as the visualisations of the network using the designers mentioned above, it is possible to visualise results on a more detailed level, by using the Junction Editor, as shown below.

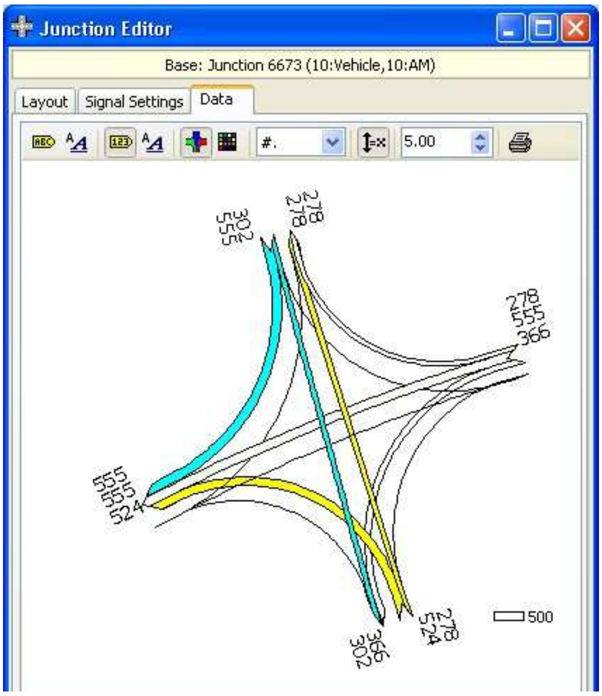


Figure 35 – Visualisation of flows on a junction

4.3 Analysing the results

4.3.1 Selected link analysis

A very common analysis is a 'selected link' analysis. In this analysis it is determined where the traffic on a certain link is coming from and going to. The SATURN template contains a predefined job using the property OtTraffic.selectedLinks. To run the job the following steps should be followed:

- Select a link in a certain direction in the network;
- Run the job 'select link car (volume-averaging).rb';
- View the results in the network using a bandwidth designer.

An example of the results is given in

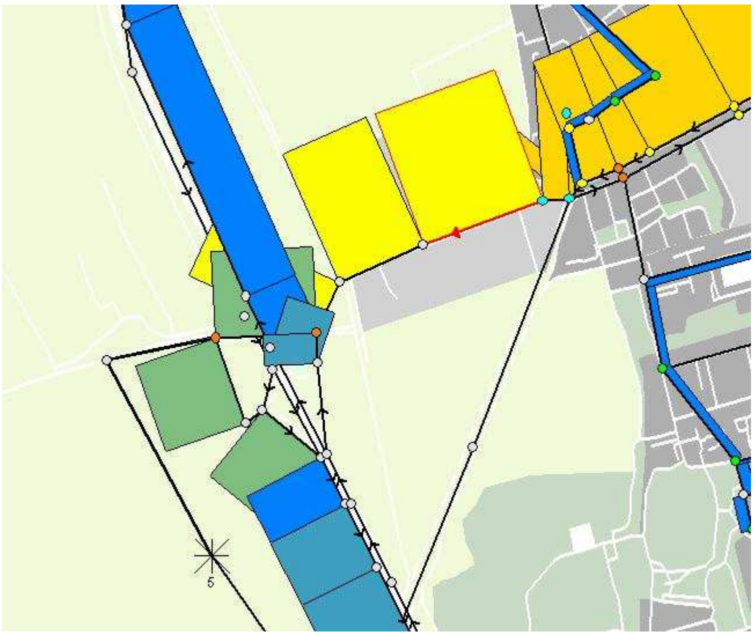


Figure 36 – Results of a select link analysis

Of course it is possible to use this property with other assignments methods (e.g. all or nothing assignment).

4.4 Reports

4.4.1 Introduction

OmniTRANS provides a Report Designer that lets you design and build your own reports; this is done using a 'Wizard' which guides you though the specification process. These can be saved and re-used when required.

Carkilometers

saturn3

Base

Roadtype	Carkilometers
Bus and bicycle	
Bus only	
Connector	33782
Cycle track	
Highway 1	12751
Highway 1 with cycleway	8874
Highway 2	5177
Local road mixed traffic	18271
Local road with cycleway	8005
Main road 1 with cycleway	10454
Main road 2 with cycleway	10483
Main road 3 with cycleway	3655
Motorway 2	35657
Motorway 3	68199
Motorway 4	7535
Rail	
Walk	
	222843

Figure 37 – Report

The general structure of a report is as follows:

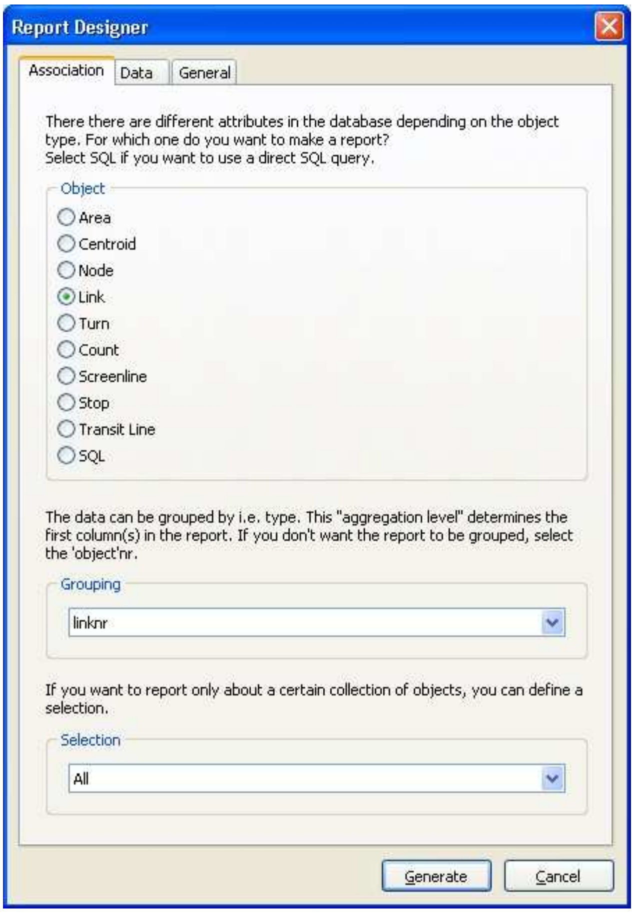
- Reports are generated for a single network object only, e.g. Links.
- Reports comprise one or more columns. The first column represents the primary aggregation field. It is possible to aggregate the chosen object on some attribute or type, or to show each individual object. In the example shown above the Link Loads have been aggregated by RoadType;
- The second (plus) columns in the table are used to show selected attributes associated with the network object. In the example the attribute cumulative 'Load' is shown in the second column;
- Reports can be sorted on any column;
- Reports can be constrained to a selection of the data.

4.4.2 The Report Designer

Start Report Designer

To activate the Report Designer choose Reports | Report Designer from the main menu (or press Ctrl+R).

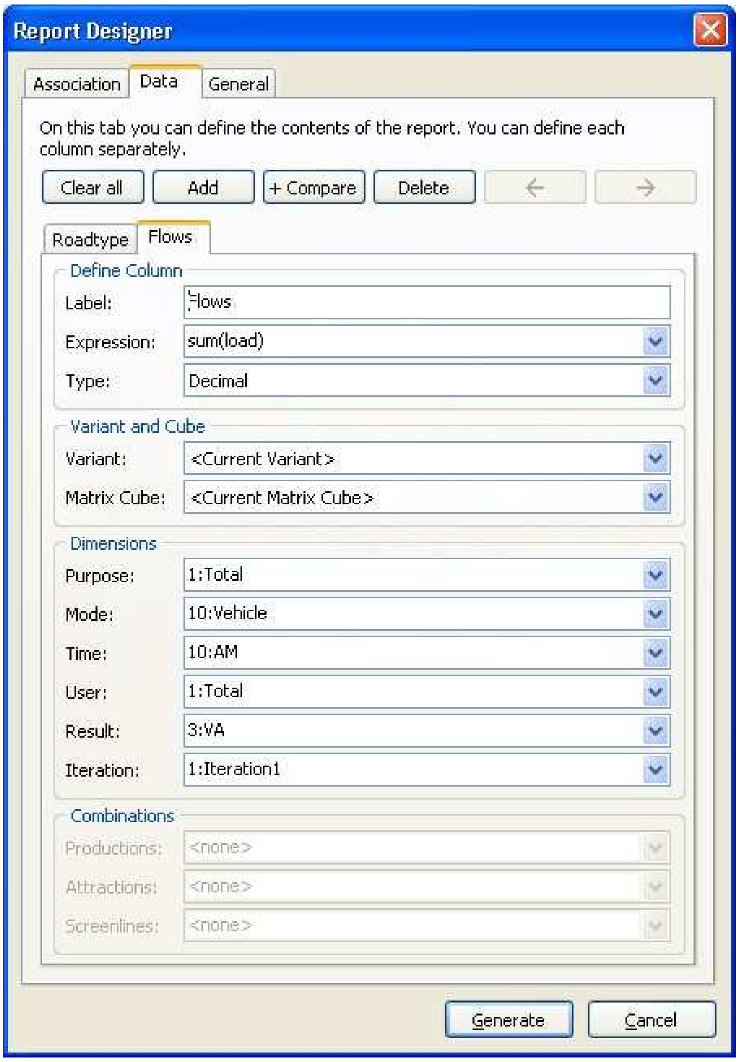
Association



In this window an Object Type must be selected to use when generating the report. The window also allows you to select the grouping (e.g. for links; linknr, linknr; direction, roadtype) and the selection (e.g. All, current selection etc.)

OmniTRANS

Data



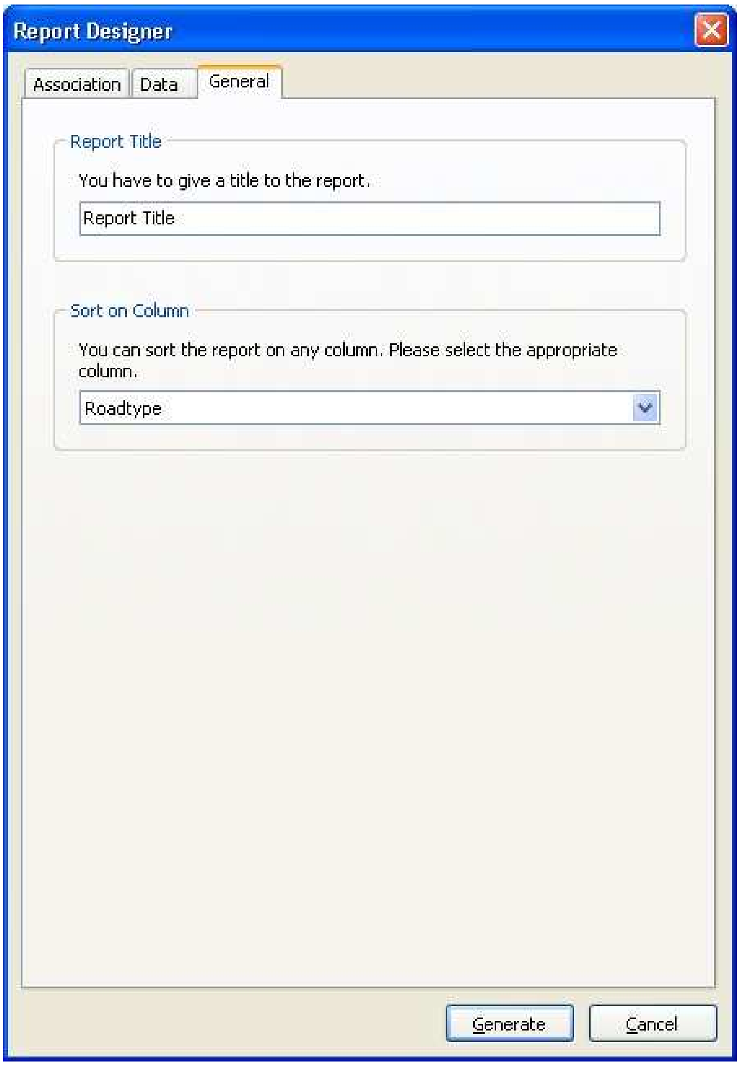
This view is used to specify how many columns are to be put in the report and the data to be used to construct each column. A separate 'sub-tab' will be generated for each column that is defined. Per column can be specified which data the segment should be showing.

The specification of the segment data is defined using the following controls:

- Define Column:
  - Name: User defined name for the variable. This text string will be used to annotate the column headings in the output report.
  - Expression: Select the data field to be used from the drop-down menu. This shows data in the database that is relevant to the chosen data type. This field is disabled when defining the first tab/column, since the first column has effectively been defined by the Primary Aggregation variable (above).

- Dimensions: If the chosen data item has a possible PMTURI combination then define. This field is disabled when defining the first tab/column as the first column has effectively been defined by the Primary Aggregation variable (above);
- Combinations: If the chosen data item has a Combination then define. This field is disabled when defining the first tab/column as the first column has effectively been defined by the Primary Aggregation variable (above).
- Total: To indicate what should be shown at the end of the report, under the column, either:
  - The count of the instances of the variable;
  - The column sum;
  - The average value;
  - Nothing at all;
- Type: To specify how the data in the column should be presented:
  - Integer: No decimal places;
  - Float: With decimal places (default of 2 decimal places);
  - String: For text based attributes.

General



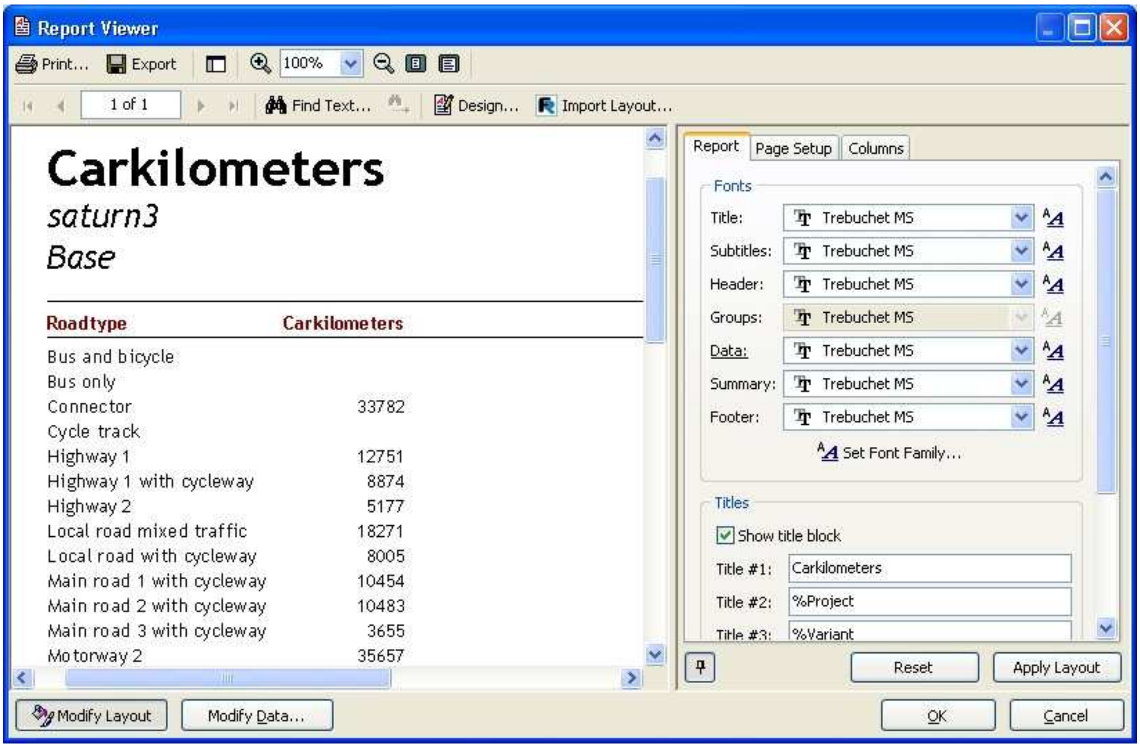
In the final view the following issues can be specified:

- Report Title: The title is printed at the top of generated pages;
- Sort on Column: Choose the column on which the report is sorted. By default it will be the first column;

Finally, the Report can be previewed and saved.

4.4.3 The Report Viewer

The Report Viewer window offers a print preview for reports and offers options to change the appearance of the report.



From the Report Viewer window the report can be sent to a printer or plotter or exported as a table to various file formats.

Appendix A

This appendix lists the most common warnings that are generated by the OtSaturnImport class. For each warning a brief explanation of the problem is given, what the consequences of this problem are and what should be done to fix the problem.

Anode equals Bnode ... is this correct?

Problem

In the SATURN 10.3 manual is stated that if in the 22222 data field the Anode is equal to the Bnode, the centroid is connected to all links both to and from the Anode. However, the implementation in SATURN seems to be different. It might be worthwhile to check the created connectors for this centroid.

Consequence

The import creates connectors both to and from node A, as is specified in the manual.

Action

Check whether the connector configuration is how it was intended to be.

A very long line has been detected...

Problem

This warning occurs when very long comment lines are used in the import file. This can result in errors and the rest of the line is therefore ignored.

Consequence

None

Action

To avoid this message, split up long comment lines in multiple comment lines.

Changed number of signs...

Problem

The specified number of yield/stop signs on a junction is not valid.

Consequence

OmniTRANS removes conflicting signs.

Action

Check whether the right sign has been removed. If not, use the junction editor to model the junction as desired.

Duplicate link ... has been found

Problem

It is not allowed to specify more than one link between two nodes. This warning is mostly due to a link being defined in both the 11111 and 33333 data fields.

Consequence

Only the first link specification in the data file is imported, any others are ignored. This behaviour is similar to SATURN.

Action

Determine where the duplicate links are specified in the data file, and delete the redundant links, or if all links are desired add a dummy node into the links.

Node .. is specified in 55555, however is never used in the network [linenr]

Problem

In the 55555 data field of the data file a node is specified which is not used in the network, i.e. it is not connected to any link.

Consequence

This node will not be imported into the OmniTRANS network.

Action

Determine whether the isolated node is intended or not. It is possible that a node number has been mistyped or a link is accidentally missing.

Node .. does not have valid coordinates and will be placed with an offset starting at (0,0)

Problem

If the coordinates of a node or centroid are not thought to be invalid (usually because none have been defined), the offending node/zone will be given coordinates near the origin (0,0) of the network drawing. SATURN automatically interpolates the undefined coordinates of a centroid of its neighbours, and therefore a network that is displayed correctly in SATURN may appear strange after importing it into OmniTRANS, especially if the coordinates of the network are much greater than zero.

Consequence

This action will have no consequences for any assignment, since the length of connector links is automatically set to 100 meters. However, as stated, the display of the network will look odd.

Action

Nodes and centroids can be located at their preferred location using the Graphical Interface. To do this, click on the node or centroid button and the edit object button as shown below.

Now the nodes/centroids can be selected and dragged to their appropriate position.

Link ... is bus-only. The link has been removed...

Problem

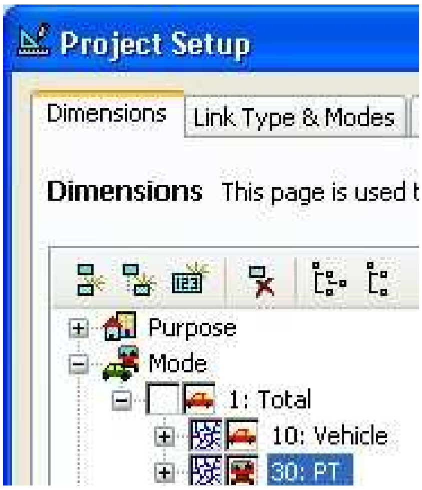
In the current version of OtSaturnImport it is not possible to import bus links, since they are related to another network (i.e. different Mode dimension).

Consequence

To avoid that traffic is assigned to these bus only links, the links are not imported. If a link is bus only in direction one and available for all users in the direction two the link is imported as a one-way link in direction two.

Action

If it is decided to input the public transit network manually into OmniTRANS, in the project setup a new mode type item should be added, including the network and transit flag (see below).



Now it is possible to specify one or more new link types in the Link Type & Modes tab, which can be used explicitly by busses. To the existing link types the bus mode should be attached.

Node ... has more than four arms. Please check the imported junction...

Problem

The junction modelling theory which is implemented in OmniTRANS can not deal with junctions with more than four arms.

Consequence

If the junction is not ammended, no junction modelling is applied and as a result no delays will be calculated for the junction. However, banned turns and manually defined turn impedances are taken into account during the assignment.

Action

This warning can be solved in two ways. First one of the arms can be disabled. It is recommended to do this for junctions with four or less ingoing links. By disabling the outgoing links, best results will be obtained. If more than four links are ingoing, the link with the smallest flow can be disabled. To do this, open the junction editor, specify a lane and set *enabled* to *no*. Second, the junction can be split into different nodes, by adding an additional node. See the Technical Note on Junction Modelling for more information.

Node ... is a junction, however has buffer network links...

Problem

The junction has been detected to have a buffer link as one or more of its arms. Since turns are not defined in SATURN for buffer links, not all turn movements for this node are known. Hence, junction modelling can not be applied to this junction.

Consequence

All turns are removed (including banned turns and turn impedances), and no delays will be calculated for this node.

Action

First it has to be verified whether there should be turn bans or impedances for this node. If not, there will be no influence on results. If there should be banned turns or impedances, they should be added to the network after the import.

The distinction between buffer and simulation network (type 2) in OmniTRANS is only used to visualise the original SATURN input. If junction modelling is desired for this node, turns from and to the buffer links should be coded manually in the network. Consecutively the junction can be modelled using the junction editor.

Node ... is a junction with merging flows. All junction information has been removed.

Node ... is a junction with weaving flows. All junction information has been removed.

Problem

In SATURN a special turn priority marker is defined for merging and weaving. OmniTRANS does not take merging or weaving on motorways into account for static assignments. For dynamic assignments both are taken into account, however, not based on the static junction modelling of OtTraffic.

Although theoretically merging on a freeway is a priority situation, the priority junction of the junction modelling should not be used. The junction modelling is based on theories which apply to the surface street network, not to motorways, and therefore coding merges of freeway on-ramps as a priority junction will lead to incorrect results.

Consequence

All junction data is removed, and no delays will be calculated for this node.

Action

Nothing has to be changed, this will lead to optimal results.

Turn movement ... has a green time AND is a filtered turn

Problem

A green time is defined explicit for this turn movement although a filter (continuously green) has been applied as well for this turn.

Consequence

The filter will be applied and the specified green times for this turn are ignored (similar to SATURN).

Action

Check whether either the green time or the filter has been defined accidentally.

Note

If a filter is specified for a turn which shares its approach lane with an (unfiltered) turn, no warning is generated and the filter is applied to both turns! It is important to verify that this is coded correctly in the ASCII network file. This can be done by checking the .lpn file of SATNET which generates 'serious warning 105' if this is coded wrong.

Total of green times ... exceeds the cycletime ...

Problem

The total of the green times and inter greens specified in the SATURN data file is greater than the specified cycle time.

Consequence

If the cycle time is exceeded, OmniTRANS calculates the cycle time as the sum of green times and inter greens.

Action

Because there are various reasons possible for exceeding the cycletime, correcting cycle times could make things even worse. Therefore the user must correct the input, otherwise strange and unrealistic results may be generated.
First it is possible that the total of green times and intergreens exceeds the specified cycletime. In SATURN this is automatically compensated by scaling of the specified green times (intergreens are kept fixed). In some circumstances this procedure may lead to wrong corrections, e.g. if one of the cycletimes is accidentally specified as 120 instead of 20 seconds, all cycletimes will be recalculated while in fact only the cycle time of one stage is wrong. Therefore OmniTRANS does not recalculate the cycle times automatically.

Total of green times ... is less than the cycle time ...

Problem

The total of the green times and inter greens specified in the SATURN data file is less than the specified cycle time. This may be on purpose, to incorporate the influence of for instance pedestrian crossings. However, if a stage is accidentally defined wrong, e.g. a green time of 45 seconds instead of 5 seconds, this will lead to wrong results. By specifying the property OtSaturnImport.adict, some control over the generation of this warning message can be exerted. A warning will be generated if the following equation is true.

$$cycletime - \left(\sum greentimes + \sum intergreens \right) > adict$$

If for instance differences of 10 seconds are considered to be due to pedestrian crossings, adict should be set to 10, and only larger differences will result in a warning.

Consequence

The capacity of the junction will be less than assumed. This might have consequences during the path building process.

Action

Check whether the differences occur to model crossings of pedestrians or because of mistakes in the data file. Correct the undesired differences in the junction editor or SATURN data file.

Signal specification is not valid. The Anode is equal to zero.

Problem

This warning is generated by a coding error in record type 3 of the 11111 data field. The number of node entries does not correspond with the NGM parameter. In most cases this problem occurs when signal stages have been redefined while the NGM parameter has not. In this occasion the stages are entered correctly and the signal specification can be used. It may be that not all the intended turns have been coded.

Consequence

The specified turns are used in the junction.

Action

Verify why the warning occurs, either because the number of node entries (NGM) is wrong, or because the Anode (and possibly Cnode) are not specified.