Veitch Lister Consulting Pty Ltd

27-03-2013

Zenith User Manual

Travel Demand Forecasting & Transport Infrastructure Planning

Brisbane Ground Floor, 80 Jephson Street Toowong Qld 4066 **Tel: +61 7 3870 4888** **Fax:** +61 7 3870 4446
Melbourne Level 7, 423 Bourke Street Melbourne Vic 3000 **Tel:** +61 3 9602 5200 **Fax:** +61 3 9602 5166

www.veitchlister.com.au

Zenith User Manual

Date	Revision	Prepared By	Checked By	Approved By	Description
27 February 2013	A	TV	JP	TV	Draft for comment

Contents

1. Getting Started

Welcome to the Zenith User Guide!

The aim of this User Guide is to help Zenith model users with the day-to-day tasks that confront modellers.

As a Zenith model user, you are also an OmniTRANS user. Welcome to the growing OmniTRANS community!

Scope of this User Guide

Travel modelling follows a fairly well practiced pattern:

1. Start by creating a new project (if necessary)
2. Create a new scenario within the project
3. Change some part of the scenario (network, demographics, policy, etc)
4. Run the model
5. Analyse model results
6. Produce graphics and other reporting material

This structure of this User Guide reflects this pattern.

Section 2 explains how to create new OmniTRANS projects,

Section 3 explains how to create new scenarios,

Sections 4 to 6 explain how to modify each of the model's key inputs: networks, demographics / land use, policy variables, and behavioural models,

Section 0 provides all important instructions on how to perform model runs,

Section 0 explains how to analyse model results through the use of thematic maps, reports and spreadsheets,

Section 9 covers important topics such as import / export, and model maintenance.

A Message for OmniTRANS Novices

This User Guide will not set out to teach you OmniTRANS. OmniTRANS has its own Getting Started Guide and User Manual, and VLC also run 3-6 day OmniTRANS courses.

OmniTRANS is, however, at the heart of everything described in this guide, and so a working knowledge of OmniTRANS is assumed. Where possible, we've provided links to specific sections of the OmniTRANS Manual, where you can catch up on OmniTRANS if you're a bit rusty.

Good luck!

- The VLC Team

2. Creating a New OmniTRANS Project

When starting out on a new modelling project, a Zenith model user never starts from a completely blank slate. There's always an existing OmniTRANS project to start from.

As a Zenith model licensee, you will have been provided with a base OmniTRANS project with a specific zone system and version number, eg. OtZenithVictoria_3477zone_v2.1.0. This is generally a good starting point for new modelling projects.

When starting a new modelling project, make a copy of this base project (simply copy the entire directory), and paste it somewhere meaningful on your local hard drive. Over time, your collection of OmniTRANS projects will grow, each one containing scenarios specific to a particular modelling project.

Naming Conventions

In terms of naming conventions, we recommend the following:

Ot[*ModellingProjectName*][*VersionNumber*]

The *ModellingProjectName* will describe the subject of the study, eg. "Metro" or "Rowville" or "DandenongExtn", etc.

The *VersionNumber* should always be of the form Vx.y.z, where x, y and z are the major, minor and incremental version numbers (eg. V2.1.0).

Version Numbers

Zenith employs a three stage version number system (eg. V2.1.0), with major, minor and incremental version numbers.

The major version number typically increments once annually, and represents a major release.

A minor version number is incremented when the model or software is upgraded.

The incremental version number is incremented when a bug fix is released.

Whenever a new release is made, VLC will release a list of associated changes.

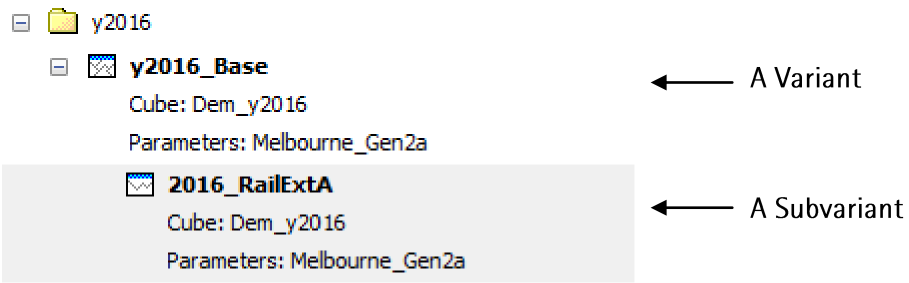
<u>Step-by-step</u>

1. Which OmniTRANS project do I want to use as my starting point?
   a. The latest base project (eg. OtVictoriaV2.0.0)?
   b. Another project (eg. OtMelbMetroV2.0.0)?
2. Use Windows Explorer to make a copy of the chosen project directory.
3. If the project contains many unneeded variants, you can be selective about which variants to copy, but this means choosing which folders to copy within the project.
4. Assign the new project an appropriate name. It is important to include the version number, to indicate which model version a project is based upon.

3. Creating New Scenarios

Creating a new scenario is one of the first tasks faced by any modeller.

OmniTRANS offers a powerful mechanism for storing scenarios: **variants and subvariants**.



The OmniTRANS Manual provides detail on exactly what is stored in a variant versus a subvariant *[Basic Concepts -> Projects and Variants]*, and the relationship between the two.

We will focus more on when to use variants and when to use subvariants as a Zenith model user.

The key principles to remember are that:

- In OmniTRANS, each variant has its own network, separate from all others.
- A sub-variant does not have its own network, but instead uses the network of its parent variant. Therefore, if you open and modify a subvariant network, you are actually modifying the network of the parent variant!

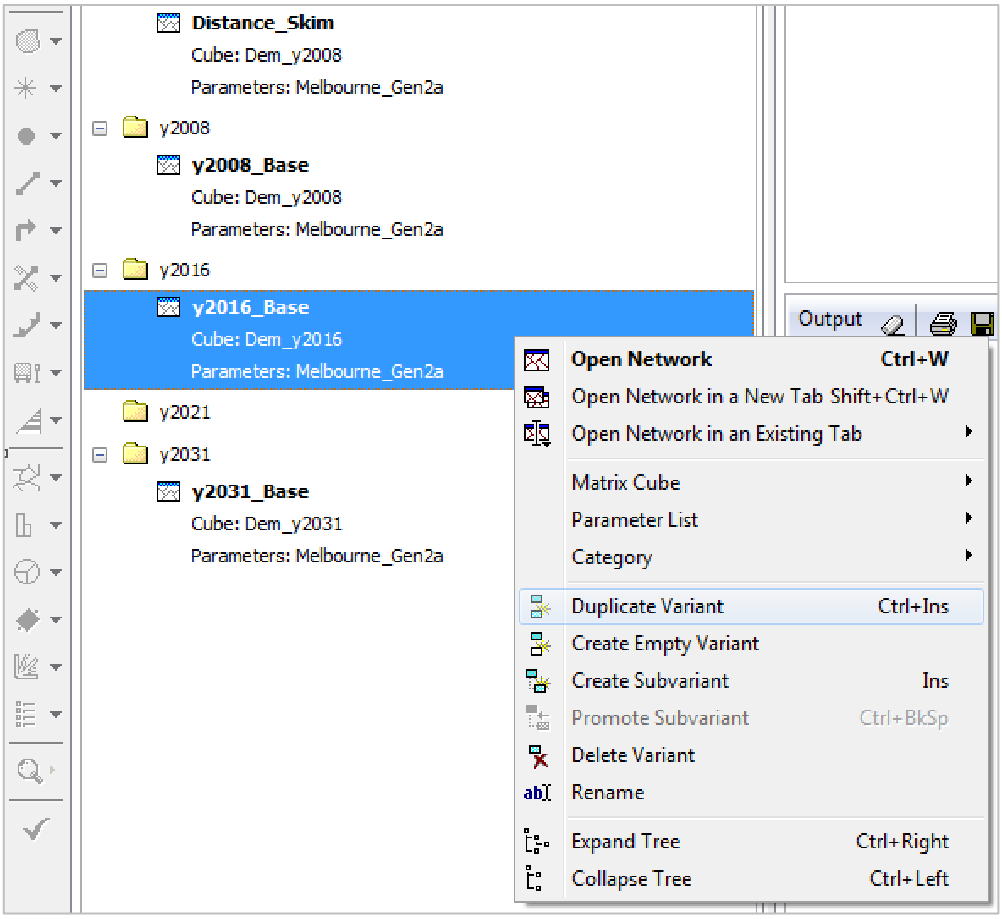
Therefore, if your scenario involves any kind of *network editing*, then you'll want to create a new <u>variant</u>. This will always mean duplicating an existing variant which provides the best starting point for your network editing. This best starting point will often be a base or future year base case variant.

If, on the other hand, your scenario involves changing something other than the network - perhaps you are changing demographics / land use or policy variables – then you will want to create a <u>subvariant</u> within the network variant that you wish to use.

NOTE: if you prefer, you can create a new variant, even if your scenario doesn't involve network editing. Just remember that this creates a redundant copy of the network data, and that you lose the logical link which says: "These runs were performed with the same network". This confidence is the true advantage of subvariants, especially when you come back to a project after an extended break!

<u>Step-by-step</u>

1. Does my scenario involve network editing?
    a. If YES – create a new variant by duplicating an existing one
    b. If NO - create a subvariant within the network variant you wish to use
2. Creating a new variant / subvariant is easy: just <u>right click</u> on the appropriate variant and bring up the menu shown below, and choose *Duplicate Variant*, or *Create Subvariant*



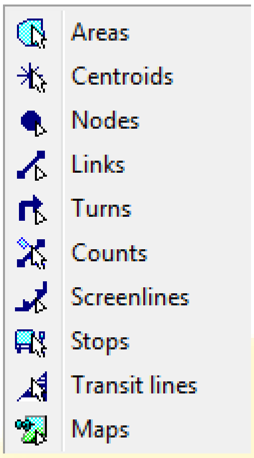
4. Editing Networks

Network editing is a modeller's bread and butter.

The starting point for any new network scenario is to duplicate an existing variant (see Section 3 for more details).

Having done this, it's time to open the variant and if possible, take a sip of your cappuccino.

OmniTRANS provides 10 distinct types of network object:



Editing a network can mean adding, deleting, or modifying one or more of these objects. OmniTRANS provides built-in tools to do this, which are easily accessible from the OmniTRANS Network Toolbar:

See the OmniTRANS Manual for more details on network editing operations *[The User Interface -> Working with Networks -> Editing Network Objects].*

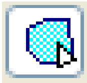
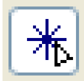
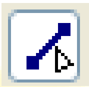
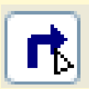
Editing a network can also mean changing the *attributes* of one or more objects – eg. increasing the capacity of a link to represent a road widening, or increasing the frequency of a transit line.

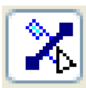
In OmniTRANS, each of the 10 object types comes with its own default set of attributes. The Zenith model does not use all of these default attributes, but does use some 'non-standard' attributes which have been added.

To avoid confusion, we've assembled a list of attributes which the Zenith model *does* use. This is found in Table 1 below.

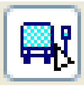
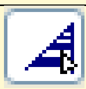
Remember that if you add a new object, you'll need to define the attributes which are in this list. As such, it's a good idea to keep a copy of the table handy.

Of course, knowing which attributes to define is only the start. The challenge is knowing what values to set them to. As such, the sub-sections which follow will detail the Zenith Network Coding Guidelines for each major object type.

Object	Icon	Required For	Name	Description of Zenith Attributes
Areas and Centroids	 	Reporting	lga	Local Government Area associated with object.
		Reporting	sla	Statistical Local Area associated with object.
		Reporting	sd	Statistical Division associated with object.
		Model Running	parkingzone	Zones which have special parking charges applied
		Model Running	walkingzone	Used to define CBD specific vehicle occupancy factors.
		Model Running	tollingzone	Indicates whether the zone is located north or south of the Yarra River. Used in the toll diversion model.
Links		Model Running	linktype	Property used to define the type of trips which can use the link[1].
		Reporting	RoadHierarchy	Hierarchical system of describing links. Used for display only.
		Model Running	speed	Defines the free flow speed along that link for any given mode during any given time period.
		Model Running	capacity	Defines the one-way hourly vehicle capacity. Definable by mode and time period, though it should really only vary by period
Turns		Model Running	impedance	This is used to ban a movement. This can be used to ban right turns, or any other illegal or impossible manoeuvre. Is definable by mode and time period.
		Model Running	toll	A toll value (in 2008 cents), which applies to vehicles who make this turn. A unique toll can be defined for cars and commercial vehicles, and by time of day. Used in the traffic assignment.
		Model Running	tollgroup	Indicates the operator of the toll point (ie. CityLink, EastLink). Used in the application of toll caps, which are defined per toll operator.

Counts		Reporting	Daily Total	Observed traffic total for a single day.
		Reporting	AM1	Observed traffic total for the 1 hour AM peak.
		Reporting	AM2	Observed traffic total for the 2 hour AM

[1] This property is used to restrict use of the link, such as a Rail only link, or a walk only link.

				peak.
		Reporting	PM1	Observed traffic total for the 1 hour PM peak.
		Reporting	PM2	Observed traffic total for the 2 hour PM peak.
		Reporting	Daily Ped	Observed pedestrian total for a single day.
Stops		Model Running	farezone	As required by a zonal fare system, each stop exists within a fare zone. Used to calculate fares in the transit assignment.
		Reporting	stn_line_group	Grouping of train stations according to the City Loop tunnels (Burnley, Clifton Hill, Caulfield, Northern Group).
		Reporting	stn_line	Grouping of train stations according to service lines and corridors (Camberwell Corridor, Epping Line, etc)
		Reporting	stn_line_segment	Train stations grouped according to services they receive in common (East Richmond-Camberwell, Riversdale-Alamein, etc)
		Model Running	stopgroup	Group used to describe which type of public transport can be accessed from that stop. All stops should have an assigned group.[2]
		Model Running	stop data	This is a link to a pop-up window for the Transit Stop Editor[3].
Transit lines		Model Running	mode	Describes the mode of the public transport service. **Do NOT define routes as the generic PT mode. Choose a specific mode (eg. tram, train, bus, etc)**
		Reporting	transittag	The real life route number or code
		Model Running	farenr	The fare system used by this transit line. In many models (including

[2] Stops which have more than one mode using the stop will only be assigned to one of those modes.
[3] Information can be found in the OmniTRANS Manual at the path: The User Interface >> Working with Networks >> Editing Network Object Attributes >> Stop Editor
*Only required for constrained public transport model run methodology

				Victoria), all transit lines will belong to a single fare system.
		Reporting	name	The route name
		Reporting	direction	This field was originally imported from Metlink data, but does not appear to have a consistent theme. We can make it consistent.
		Reporting	operator	Name of operator as described by Metlink data.
		Model Running	frequency	The number of services per hour during the time period.
		Model Running*	seats	Number of seats per vehicle on this route
		Model Running*	crushcapacity	Crush capacity per vehicle on this route.
		Model Running	speedfactor	A factor used to modify the travel speed for a specific transit line. Useful if the travel speed is a constant factor of the traffic speed.
		Model Running*	travel time	This is a link to a pop-up window for the Transit Line Data Editor[4]. **The user does not need to edit this, as it is populated automatically during a model run.**

Table 1 - Zenith Model Attributes for All Objects

[4] Information can be found in the OmniTRANS Manual at the path below:
The User Interface >> Working with Networks >> Editing Network Object Attributes >> Transit Line Editor

4.1. Coding Links

Links are used to represent all forms of network infrastructure: roads, railway lines, bus lanes, bus ways, tram lines, walk and cycling paths, and the paths taken by ferries.

The fundamental tasks of adding, deleting, copying, breaking and reshaping links are made easy by OmniTRANS, and are described thoroughly in the OmniTRANS Manual.

The art (and the focus of these guidelines), is in knowing:

- How to code link attributes (link types, speeds and capacities)
- How to code centroid connectors
- How to code the interface between the road network and the rail network
- How to code transit only lanes (ie. dedicated bus lanes, trams which have dedicated road space)

We'll start by talking in general about *link attributes*.

4.1.1. Link Attributes

In the Zenith model, there are four key attributes which must be defined for each link:

1. linktype,
2. roadhierarchy*,
3. speed,
4. capacity.

These are defined using the Attribute Editor, shown in the Figure below.

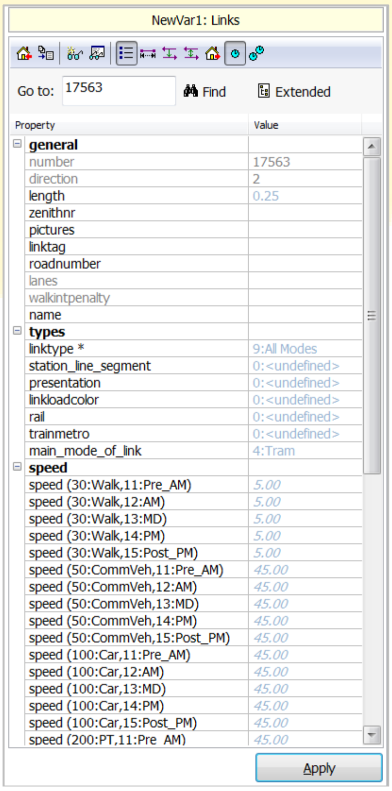


Figure 1 - Link Attribute Editor

Link attributes are separately defined for each direction of each link. To select a particular link direction, click beside the link (not on the link directly). The side to select mirrors the side of the road which cars are allowed to drive on (eg. in Australia we drive on the left).

A common rookie error is to edit one direction of the link, while forgetting about the other!

To edit both directions at once, hold shift and select both directions; then type in your attribute values (note, by holding shift you can this can extend to as many links as you like. Press ctrl to remove a link from the selection).

The OmniTRANS attribute editor also provides tools to copy data from one direction to another, or to select a link as the "default" when adding new links.

NOTE: Other standard OmniTRANS fields, such as freespeed, satflow and speedatcap, are not currently used by the Zenith model, but might be in the future if a dynamic traffic assignment model is implemented in Zenith.

The role of each attribute is now described.

4.1.1.1. Link Type

The link type plays three key roles:

a. it defines which modes are permitted to traverse the link,
b. it defines a default speed (free-flow speed) for the link,
c. it defines a speed flow curve for the link.

We now discuss each of these roles.

Mode Restrictions
The link type is used to control which modes can travel along a link.

For example, Centroid Connectors (link type 1) can be traversed by cars, commercial vehicles, pedestrians, and cyclists, but not public transport. You will never see a public transport route travel along a centroid connector!

As another example, walk only links (link type 62) allow pedestrians and cyclists, but no form of motorised transport.

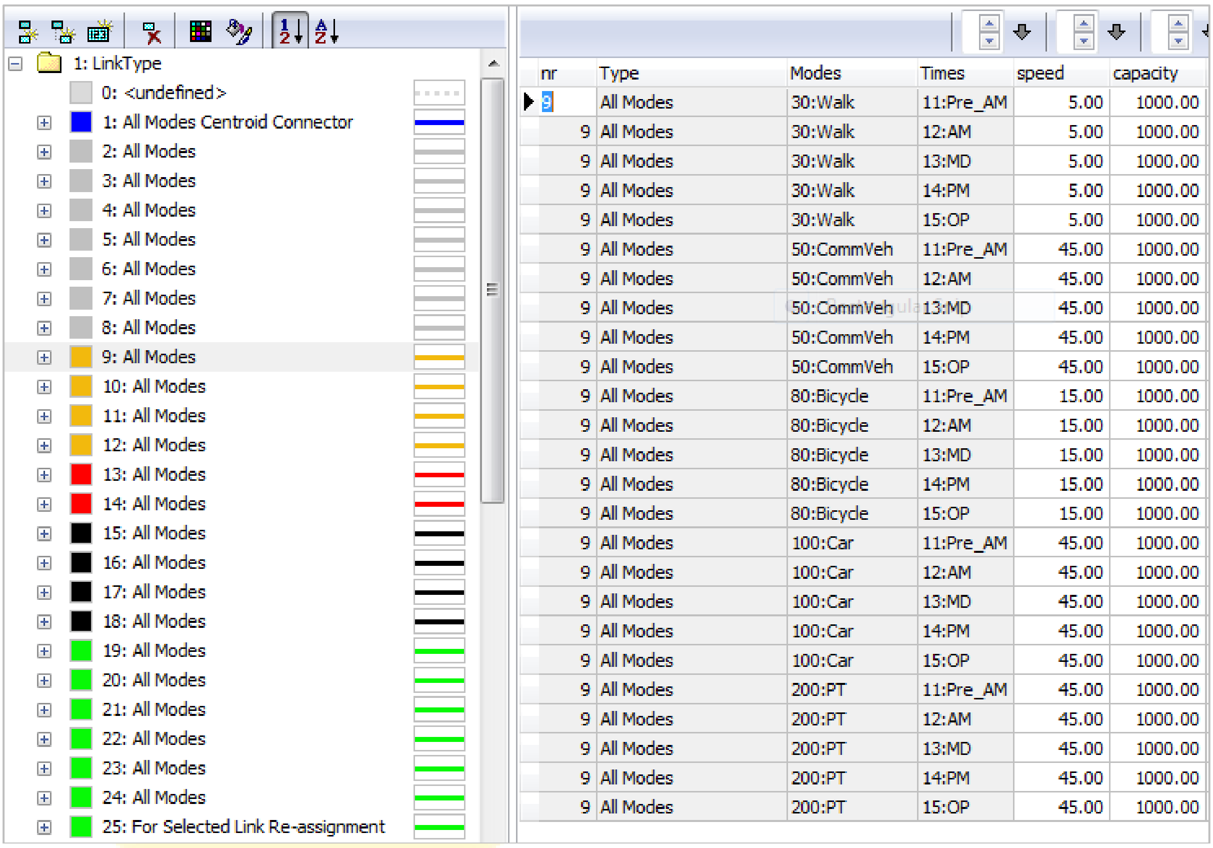
The rules which govern this are defined in the OmniTRANS Project Setup Window, under the Link Types tab.

Default Speeds
Each link type has a default speed (and capacity) for each mode and time period. The speed referred to here is the "free flow speed", rather than congested speed (see Section 4.1.1.3 for details).

For example, in Figure 2 below, we can see that link type 9 has a default speed of 45km/hr for modes Car, PT and CommVeh, 5km/hr for Walk and 15km/hr for Bicycle. Our convention is to adopt the same speed across all time periods.

So, when creating a new link, if you set the link type to 9, it will automatically populate the speed and capacity fields with their default values. Our standard practice is to not modify these speeds; rather, if you need to modify speeds, we suggest you modify the link type instead (to a link type with the desired default speed).

We do, on the other hand, alter capacities on a link by link basis (and rarely use the default capacity).



nr	Type	Modes	Times	speed	capacity
9	All Modes	30:Walk	11:Pre_AM	5.00	1000.00
9	All Modes	30:Walk	12:AM	5.00	1000.00
9	All Modes	30:Walk	13:MD	5.00	1000.00
9	All Modes	30:Walk	14:PM	5.00	1000.00
9	All Modes	30:Walk	15:OP	5.00	1000.00
9	All Modes	50:CommVeh	11:Pre_AM	45.00	1000.00
9	All Modes	50:CommVeh	12:AM	45.00	1000.00
9	All Modes	50:CommVeh	13:MD	45.00	1000.00
9	All Modes	50:CommVeh	14:PM	45.00	1000.00
9	All Modes	50:CommVeh	15:OP	45.00	1000.00
9	All Modes	80:Bicycle	11:Pre_AM	15.00	1000.00
9	All Modes	80:Bicycle	12:AM	15.00	1000.00
9	All Modes	80:Bicycle	13:MD	15.00	1000.00
9	All Modes	80:Bicycle	14:PM	15.00	1000.00
9	All Modes	80:Bicycle	15:OP	15.00	1000.00
9	All Modes	100:Car	11:Pre_AM	45.00	1000.00
9	All Modes	100:Car	12:AM	45.00	1000.00
9	All Modes	100:Car	13:MD	45.00	1000.00
9	All Modes	100:Car	14:PM	45.00	1000.00
9	All Modes	100:Car	15:OP	45.00	1000.00
9	All Modes	200:PT	11:Pre_AM	45.00	1000.00
9	All Modes	200:PT	12:AM	45.00	1000.00
9	All Modes	200:PT	13:MD	45.00	1000.00
9	All Modes	200:PT	14:PM	45.00	1000.00
9	All Modes	200:PT	15:OP	45.00	1000.00

Figure 2 - Default Speeds and Capacities for Link Type 9

Speed Flow Curves

A Speed Flow Curve controls how the speed on a link reduces as traffic load increases (due to congestion).

By default, the Zenith model only includes two speed flow curves: one for freeway roads, and one for non-freeway roads. This is defined by the ZenithModelRun parameter 'speed_flow_curves', which is usually defined in the assignment yaml file (see Section 6.1.2 for more details about yaml files).

Therefore, if you assign a "freeway" link type to a link, then you will be assigning it a freeway speed flow curve.

4.1.1.2. Road Hierarchy

The road hierarchy attribute is for cosmetic purposes only. It provides a useful way to visualise the network, but is not used in any modelling calculations.

4.1.1.3. Speed

The link speed defines the "free flow speed" for a link. Free flow speed is defined as the expected travel speed in the absence of congestion; at 3am, for example. For trafficable roads, the free flow speed defines the traffic speed when the volume of traffic is zero. As the traffic volume increases, the speed reduces based on the speed-flow curve associated with the link type.

Note that our standard practice is to not modify the speed attribute. Instead, we select a link type which has our desired speed as default. This can save you a lot of time in network editing, as changing the link type changes the speeds for all modes and capacities at once!

> Don't change the speed for a link. Instead, change the link type to one which has your desired speed as default.

The coding of link speeds is very important, as they (together with capacities) directly determine travel times as a function of demand. Being a link based model (rather than junction based) link speeds and capacities are the only determinants of travel times under congestion (apart from speed flow curves themselves).

Travel speeds can be defined for each of the networked modes: Car, CommVeh, PT, Walk, and Bicycle, and each of the networked time periods: Pre_AM, AM, MD, PM and OP.

Generally, for mixed use links (ie. roads which allow cars, commercial vehicles and public transport routes), the speed will be set equal for Car, CommVeh and PT. The Car and CommVeh speeds are used to calculate traffic speeds, and it is important that both are set correctly. The PT speed is not always used by the model; in cases where public transport routes share infrastructure with general traffic, the Zenith model will use the modelled traffic speed output from the traffic assignment to calculate the travel times of public transport routes. This means that changes in traffic congestion will impact transit travel times.

In cases of PT only links (ie. railway lines), the PT speed is used by the model to determine travel times.

The walk speed is generally set to 5km/hr for all links for which walking is permitted, while the bicycle speed is usually set to 15km/hr.

4.1.1.4. Capacity

The link capacity defines the one-way vehicle capacity, per hour.

In link based models, the capacity is typically defined as the flow rate when congested traffic speeds are half the free flow speed. In other words, for a road with free flow speed of 50km/hr, the capacity is defined as the traffic flow rate when speeds drop to 25km/hr. Under this definition, the capacity does not act as a hard upper limit on traffic flow; it is possible for traffic flow rates to exceed capacity.

While this definition is mathematically simple, it does not really help the modeller to define capacities. For the modeller, the starting point for defining capacities is the hourly vehicle saturation flow per lane.

For freeways the saturation flow is typically 2,000 vehicles per hour (vph) per lane.

For other roads it is 1,800 vehicles per hour (vph) per lane.

Having established a saturation flow rate, the modeller must then account for the effect of junctions and other impediments to traffic (such as on street parking) on the link capacity. In the real world, it is these impediments that determine road network capacity; being a link based model, we must consider these impediments when defining link speeds and capacities. The basis for doing so varies between expressways, arterials and collector roads. This will be discussed further in later sections.

Capacities can be set for each of the networked modes supported by the link type, and each of the networked time periods. The capacity of Car and CommVeh modes should be set to be equal (they are not added together. We are really just storing the same information twice. This isn't ideal, but is a limitation of the OmniTRANS data model). Capacities for the PT, Walk and Bicycle modes are never used by the model, and are generally set to 10,000.

4.1.2. Coding Centroid Connectors

Centroid connectors are "imaginary" or "abstract" links that connect travel zones to the network proper. Every travel zone must be connected to the network via at least one centroid connector.

Centroid connectors are not real roads, and so they shouldn't be subject to capacity constraints. As such, they are typically assigned an artificially high capacity of 10,000 vph.

A link type code of 1 is reserved for all centroid connectors, except walk/cycle only centroid connectors, which use a different link type (in Victoria, link type 61).

All centroid connectors are allocated a free-flow speed of 40 kilometres per hour (kph) for car travel, which remains (almost) unaltered as traffic is loaded onto the network during assignment because of their high capacity (10,000 vph). The walking speed on centroid connectors is set at 5 kph, while Bicycles can travel at 15km/hr.

In the Zenith model we generally code a single centroid connector for each zone centroid. This is, of course, a poor approximation to reality, but it is our view that adding additional centroid connectors doesn't, in general, improve accuracy. There are, of course, exceptions to this rule of thumb. Centroid connector coding can have a significant impact on local traffic flows, particularly in and around large zones. To mitigate against this, we adopt the following principles when coding travel zones and centroid connectors:

- Avoid having very large travel zones (in terms of travel activity),
- Adopt a fairly even zone size (again, in terms of travel activity) across the modelled area,
- Code the zone centroid at the centre of activity, rather than the spatial centre, and
- Connect the centroid connector to the real network at the point where most traffic is likely to enter / leave the network.

4.1.3. Coding Road Links

The coding of road networks is based largely on the road hierarchy. The Zenith model employs a six level road hierarchy:

- expressways
- major arterials
- minor arterials / sub-arterials

- major collectors
- minor collectors
- access streets

We have found that our hierarchy matches closely with the colouring of roads in Melway Maps, and as such, the Melways can serve as a good guide for coding existing and new roads.

The Table below summarises the coding of link types, speeds and capacities for each road type, together with its matching road colour in the Melways.

Road Type	Melway	Unidirectional Capacity	Capacity (Notes)	Free-flow Speed	Speed / Flow Curve	Free-flow Speed (Notes)
Expressways:						
High Standard (Interchanges > 2.0km)		1800 / lane / hr	If gradient >4% and narrow shoulders (eg major bridge, tunnel exit) use Moderate Standard (=1600/lane).	-5 to -0 kph	F/F	
Moderate Standard (Interchanges < 2.0 km)		1600 / lane / hr	If high levels of weaving involved, use Low Standard (=1400/lane)	-10 to -5 kph	F/F	factors = length of ramps, weaving levels, etc
Low Standard (Interchanges < 1.5 km)		1400 / lane / hr		-10 to -5 kph	Urban	factors = length of ramps, weaving levels, etc
1st Stage (2 lane, 2 way)		1100 / hr		-5 to -0 kph	Urban	
On-ramps (Lane Gain)		900-1400 vph	Upstream intersection dictates. Min = 900 vph, if all access movements are signalised and get low-mod green %. Max = 1400 vph, if an unsignalised LT is a major feeder movementt.	?	F/F	depends on start up speeds and ramp length
On-ramps (Merge)		900 vph	Merging capacity is dependent on the downstream traffic density in the LH lane (900 vph is an average = 50% of a F/way lane).			
Off-ramps (Lane Drop)		900-1600 vph	Downstream intersection dictates. Min (900) reflects low-mod green% or r'bout. Max (1600) reflects divergance capacity, if downstream has high capacity.	?	Urban	depends on ramp length, and probability of stopping at terminal intersection
Off-ramps (Diverge)		900 vph	Diverge capacity is dependent on the upstream traffic density in the LH lane (900 vph is an average = 50% of a F'way lane)			

Road Type	Lanes	Unidirectional Capacity	Capacity (Notes)	Free-flow Speed	Speed / Flow Curve	Free-flow Speed (Notes)
Major Arterials:						
Signal Controlled	1 lane	900 vph		-15 to -5 kph	Urban	factors = side road volumes (ie green%)
	2 lanes, undivided	1700 vph				
	2 lanes, divided	1800-2200 vph	factors include K/s parking, green-time % (aux. lane lengths, turning demands, side road			
	3 lanes	2700-3300 vph				
Roundabout Controlled	1 lane	900 vph		-10 to -5 kph		factors = side road volumes
	2 lanes	1800 vph				
Priority Controlled	1 lane	900 vph		-5 to -0 kph		factors = aux. lane standards
	2 lanes	1800-2000 vph				

Road Type		Melway	Unidirectional Capacity	Capacity (Notes)	Free-flow Speed	Speed / Flow Curve	Free-flow Speed (Notes)
Minor Arterials / Sub-arterials:							
Signal Controlled	1 lane		800 vph		-15 to -5 kph	Urban	factors = side road volumes (ie green%)
	2 lanes, undivided		1500 vph				
	2 lanes, divided		1600 vph				
	3 lanes		2400 vph				
Roundabout Controlled	1 lane		800 vph		-10 to -5 kph		factors = side road volumes
	2 lanes		1600 vph				
Priority Controlled	1 lane		800 vph		-5 to -0 kph		factors = aux. lane standards
	2 lanes		1600 vph				
Major Collector Roads:							
All Control Types	1 lane		700 vph		-15 to - 5kph	Urban	factors = priority, junction spacing, etc
	1 lane, divided		800 vph				
	2 lanes, divided		1500 vph				
Minor Collector Roads:							
All Control Types	1 lane		600 vph		-15 to - 5kph	Urban	factors = priority, junction spacing, etc
	1 lane, divided		700 vph				
	2 lanes, divided		1400 vph				
Access Streets							
Urban	1 lane		400 vph		-20 to -5 kph	Urban	factors = thresholds, priority, etc
	1 lane, divided		500 vph				
Unsealed	1 lane		100-200 vph		?		safe maximum speed
Special Situations							
Extra T2 Lanes			+25% of gen. cap				
Extra T3 Lanes			+6% of gen. cap				

Table 2 - Zenith Road Network Coding Guidelines

4.1.3.1. Coding Expressways

The coding of an expressway will depend on the type of expressway (freeway, substandard freeway, freeways with T2 lane(s), T3 lane(s), and tollways), and the number of lanes each way.

The Zenith Road Network Coding Guidelines (Table 2 above), differentiate between three standards of expressway: high, moderate and low.

The standard is related primarily to the spacing of interchanges (> 2km / interchange for high standard, between 1.5 and 2km for moderate standard, and < 1.5 for low standard), and the level of weaving expected between interchanges.

For high standard expressways, the (free flow) speed is generally set at the speed limit, or 5 kph below, while the speed of medium and low standard expressways is generally 5-10 kph below the speed limit. The length of ramps, and weaving levels should be taken into consideration when defining the speed.

The free flow speed on an expressway can be set by choosing a link type of between 51 and 60. The free flow speed is 70km/hr for link type 51, 80km/hr for link type 52, and then increases in 5km/hr increments up to a speed of 120km/hr for link type 60. Link types 51 to 60 all adopt the freeway speed flow curve.

Consult the Link Types tab of the Project Setup window for more information.

In terms of capacity, we generally assume 1800 / lane / hr for high standard expressways, dropping to 1600 and 1400 / lane / hr for medium and low standard expressways respectively.

The capacity of high standard expressways is sometimes reduced to 1600 / lane / hr in cases where the gradient is greater than 4%, and road shoulders are narrow, such as a major bridge or tunnel exit. The capacity of medium standard expressways is sometimes reduced to 1400 / lane / hr if high levels of weaving exist.

4.1.3.2. Coding Expressway Ramps

Generally speaking, there are two types of on-ramps (lane gain and merge), and two types of off ramps (lane drop and diverge).

The standard speed and capacity associated with each type is listed in Table 2 above.

The required speed can be implemented by choosing an appropriate link type. Link types 41 to 50 provide a range of speeds, ranging from 40km/hr to 100km/hr.

Freeways are generally coded as duel carriage ways, with separate one-way links for each direction, as seen in Figure 3 below. Ramps are coded explicitly; the Figure provides examples of diamond interchanges.

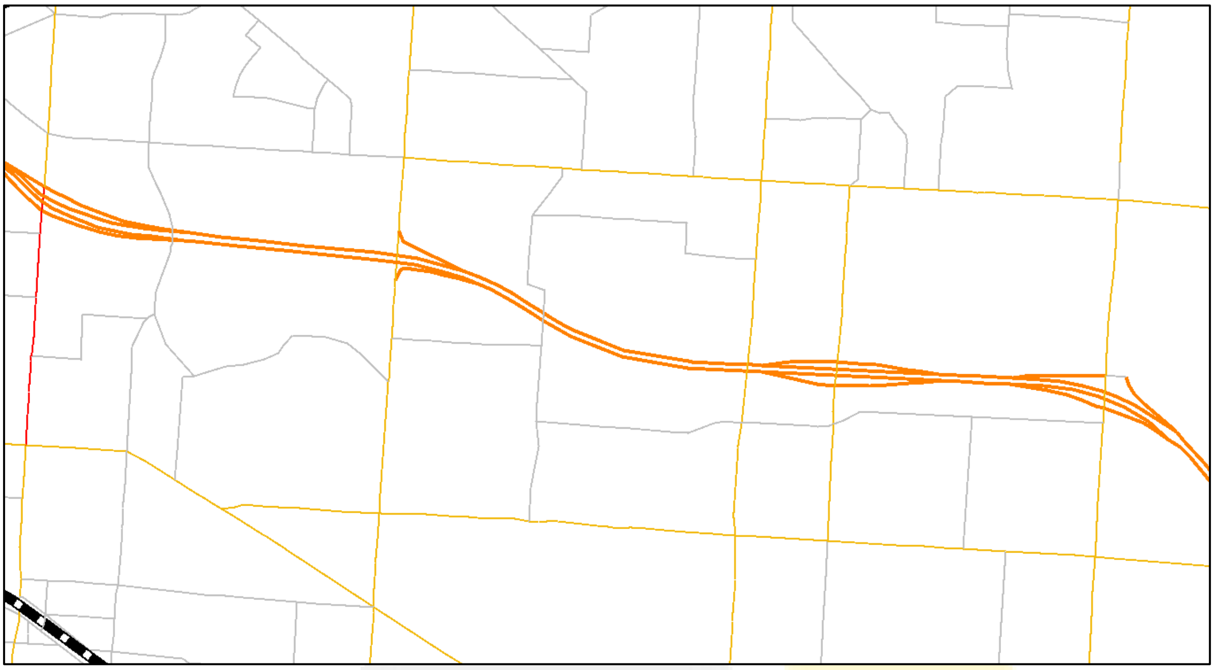


Figure 3 – Geometric Coding of Freeways and Freeway Ramps

4.1.3.3. Coding Arterials, Collectors and Access Streets

The Zenith Road Network Coding Guidelines (refer to Table 2 above) include 5 types of non-expressway road. This are listed below, in their corresponding Melway colour.

Road type
Major arterials
Minor arterials
Major collectors
Minor collectors
Access streets

The free flow speed for arterials, collectors and access streets is generally lower than the posted speed limit. This is because the speed limit is unlikely to be achieved, on average, even under uncongested conditions. Traffic signals, roundabouts, priority junctions, etc, all result in stop-start behaviour, reducing the average travel speed.

The speed can be coded between 0 and 20 kph slower than the posted speed, depending on the road type and the effects of junctions. The Zenith Road Network Coding Guidelines provide standard values and ranges for different road types, and the factors to consider when estimating speeds.

Remember, when defining a link speed, don't change the speed attribute directly; instead, choose a link type which has the target speed as its default.

Principles Underlying the Definition of Capacity on Non Expressway Roads

Defining capacities for arterials, collectors and access streets is more complex than it is for freeways. For almost all non-freeway roads it is the capacity provided on the approaches to intersections that determines their capacity, rather than their mid-block cross-section.

Table 2 provides standard values for different road types, and the factors to consider when deviating from these values.

Learning to accurately code road networks takes time, and is aided by a thorough understanding of traffic engineering principles.

This section aims to describe these principles.

In a perfect world we would know the configuration of every intersection in terms of lane allocation on each approach, the traffic signal phasing and timing if signalised, and stop and give-way priorities at un-signalised intersections - from which we could calculate a fairly accurate hourly capacity for each traffic movement. Future versions of the Zenith model might include this information through explicit coding of junctions, but until then, we must account for intersection effects in our link capacities.

For signalized intersections, the basic philosophy is to roughly estimate the amount of green time that each approach will have based on the known (or expected) traffic demand on each approach, from which the approach capacity can be estimated from the number of approach lanes. By way of example, if we have an approach to a signalized intersection that has two though traffic lanes and no flaring, and receives 25% green time at the signal then it would have a nominal capacity of 900 vph hour - i.e. 1,800 (saturation flow per lane) x 2 x 0.25. This equates to 450 vph per lane (900 vph / 2 lanes). However, if it receives more than 25% of the green time each cycle, then its capacity would be higher, and with less green time its capacity would be lower.

The reality however may be very different. If in the example above (i.e. two lane approach, no flaring with the 25% green time) left turning vehicles are being impeded by pedestrians, then the capacity of the left lane could be considerably lower than the 450 nominal capacity per hour - perhaps as low as 200-300 vph - making the capacity of the approach only 650-750 vph, instead of 900 vph.

Should flaring be introduced at an intersection then this can significantly alter the capacity equation. Typically a short right turn lane that has storage for 5 vehicles will add about 200 vph to the capacity of an approach if the cycle time is 90 seconds (i.e. 5 x 40 cycles per hour).

At the higher end of the spectrum, a larger 6 lane divided road with left and right flaring and 40 percent of the green time will have a capacity of approximately 2,560 vph (i.e. 3 x 1,800 x 0.4 + (2 x 200).

Estimating roadway capacity based on constraints on approaches to intersections, while not perfect, does however have major benefits over more traditional capacity definitions based on the mid-block configuration of a road. For example, it avoids gross over-estimation of intersection capacity when two major arterial roads carrying large volumes of traffic intersect.

Apportioning green time across the various approaches to an intersection results in a more realistic assessment of the capacity available at our heavily loaded intersections than the alternate link based method.

Approaches to single lane roundabouts are usually allocated a capacity of 900 vph, and two lane roundabouts 1,800 vph. This value can however be lowered for particular approaches if high circulating traffic demands are evident.

The capacity of lower order roads - collectors and access streets - is generally lower than that of arterials. This reflects that on average, lower priority is given to such roads at intersections. The lower capacity also leads to more sensible route choice for drivers while driving through residential neighbourhoods.

4.1.4. Coding Railways Lines

Link types 70 to 89 are reserved for railway lines. Link type 70 has a default speed of 20 km/hr, with each successive link type increasing in speed by 5 km/hr, up to a maximum of 120 km/hr for link type 89.

In the case of existing routes, speeds should be set based on the known timetables. For new infrastructure, the speed needs to be estimated.

However, it is very important to remember that dwell times are also included in the travel time of a transit line, on top of the link travel times. Dwell times need to be accounted for when estimating link travel times / speeds.

The capacity of railway lines (definable for the PT mode) is not used by the Zenith model, and is typically set to 10,000.

Railway links cannot be traversed by cars, commercial vehicles, pedestrians, or cyclists.

4.1.5. Coding Station Access Links

It is standard practice in the Zenith model to use "transit access links" to connect transit station nodes (rail stations, ferry stops, etc) to the surrounding network. Transit access links are coded as link type 63. An example is provided in the Figure below, which shows three transit access links (in red) connecting to a single train station.

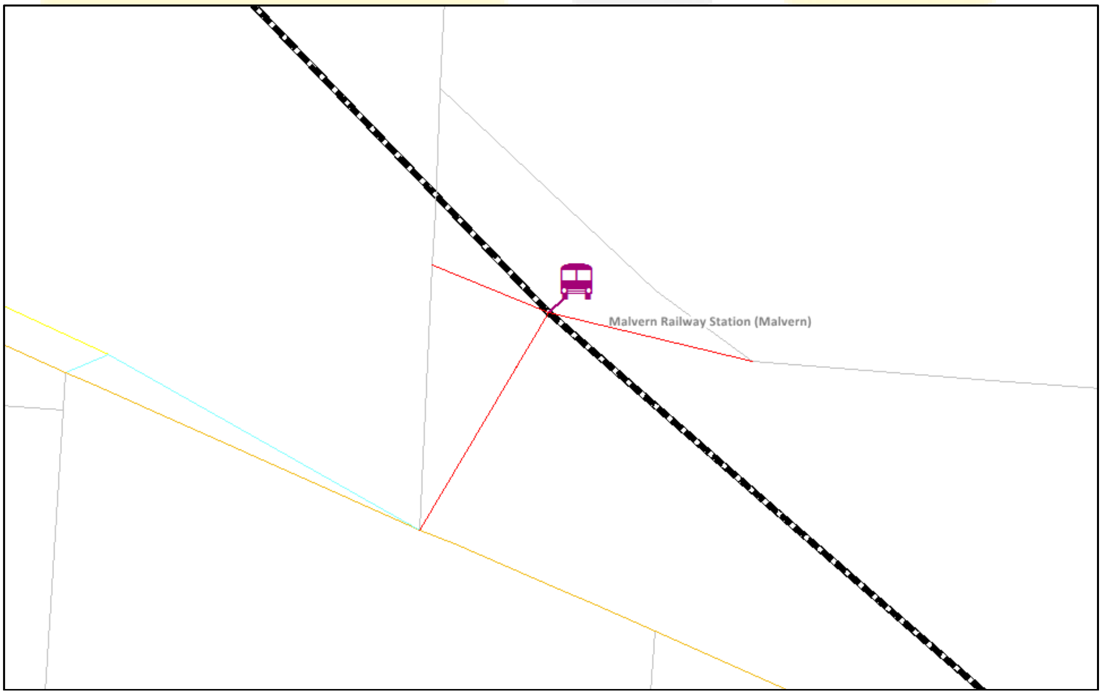


Figure 4 - Connecting Malvern Station to the Surrounding Network using Transit Access Links

Transit access links have two special characteristics, which makes them useful:

- In the traffic assignment, cars are <u>not</u> able to use these links. Otherwise, in cases where a station has connections on either side of the railway line (as above), cars would be able to drive through the train station (which they obviously can't do in real life, without some very ambitious driving!)
- In the public transport assignment, cars *<u>are</u>* able to use these links (important for park 'n' ride, and kiss 'n' ride), but are not permitted to traverse two transit access links consecutively (again preventing cars from driving through stations)

Transit access links do not have a car speed defined; but cars can use the link as part of a public transport journey, using the speed defined in the ZenithModelRun parameter 'pt_access_link_types', which is usually defined in the assignment yaml file. A car and walk speed of 5 km/hr is generally assumed.

The capacity of transit access links (definable for the walk mode) is not used by the Zenith model, and is typically set to 10,000.

4.1.6. Coding Ferry Links

Ferry links have a lot in common with railway links. Their links are functionally equivalent, though for display purposes and clarity, we choose to allocate them a different set of link types (NOTE: The Victorian Model does not currently have link types for Ferries).

4.1.7. Coding Bus Lanes and Bus Only Infrastructure

If a bus shares road space with traffic, then the bus route should be coded on the road itself. The Zenith model automatically uses congested traffic speeds from the traffic assignment to recalculate the bus travel times during the public transport assignment.

However, if the travel time of the bus is not affected by traffic (for example, if there is a dedicated bus lane), or if the bus operates on dedicated bus infrastructure (eg. busways), then separate "bus only" links should be added. Bus only links are coded with link types 110 through 129. Link type 110 has a default speed of 10 km/hr, with the speed increasing by 5 km/hr with each successive link type, up to a maximum of 105 km/hr for link type 129.

In the case of bus lanes, the bus only links should be coded parallel to the actual road. It is common to join the bus only links to the actual road at nodes where the bus stops. An example is shown in the Figure below which shows (in blue) bus lanes in bound on the Eastern Freeway, north-south on Hoddle Street, and east-west on Johnston St. Coding this way has two disadvantages, however:

- the length travelled by the bus is slightly overstated, generally by about 5%
- it is generally necessary to join two links (end on end) to connect successive stops, rather than use just one. This is because each pair of nodes can only be joined by a single link, and there is often a single link already connecting successive stops (the actual road)

Despite these disadvantages, you will find many cases in the Zenith model where bus / tram lanes are coded in this way. An example is provided in Figure 5 below.

Figure 5 - Bus Lanes (in blue) on the Eastern Freeway, Hoddle Street and Johnston Street

There is, however, a better way, which we are gradually moving towards.

Our new practice is to separate the bus stops from the actual road network, with transit access links used to connect each stop to the actual road network. An example is shown in the Figure below.

Figure 6 - A better approach to coding bus lanes (in blue) on Hoddle and Johnston Streets

The coding of busways is comparatively simpler; they can be coded much the same way as railway lines, with care given to providing appropriate access to each stop.

4.1.8. Coding of Tram Infrastructure

As with busses, trams can run on-road, sharing lane space with cars. They can also have dedicated lane space, separating them from traffic flows (as they do in the CBD).

In the first case (sharing lane space), it is appropriate to code transit lines along normal road network links. The travel time of such tram routes will be equal to the congested traffic speed during a model run, plus dwell time.

In the second case (dedicated tram lanes), the standard practice is to code separate tram only links, which separate them from the traffic stream. The coding is identical to the coding of bus lanes described in the previous section. Tram only links are assigned link types of 90 to 95, depending on the speed.

4.1.9. Checking Link Coding

There are two main ways to check the coding of network links:

1. Producing and inspecting plots of network attributes
2. Using the OmniTRANS Shortest Path Engine to check network connectivity

Each will now be described.

Checking Link Types

To check link types, we recommend that you colour links by link type. To do this, select LinkType as the Colouring option in the Link Settings, as shown in Figure 7 below. However, this won't differentiate, for example, between link types 9,10,11 and 12, which all share the same colour.

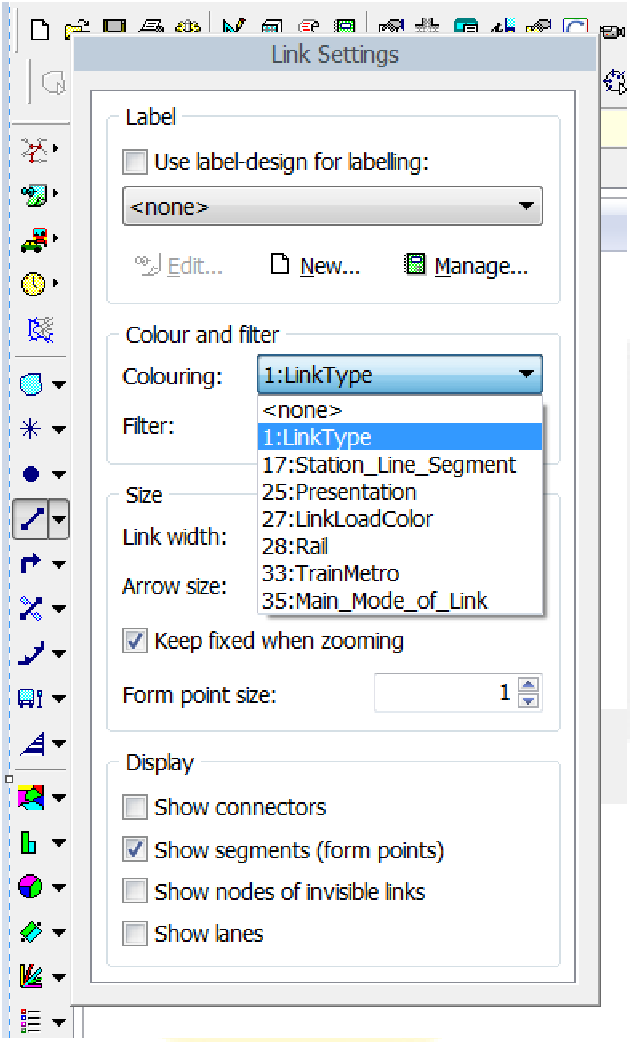


Figure 7 - Colouring Links By Link Type

A further check can be performed by labelling links with their link type, as demonstrated in Figure 8 below.

The combined plot is shown in Figure 9 below.

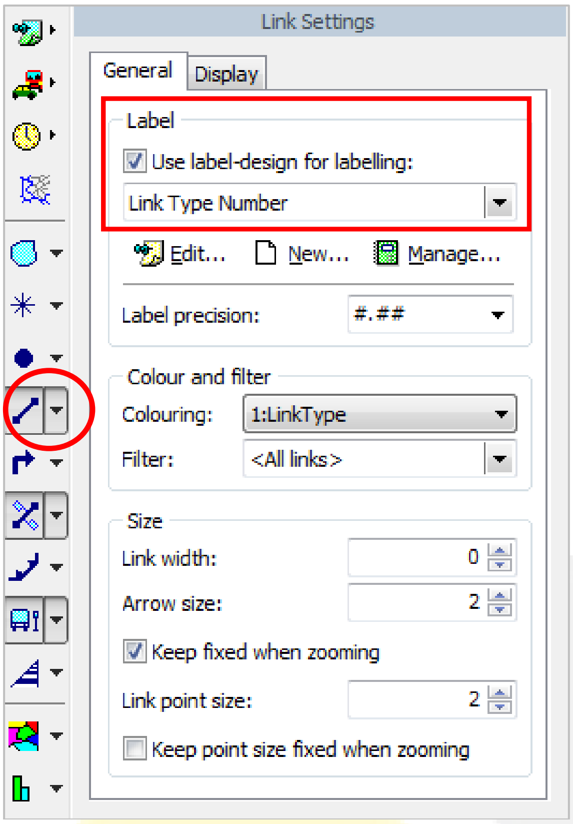


Figure 8 - Labelling Links with their Link Type Number

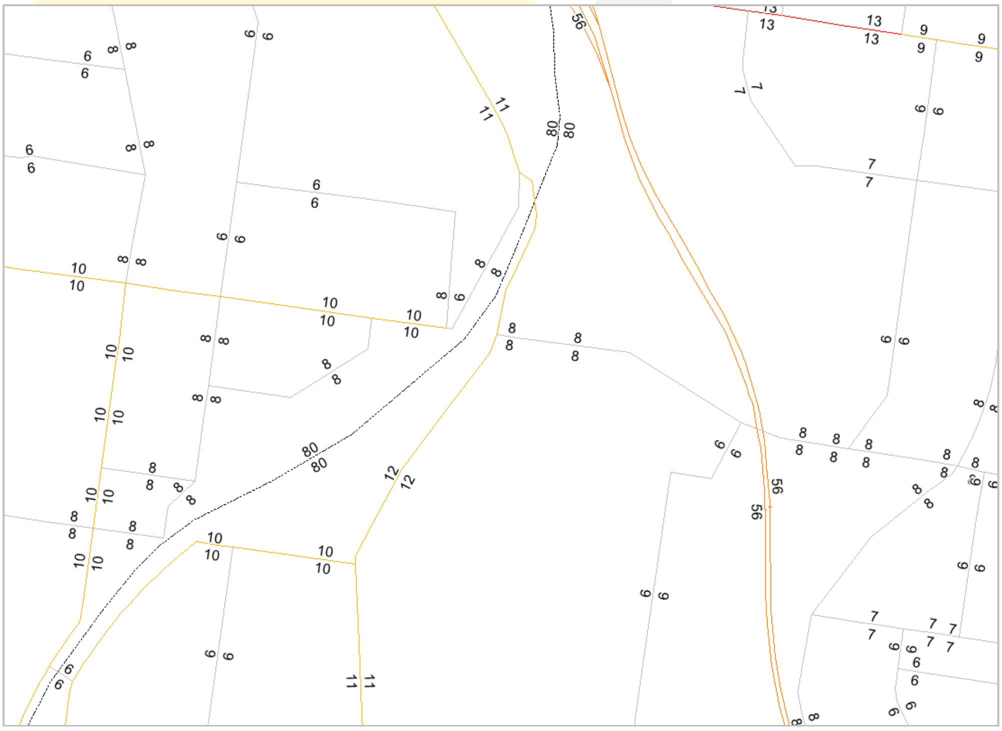


Figure 9 - A Plot of Link Types

Checking Speeds and Capacities

To check speeds and capacities, we have setup the following bandwidth designs:

Speed bandwidth designs:

- (NR) Review - Car FFS speeds (7-9am) - displays the car free flow speed for a single time period.
- (NR) Review - PT FFS speeds (7-9am) displays the public transport speed across a single time period.

Capacity bandwidth designs:

- (NR) Car - Capacity (7-9am) - displays the car capacity across a single time period.

Using the OmniTRANS Shortest Path Engine

The OmniTRANS Shortest Path Engine is a tool which can calculate and display the shortest path between any user defined pair of zone centroids.

The user is able to select an origin, select a destination, and the tool will plot the path. This is very useful in checking that sensible paths are found after building new infrastructure.

For instructions on using the Shortest Path Engine, look in the OmniTRANS Manual *[The User Interface -> Working With Networks -> Shortest Path -> Shortest Path Display]*

4.2. Coding Public Transport Routes and Stops

The public transport network is represented by two OmniTRANS network objects: transit lines and stops.

4.2.1. Coding Transit Lines

Public transport routes are modelled by Transit Line objects. It is standard practice in Zenith models to code a separate transit line for each "route variation". A route variation may take a different path (eg. a deviation), or employ a modified stopping pattern.

Transit lines objects can be added, deleted, rerouted, extended and shortened by using standard OmniTRANS tools *[The User Interface -> Working with Networks -> Editing Network Objects]*

4.2.1.1. Transit Line Attributes

Besides their geometry, transit lines have several key attributes which must be set by the user:

- mode - *the transit mode (bus, rail, tram, etc)*
- transittag – *the real life route number or code*
- farenr - *specifies the fare system to be used in calculating fares*
- name – *the name of the route*
- ptdirection – *inbound, outbound or N/A*
- operator – *the operator of the route*
- frequency – *services per hour (definable for each time period)*
- seats - *seats on board the transit vehicle; only used when crowding model is employed*
- crushcapacity - *the crush capacity of the transit vehicle; only used when crowding model is employed*

- speedfactor - *a factor which adjusts the speed of the route; used when the speed should be a factor of the congested link traffic speeds*

Each of these attributes can be viewed and edited using the Attribute Editor, seen in Figure 10 below. It is important that the mode be set to an actual public transport mode, and not just the default mode, PT. This is because the PT mode does not have any attributes (eg. boarding penalties, in vehicle time weights) associated with it.

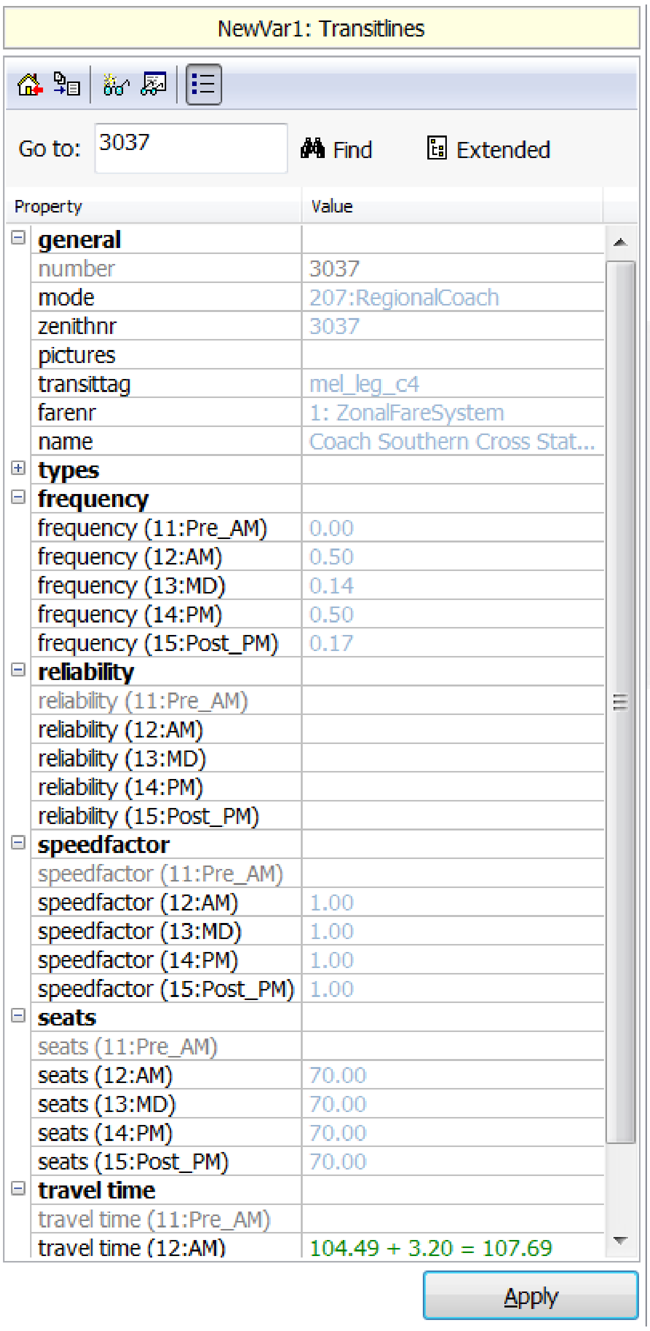


Figure 10 - The Transit Line Attribute Editor

4.2.1.2. *Defining Frequencies*

Defining the frequency of new transit lines is easy; you just need to remember that the frequency is a frequency per hour. However, calculating the frequency of a transit line from a real life timetable is more complex, as individual runs of a service can span more than one time period. For example, a run could begin at 6:30am (in the Pre_AM period), and reach its terminus at 7:30am (in the AM Peak). In these cases, it is standard practice in Zenith models to assign runs to time periods based on the moment when they are closest the CBD. This principal is generally applied by taking the end time of the service for inbound routes, and the start time of the service for outbound routes. For routes which are neither inbound or outbound, the average of start time and end time is often used, though the start or end time would do just as good a job.

Another equally valid approach is to allocate a portion of the frequency to each time period, based on the service duration in each period. For example, for a service beginning at 6:45am, and ending at 7:45am would add 0.25 of frequency to the Pre_AM period and 0.75 of frequency to the AM (peak) period. This approach can be simpler to apply.

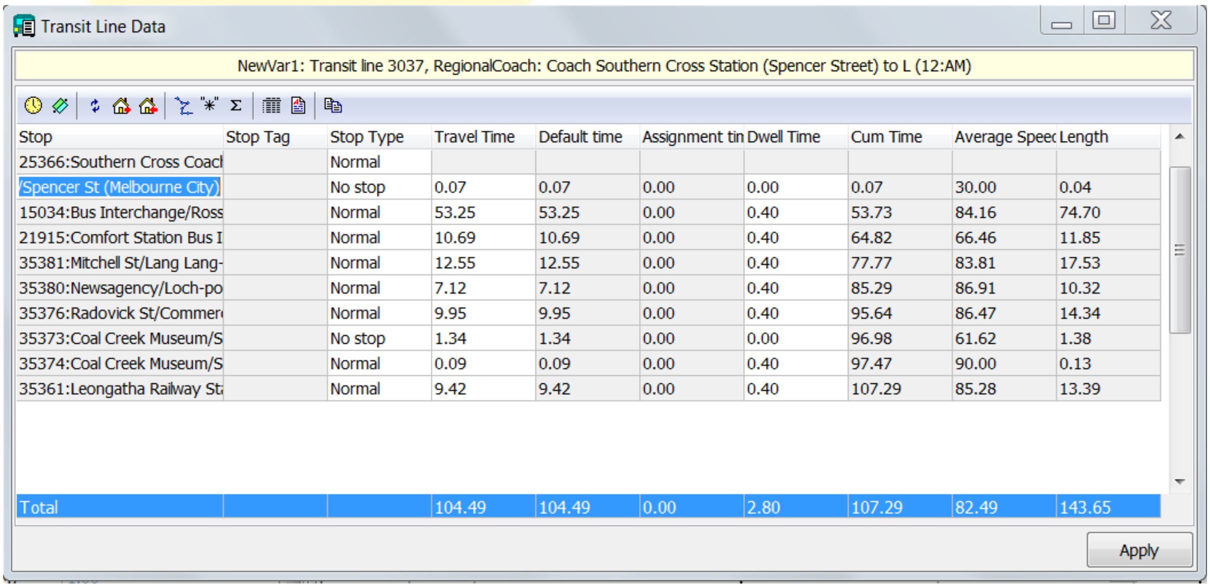
There is no "best" way to do this, except to ensure that the network coding reflects what you're trying to achieve.

4.2.1.3. *Associating Transit Lines with Fare Systems*

The farenr can be selected from the available fare systems, using a drop down menu. In Zenith models there is generally just one fare system, though there is no rule against having more than one fare system. Failure to set the farenr will result in no fare applying to the transit line.

4.2.1.4. *Transit Line Travel Times*

For each transit line, OmniTRANS also stores travel times for each link traversed by the route. These can be accessed via the travel time attribute of the Attribute Editor, seen in the Figure below.


Transit Line Data

NewVar1: Transit line 3037, RegionalCoach: Coach Southern Cross Station (Spencer Street) to L (12:AM)

Stop	Stop Tag	Stop Type	Travel Time	Default time	Assignment tin	Dwell Time	Cum Time	Average Speec	Length
25366:Southern Cross Coacl		Normal							
Spencer St (Melbourne City)		No stop	0.07	0.07	0.00	0.00	0.07	30.00	0.04
15034:Bus Interchange/Ross		Normal	53.25	53.25	0.00	0.40	53.73	84.16	74.70
21915:Comfort Station Bus I		Normal	10.69	10.69	0.00	0.40	64.82	66.46	11.85
35381:Mitchell St/Lang Lang-		Normal	12.55	12.55	0.00	0.40	77.77	83.81	17.53
35380:Newsagency/Loch-po		Normal	7.12	7.12	0.00	0.40	85.29	86.91	10.32
35376:Radovick St/Commer		Normal	9.95	9.95	0.00	0.40	95.64	86.47	14.34
35373:Coal Creek Museum/S		No stop	1.34	1.34	0.00	0.00	96.98	61.62	1.38
35374:Coal Creek Museum/S		Normal	0.09	0.09	0.00	0.40	97.47	90.00	0.13
35361:Leongatha Railway St		Normal	9.42	9.42	0.00	0.40	107.29	85.28	13.39
Total			104.49	104.49	0.00	2.80	107.29	82.49	143.65

Apply

Figure 11 – Transit line data window

<u>However, it is not necessary for the modeller to set these travel times.</u> They are automatically set during a Zenith model run, based on the underlying link travel speeds for the PT mode, and the relevant time period. This means that any edits made to the transit line travel times will be overwritten during the run.

> Don't bother edit the 'timetable' for a transit line – it is a waste of time as your edits will be overwritten when the model runs!

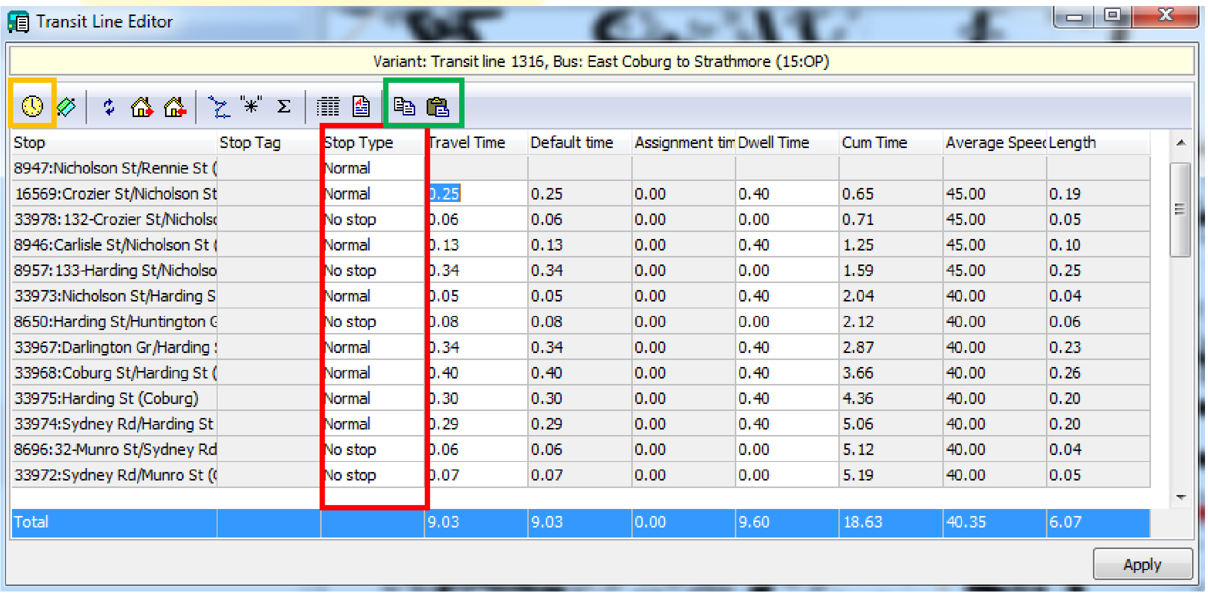
4.2.1.5. *Transit Line Stopping Patterns (eg. express services)*

The stopping pattern for a transit line can be controlled by editing the Stop Type, as shown in the Red rectangle in Figure 12 below. The following Stop Types are supported:

- Normal – boarding and alighting are allowed
- No transfers – boarding and alighting are permitted but transfers cannot take place between routes at the stop
- No stop – no boarding or alighting allowed. Used to represent express stops
- Board only / Alight only – pretty self explanatory!

NOTE, however, that these attributes must be modified for each time period separately. The active time period can be set using the button highlighted in the orange rectangle, and the copy / paste buttons (in the green rectangle) can be used to copy data from one period to another.

> Don't forget to change the stop type in every period!



Transit Line Editor

Variant: Transit line 1316, Bus: East Coburg to Strathmore (15:OP)

Stop	Stop Tag	Stop Type	Travel Time	Default time	Assignment tim	Dwell Time	Cum Time	Average Speed	Length
8947:Nicholson St/Rennie St (		Normal							
16569:Crozier St/Nicholson St		Normal	0.25	0.25	0.00	0.40	0.65	45.00	0.19
33978:132-Crozier St/Nicholsc		No stop	0.06	0.06	0.00	0.00	0.71	45.00	0.05
8946:Carlisle St/Nicholson St (		Normal	0.13	0.13	0.00	0.40	1.25	45.00	0.10
8957:133-Harding St/Nicholso		No stop	0.34	0.34	0.00	0.00	1.59	45.00	0.25
33973:Nicholson St/Harding S		Normal	0.05	0.05	0.00	0.40	2.04	40.00	0.04
8650:Harding St/Huntington G		No stop	0.08	0.08	0.00	0.00	2.12	40.00	0.06
33967:Darlington Gr/Harding S		Normal	0.34	0.34	0.00	0.40	2.87	40.00	0.23
33968:Coburg St/Harding St (		Normal	0.40	0.40	0.00	0.40	3.66	40.00	0.26
33975:Harding St (Coburg)		Normal	0.30	0.30	0.00	0.40	4.36	40.00	0.20
33974:Sydney Rd/Harding St		Normal	0.29	0.29	0.00	0.40	5.06	40.00	0.20
8696:32-Munro St/Sydney Rd		No stop	0.06	0.06	0.00	0.00	5.12	40.00	0.04
33972:Sydney Rd/Munro St ((		No stop	0.07	0.07	0.00	0.00	5.19	40.00	0.05
Total			9.03	9.03	0.00	9.60	18.63	40.35	6.07

Apply

Figure 12 - Editing Stop Types using the Transit Line Editor

4.2.1.6. Dwell times

As you can see in Figure 12 above, the dwell time can be separately defined for each stop along a route (and by time periods!). While this is useful, it is also tedious. As such, we generally define default values (per mode), using the parameter 'dwell_times_override' on ZenithModelRun. This parameter is usually found in the assignment yaml file.

If the dwell_times_override parameter is defined, then any edits you make to dwell times will be overridden when you run the model. If you wish to have more control, you can simply remove the 'dwell_times_override' parameter from the yaml, or remove the default value for particular mode(s).

> Don't bother defining dwell times. The default values specified in the assignment yaml will override any changes you make!

4.2.2. Coding Transit Stops

Stops provide a way for passengers to board and alight transit lines.

OmniTRANS provides tools for adding, deleting and moving stops *[The User Interface -> Working with Networks -> Editing Network Objects].*

4.2.2.1. Stop Attributes

When adding stops, it is important define the following attributes which are used by the Zenith model:

- farezone – *defines which fare zone the stop is part of. This is used in the calculation of fares!*
- stn_line_group (rail stops only) – *defines which line group a rail station is part of*
- stn_line (rail stops only) – *defines which line a rail station is part of*
- stn_line_segment (rail stops only) – *defines which line segment a rail station is part of*
- stopgroup – *defined as rail, bus or tram. Influences route choice decisions*

Besides these attributes, it is important that you set appropriate stop types for all the transit lines passing through the stop.

The stop type is editable on a per transit line basis using the Transit Line Editor (refer to Figure 12 above).

The stop type is also editable on a per stop basis by opening the Stop Editor (icon on the toolbar). The Stop Editor (shown in Figure 13 below) allows you to edit the stop type for all transit lines passing through a stop.

NOTE that you need to set the stop type for each period separately. To switch between periods, click on within the Stop Editor. You can use the copy / paste tools to copy between periods.

There is no need to modify the dwell time as the Zenith model assumes default dwell times per mode, which will overwrite any edits you make here.

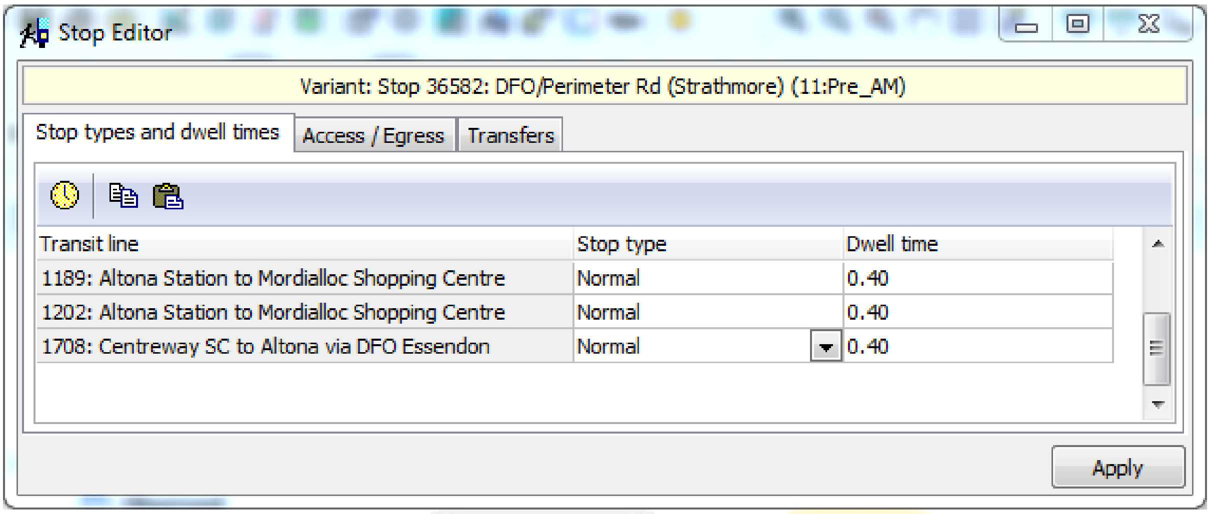


Figure 13 - The Stop Editor

4.2.2.2. Access / Egress Penalties and Waiting Times

Access and egress penalties and waiting times are calculated and applied at the first point of boarding the transit system, and the last point of egress.

Both the penalty and the waiting time can be calculated as a user defined factor of the headway (based on the combined frequency of feasible services), or as a user defined constant.

Default access and egress penalties and waiting times are defined in the Transit Transfers tab of the Project Setup Window. They are separately defined for each access mode and transit mode pair (for example, walk access / egress to train, or car access / egress to bus, etc).

Penalties and waiting times can be overridden for individual stops, by using the Access / Egress tab of the Stop Editor window, shown in the Figure below. First select the stop, then press  on the toolbar.

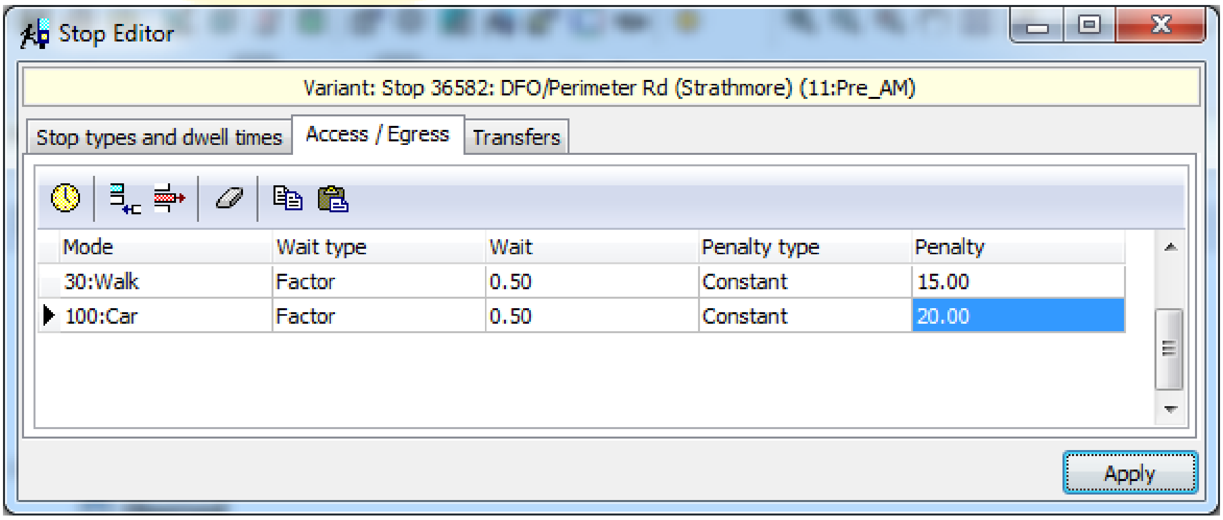


Figure 14 -Setting Access / Egress Penalties and Waiting Times

Walk and car are the only transit access modes modelled in Zenith, and so it only makes sense to choose these as an access mode.

The standard practice in Zenith models is to calculate waiting time as a factor of the headway. A factor of 0.5 is generally assumed (ie. half the headway). Conversely, the penalty is generally set to be a constant (in minutes). The penalty is applied on both access (first boarding) and egress (last alighting), whereas the waiting time is only applied on access.

The penalties are generally used as a catch all to reflect factors not incorporated into the model: safety, amenity, car parking, drop off points, etc. We hope over time to include more of these variables into the model, thus reducing the need for these penalties.

For new stations, it is generally wise to review the penalties of nearby stops, and to use an average of such penalties, or simply adopt the default values (by not specifying anything in this window).

4.2.2.3. Transfer Penalties and Waiting Times

Transfer penalties and waiting times are calculated and applied whenever a passenger transfers from one transit line to another. Transfers can take place at a single stop, or by walking between stops. This section relates to transfers at a single stop; the next section deals with walking between stops.

As with access / egress penalties and waiting times, there are global values defined for transfers between pairs of transit modes. They can be viewed and edited in the Transit Transfers tab of the Project Setup Window.

Override transfer penalties and waiting times can be defined for transfers between specific transit line pairs at a stop, or as stop global values which apply to all pairs of transit lines at the stop. This facilitates the modelling of timetable effects, and the coordination of services, within a frequency based algorithm.

Such overrides can be viewed and edited by opening the Transfers tab of the Transit Stop Data window. First select the stop, then press  on the toolbar.

Two examples are shown below.

In Figure 15, a waiting time of 3 minutes, and a penalty of 15 minutes is defined for the transfer from transit line 1189 to transit line 1202. Note that the "Wait type" is set to "Constant", indicating that the value of 3 is a fixed waiting time.

In Figure 16, the transfer penalty for the stop is set to 5 minutes (between all pairs of transit lines).

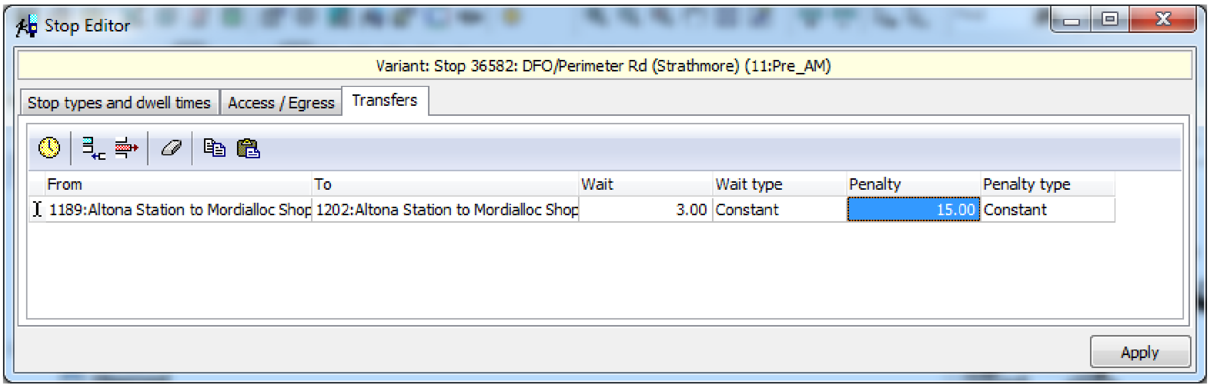


Figure 15 – Coordinated services using the transfers tab of the Stop Editor

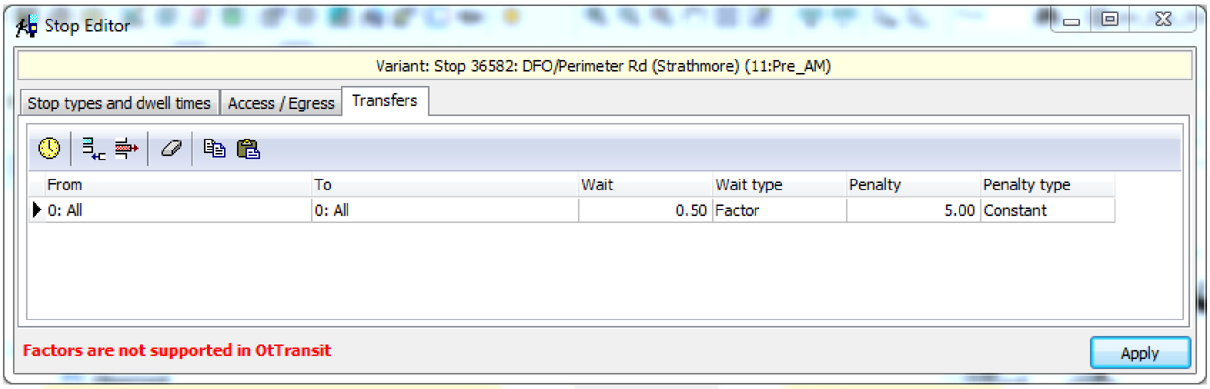


Figure 16 - Overriding the transfer penalty for a single stop

4.3. Coding Tolls

Tolls are stored as an attribute of Turn objects.

Each Turn object connects three nodes, joined by two Link objects. OmniTRANS provides tools for adding and deleting turns.

Because they are stored on a turn, tolls are only incurred by vehicles that traverse the turn. This gives greater flexibility in defining tolls than would be possible by defining tolls on links.

Turns have two attributes which are important when coding tolls:

- toll - *the toll, in 2008 cents, is definable for all networked modes, though it only needs to be specified for Car and CommVeh. It is also definable for all networked time periods, which allows the specification of period specific tolls*
- tollgroups - *assigns the toll to a particular group or operator. Is used to apply toll caps, which are definable for each group. New groups can be created by going to the Types tab of the Project Setup Window. Toll caps can be defined using the 'toll_caps' property on ZenithModelRun. Default values are usually defined in the assignment yaml file.*

When converting current tolls back to 2008 cents, we generally just use changes in Consumer Price Index (CPI) as a measure of inflation. The toll in 2008 dollars should always be lower than the current toll.

4.4. Coding Banned Turns and Turn Delays

It will come as no surprise that banned turns and turn delays are modelled using Turn objects!

Each Turn object connects three nodes, joined by two Link objects. OmniTRANS provides tools for adding and deleting turns. One trick to remember is that a turn object often already exists, even if it is not being displayed. Before trying to add a turn object, make sure that the Show Turns Without Manual Data option is switched on, as highlighted in red in the Figure below.

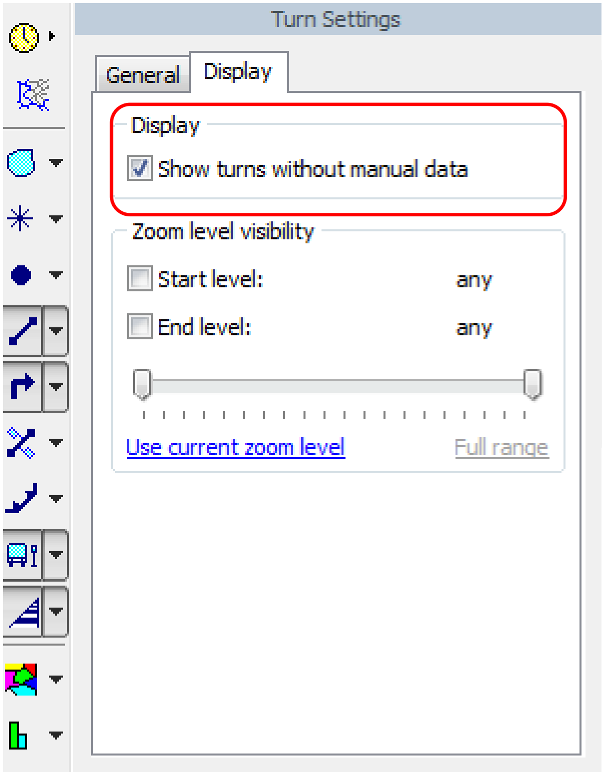


Figure 17 - The Show Turns Without Manual Data option

To ban a turn, simply select an existing turn (or create a new one), then open the attribute editor. Set the impedance value to "Banned" for Car and/or Commercial Vehicles, in the time periods where the turn ban should apply. An example is shown in Figure 18 below.

To add a delay instead, simply change the value banned to a numeric value. This value represents a fixed turn delay (in seconds).

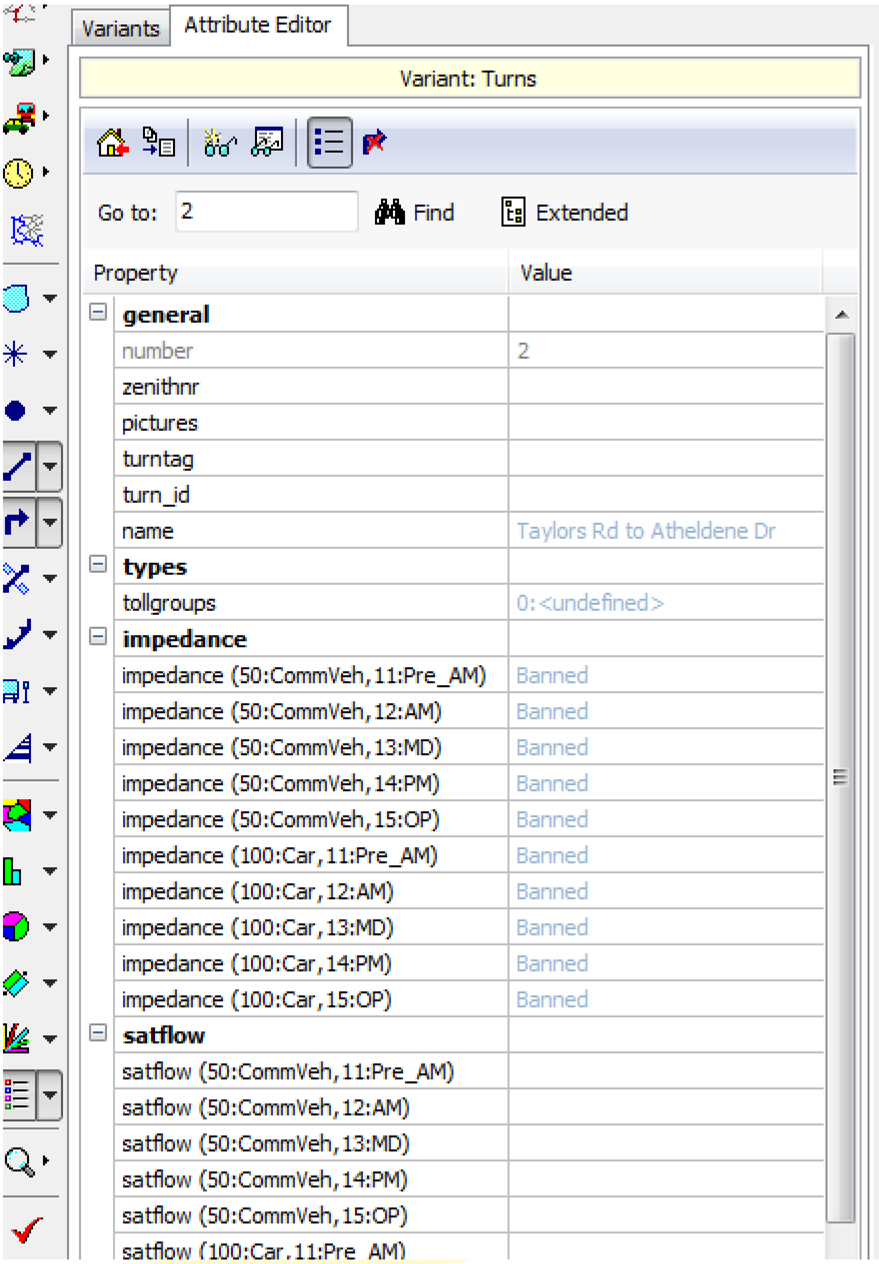


Figure 18 - The Attributes of a Banned Turn

4.5. Coding Travel Zones

Your model will already have a built in travel zone system – usually with several thousand travel zones.

For most model users, there is no need to modify the travel zone system, and we generally recommend that if you do need to change travel zones, that you contact user support. However, for the advanced user, here are some instructions on how to modify your travel zone system.

Modifying the Travel Zone System

One of limitations of OmniTRANS is that each OmniTRANS project is only capable of supporting a single travel zone system.

If you wish to aggregate or disaggregate the travel zone system in any particular area, you need to create a new OmniTRANS project; the new project can then store each of the variants built with the new travel zone system. We also strongly recommend this approach if you wish to spatially relocate zone centroid objects (generally it's just simpler not to!)

The process is not for the light hearted, and we recommend that you seek guidance before trying it. Nonetheless, this is how you can do it:

1. Export the network to MapInfo format, using the MapInfo network exporter (under the Zenith -> Export menu),
2. Make all necessary changes to the travel zone system using MapInfo, or another GIS tool which can make use of MapInfo files. This may include modifications to zone centroids, as well as the links which connect them to the network. Pay particular care to the centroid attributes: ParkingZone, WalkingZone, SLA, LGA, etc. You will want to check the connectivity of the network using the MapInfo Optimiser tool (contact technical support to obtain a copy of this tool).
3. Create a new OmniTRANS project based off your existing one. The easiest way to do this is to:
    i. Open Windows Explorer and navigate to the project folder
    ii. Select one variant only, plus the following folders: data, jobs, external user data, maps, matrix, reports, plus all the files within the project folder (.db, .px, among others)
    iii. Copy the selected data into a new folder. The name of the folder should be the name of the new project.
4. Open the Project
5. Right click on the single variant, and create an empty variant.
6. Delete the initial variant
7. Go to the menu: Project -> Database -> Check and Repair, and then select:
    i. Remove (Data from) Unknown Objects
    ii. Remove Data from Unknown Categories
    iii. Remove Unused Super Objects
8. The OmniTRANS project now has no network data!
9. Select the new (empty) variant, and import the modified MapInfo network (go to Zenith -> Import -> Import MapInfo Network).
10. Create a series of new matrix cubes to store demographic and land use data at the new zone system

Good luck!

4.6. Editing Networks Using GIS Tools

The network editing capabilities built into OmniTRANS are very powerful, and will handle day-to-day coding of scenarios. However, on occasion, it can be advantageous to edit networks in MapInfo or using another GIS tool, rather than OmniTRANS.

One example is the modification of the travel zone system, as outlined in the previous section. However, there are other situations where MapInfo is useful, such as merging the road network from one variant with the public transport network from another variant.

One of the limiting features of OmniTRANS (in terms of network editing) is that it always maintains the network in a topologically valid state. By topologically valid, we mean that:

- every link must begin and end at a node or centroid object
- every stop object must be situated on a node
- every transit line must traverse a set of connected links
- a turn object must connect three nodes, traversed by two links

In OmniTRANS you will never see a network that fails these criteria. None of the network editing tools in OmniTRANS will allow you to break these rules.

In most this cases, this is a positive; for example, if you move a node, it will automatically move all the attached links, turns, and transit line objects, and the associated stop (if there is one). This saves you time!!

However, for certain very complex network edits, or in cases where you are trying to merge inconsistent data sets (ie. combine a public transport network with an inconsistent road network), it is necessary for the network to be *temporarily* invalid, while the user makes changes to make the network valid again. This is possible in a GIS package like MapInfo.

To facilitate this, Zenith comes with MapInfo import / export functionality. See the section on Import / Export for more information [link].

The export produces a series of MapInfo layers, that together represent the full set of network objects: links, nodes, centroids, stops, transit lines, areas, etc. After editing these layers, and before attempting to re-import, it is sensible to check the topological validity of the MapInfo network. This can be done using a MapInfo tool developed by VLC called the "network optimiser". After "optimising", the network can be re-imported into an empty variant in OmniTRANS.

The Network Optimiser

The MapInfo network optimiser tool VLC to ensure that any networks produced in MapInfo are fit to be imported into OmniTRANS and then used in a model run.

The Optimisation process is split into two distinct steps; Infrastructure and PT. The Infrastructure optimisation ensures that the centroids, nodes, centroid connectors, links and turns are consistent. The PT optimiser ensures that the stops, routes, route stops, access penalties and transfer penalties are consistent with the already optimised links and nodes.

4.6.1. Infrastructure Optimiser

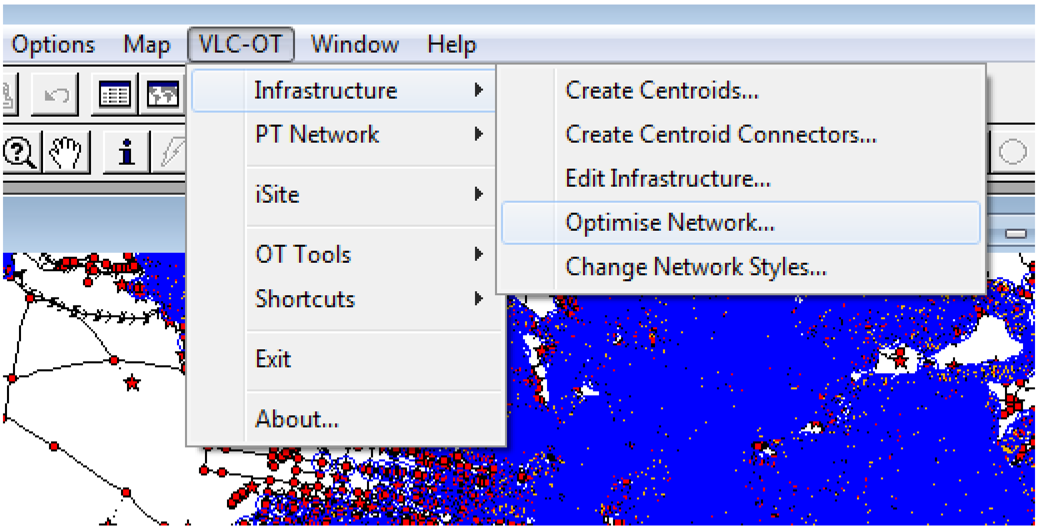


Figure 19 – Accessing the Infrastructure Optimiser

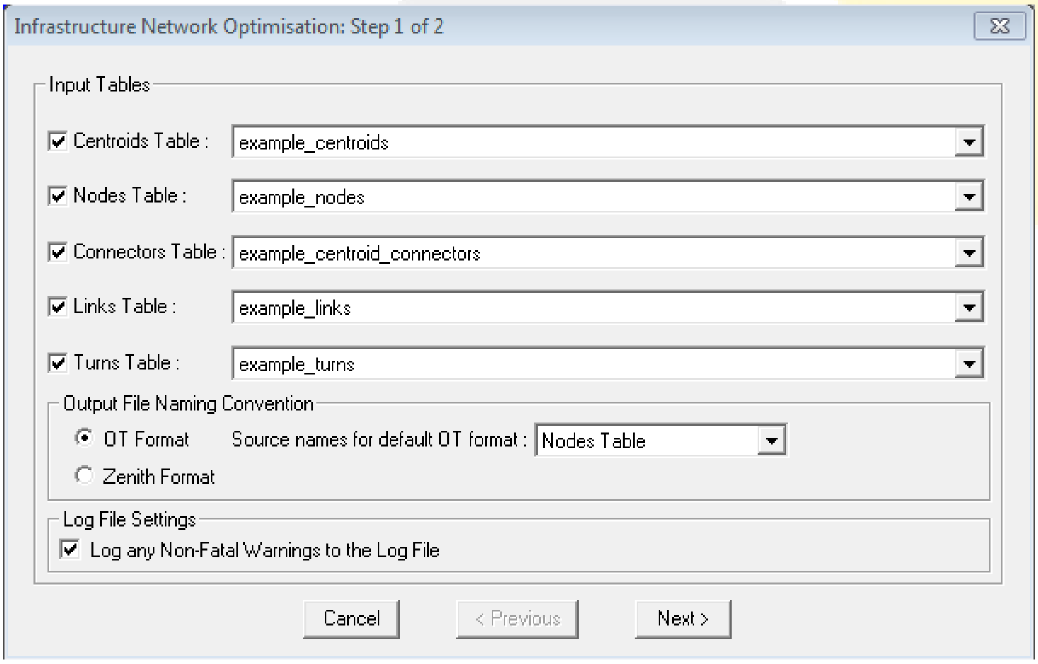


Figure 20 – Inputs to the Infrastructure Optimiser

The Optimiser should automatically detect the appropriate tables for each element of the process. It's worth having a quick check to make sure that it has got everything correct. You can also select (if you have differing naming for your tables) which table

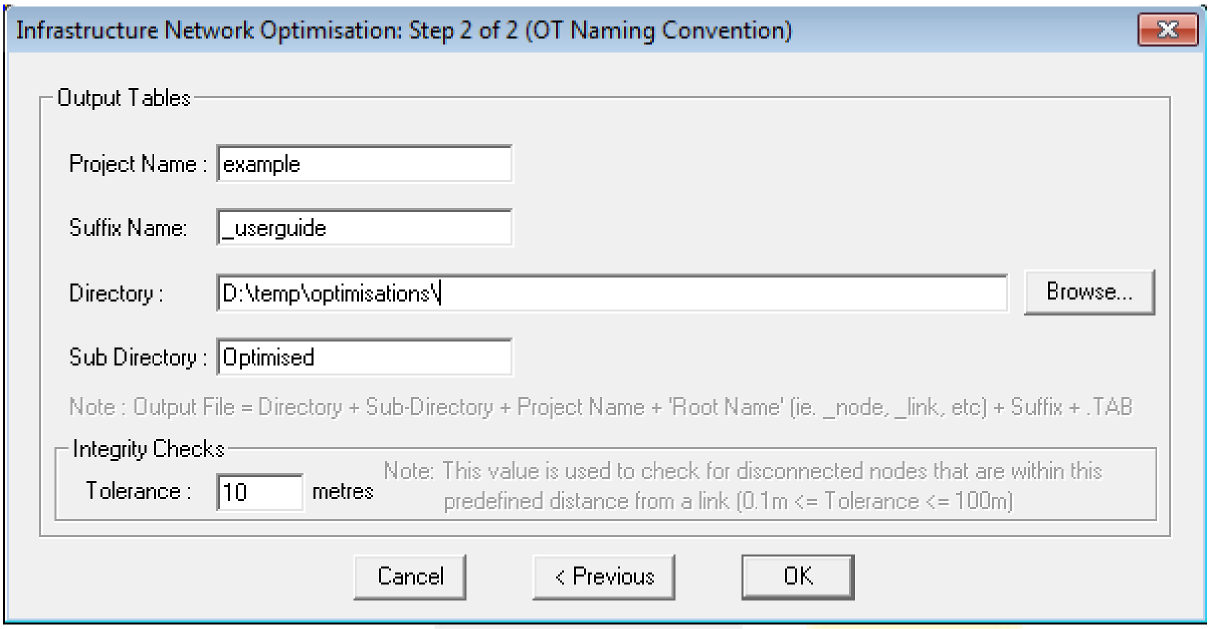


Figure 21 – Options for the Infrastructure Optimiser

Choose appropriate information for the naming and leave the tolerance at its default.

```
Optimising Highway
 Checking that turns reference nodes and links
 Fatal Error(s)
   There were 4 nodes(s) where inward and/or outward travel is prevented
     NodeNum(s): 16314,16322,16337,49755
     0.0 / 162.291227 -- 665.87 MB
   There were 1 turn(s) with missing links
     Turnnr(s): 209
     0.0 / 162.291227 -- 665.87 MB
   0.0 / 162.291227 -- 665.87 MB
 Non-Fatal Warning(s)
   There were 12 link(s) where the Capacities were different for each direction
     Linknr(s): 1863,8007,11703,11945,23698,33868,34606,39186,117980,119623,120026,121043
     0.0 / 162.291227 -- 665.87 MB
   There were 24 centroid_connector(s) connected to 1 way links
     Linknr(s): 121163,121164,121168,121169,121171,121175,121179,121203,121218,121224,1212
     124670,124675,124713,124764
     0.0 / 162.291227 -- 665.87 MB
   There were 90 centroid_connector(s) connected directly to either PT or walk links
     Linknr(s): 121148,121150,121157,121174,121178,121184,121186,121188,121190,121192,1211
     121226,121230,121234,121236,121240,121242,121244,121246,121248,121250,121254,12125
     121289,121292,121296,121299,121303,121309,121328,121329,121336,121406,121416,12142
     122480,122502,122505,122776,122777,122833,122853,122933,123273,123353,123373,12356
     124205,124207,124210,124216,124219,125707,125708,125709,125717,130193
     0.0 / 162.291227 -- 665.87 MB
   There were 634 node(s) that are not used
```

Figure 22 – An unsuccessful Infrastructure Optimisation

Figure 22 shows the log provided by the optimiser following an unsuccessful optimisation. There are two types of messages ; Fatal Errors and Non-Fatal Warnings. Fatal errors MUST be fixed before importing into OmniTRANS. Non-Fatal warnings are extra information about potential issues with the network. Most of our networks contain a number of non-fatal warnings and as such they can often be ignored.

The above suggests that we have some one way links that are hitting nodes that result in dead ends, suggesting that we might have an issue with a freeway or freeway ramps connectivity. We're also told that one of our turns (toll or turn ban) is attached to the wrong node or should be removed. These are just two of many possible issues that might arise during infrastructure optimisation. These issues should be identified and fixed on the PRE-optimisation network and then the optimiser should be run again to ensure no new errors have been created.

Once this has been fixed we can progress to the PT Optimiser

4.6.2. Public Transport Optimiser

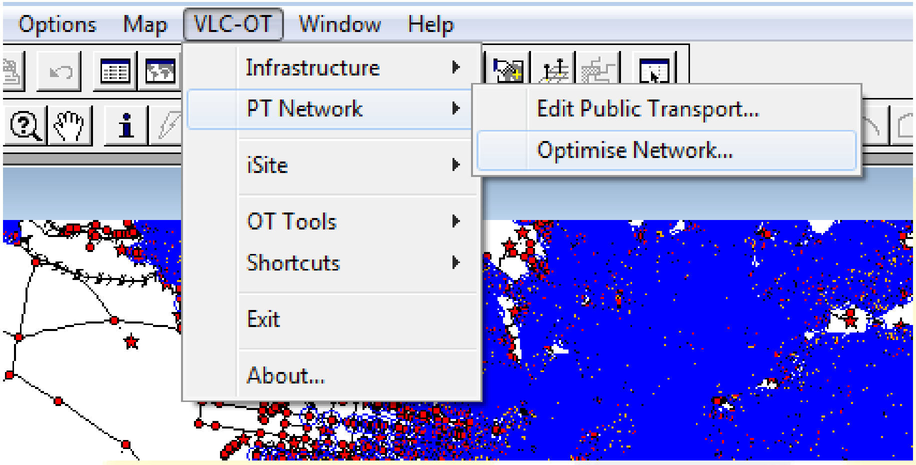


Figure 23 – Accessing the PT Optimiser

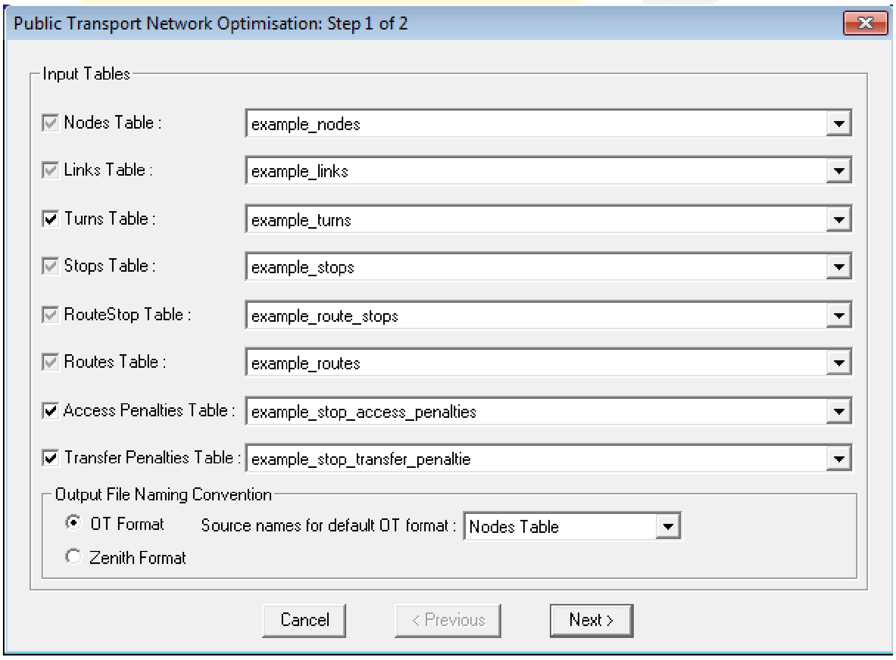


Figure 24 – Inputs to the Infrastructure Optimiser

Like the Infrastructure optimiser the PT optimiser should select the correct tables if they are named appropriately. Give them a quick check to ensure everything is right.

```
Optimising PT
 Fatal Error(s)
  There were 10 route misalignment(s)
   TransitLinenr(s) - [node1 -> node2]: 28218 - [16322 -> 49778]
   TransitLinenr(s) - [node1 -> node2]: 28218 - [49778 -> 49755]
   TransitLinenr(s) - [node1 -> node2]: 28237 - [16337 -> 16744]
   TransitLinenr(s) - [node1 -> node2]: 28237 - [16744 -> 16314]
   TransitLinenr(s) - [node1 -> node2]: 28238 - [16337 -> 16744]
   TransitLinenr(s) - [node1 -> node2]: 28238 - [16744 -> 16314]
   TransitLinenr(s) - [node1 -> node2]: 28240 - [16322 -> 49778]
   TransitLinenr(s) - [node1 -> node2]: 28240 - [49778 -> 49755]
   TransitLinenr(s) - [node1 -> node2]: 28242 - [16322 -> 49778]
   TransitLinenr(s) - [node1 -> node2]: 28242 - [49778 -> 49755]
   0.0 / 324.319769 -- 1024.98 MB
  0.0 / 324.319769 -- 1024.98 MB
 77.099317 / 324.319769 -- 1024.98 MB
```

Figure 25 – An unsuccessful PT Optimisation

Figure 25 shows the log provided by the optimiser following an unsuccessful optimisation. The PT component only has one type of error; Fatal Errors. Fatal errors MUST be fixed before importing into OmniTRANS.

The above suggests that we have transitline passing from one node to another that with no link between said nodes. This is a very common error when editing networks in MapInfo. Once these errors have been fixed on the PRE-optimisation network then the optimiser should be run again to ensure no new errors have been created.

5. Editing Demographics / Land Use

Demographic and land use data are fundamental inputs to any travel model. People, and the activities undertaken by those people, are the fundamental drivers of travel in our cities.

This section explains how to create and edit a new demographic / land use scenario.

5.1. Creating and Editing Demographic Scenarios

In OmniTRANS, demographic and land use data is stored for each travel zone.

OmniTRANS allows you store many different demographic and land use scenarios. Each scenario is stored in a separate "matrix cube". The term "matrix cube" is a fairly poor description, "demographic scenario" would be better! It owes its name to certain Dutch modellers, who tend to generate one set of trip matrices per demographic scenario – we can do better than that!

To view your current set of Matrix Cubes, open the Matrix Cube Manager, by pressing on the toolbar. The Matrix Cube Manager is shown in Figure 26 below.

The drop down menu (inside the Red box) allows you to select an existing Matrix Cube. Matrix cubes can be added and deleted using the buttons in the orange box. Such matrix cubes are typically named "Dem_y{year}_{ScenarioDescription}".

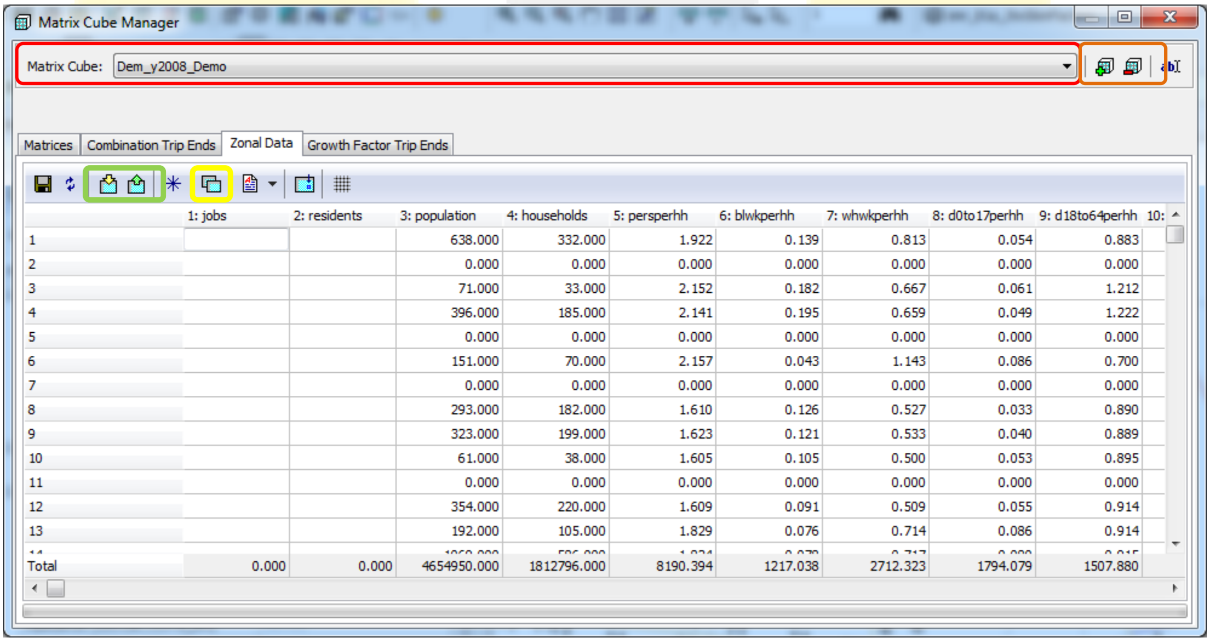


Figure 26 - The Matrix Cube Window

To view the Zonal Data for a Matrix cube, simply select the Zonal Data tab. Each row corresponds to an equivalently numbered Centroid object in the network. OmniTRANS provides tools for importing and exporting this data to external data formats, highlighted in the green box. Data can also be copied and pasted for a selection of cells using Ctrl-C, Ctrl-V.

Data can also be copied between Matrix Cubes by pressing the button inside the yellow box.

A demographic and land use scenario can be associated with a variant, by clicking on the Cube attribute of a variant, highlighted in Figure 27 below. This Matrix Cube will be used during the Zenith model run, unless it is explicitly overridden by the modeller in the run script (which is not recommended).

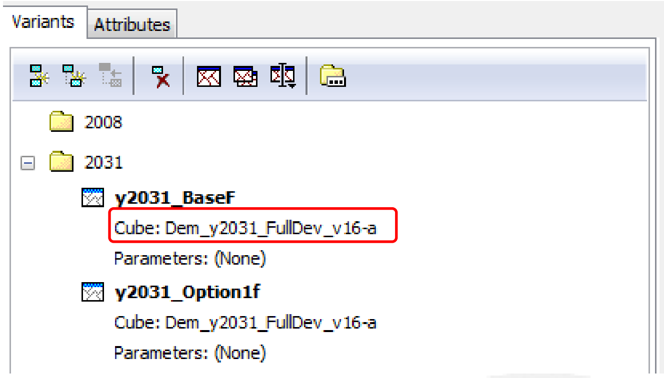


Figure 27 - The Cube Attribute of a Variant

Remember, after creating a new Matrix Cube, it is important to associate it with the appropriate variant(s).

Demographic and land use data can also be viewed and edited within the variant (network) window, by selecting a centroid, and opening the Attribute Editor, as shown in the Figure below. Don't forget to hit the Apply button after making changes! Demographic and land use data can also be displayed as histograms, pie charts, area colours (showing densities).

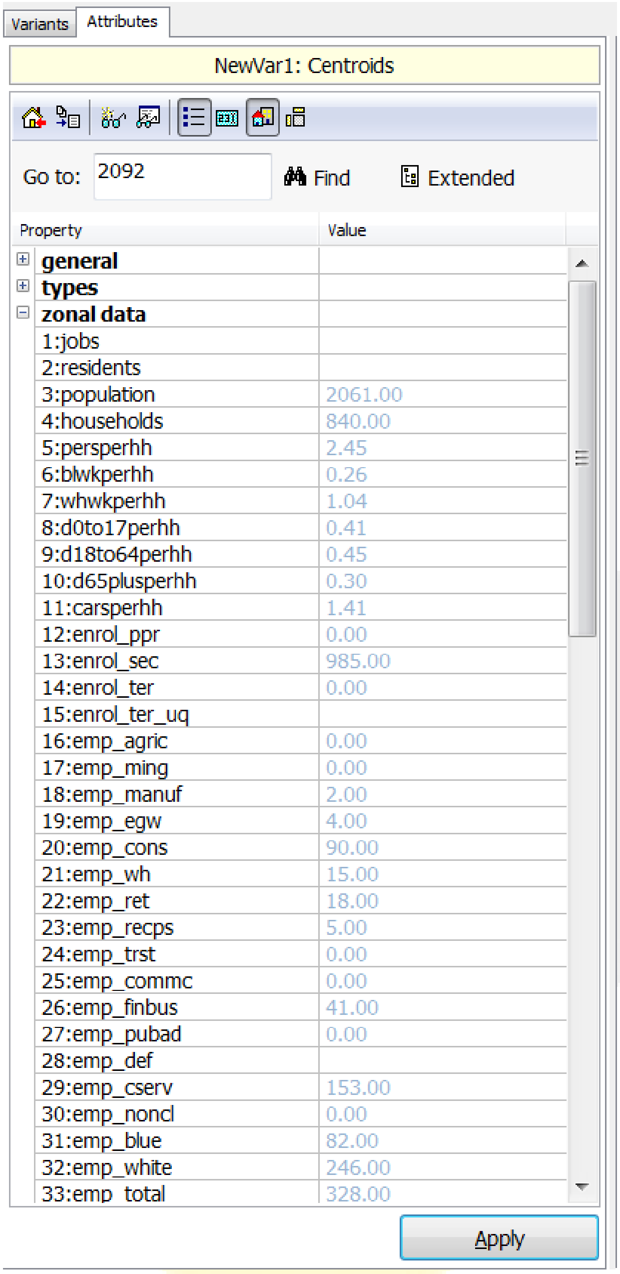


Figure 28 - Viewing and Editing Demographic and Land Use Data using the Centroid Attribute Editor

6. Editing Model Parameters

Model parameters round out the triumvirate of key model inputs (the other two are, of course, networks and demographics / land use).

Broadly speaking, there are two types of model parameters which can be modified in Zenith:

a. Policy variables (eg. fuel price, fares)
b. Behavioural parameters (eg. trip rates, modal constants)

We will discuss both in the following sub-sections.

There are also three key places where model parameters can be specified:

a. The OmniTRANS Project Setup
b. Zenith yaml files
c. A Zenith Model Run script

We will examine these first.

6.1. Storage of Model Parameters

6.1.1. The OmniTRANS Project Setup

OmniTRANS stores some standard model parameters in its Project Setup. The Project Setup Window, shown in Figure 29 below, can be accessed by pressing the button on the OmniTRANS toolbar.

The Project Setup Window has eight tabs:

1. Dimensions
2. Link Type & Modes
3. Types
4. Zonal Data
5. Combinations
6. Transit Transfers
7. Transit Fares
8. Extended Parameters

It is numbers 6 and 7 (Transit Transfers and Transit Fares) that concern us here, as they contain model parameters.

The **Transit Transfers** tab contains default penalties for access to, egress from, and transfer within the public transport system. Section Blah will explain how to modify these.

The **Transit Fares** tab defines the fare system(s) which exist in your model. Section Blah will explain how to modify these.

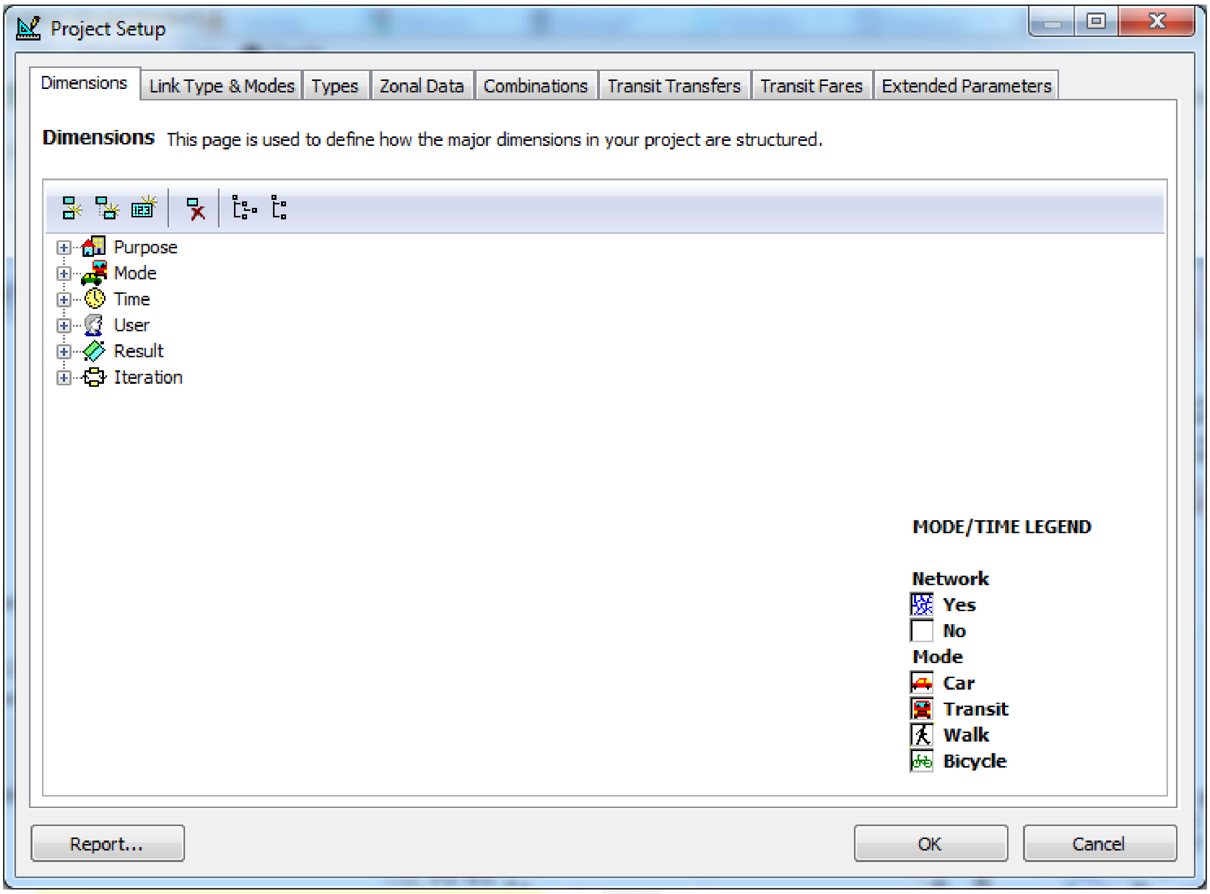


Figure 29 - The OmniTRANS Project Setup Window

6.1.2. Zenith yaml files

Open Windows Explorer and open the folder for your OmniTRANS project. Then open the 'data' folder within the project. Inside, you should find at least one file with the file extension 'yaml'.

If you open a yaml file in notepad you'll pleasantly discover its simplicity – it is just a text file.

Editing these simple text files allows you to control a wide array of model parameters and configuration.

Certain yaml files are automatically loaded when a ZenithModelRun is executed. To find out which files, navigate back to your project directory, and open the file 'project.yaml'. Inside you will find one or more yaml files referenced – these files are automatically loaded during a model run.

For example, in my project, my 'project.yaml' looks like this:

```
---

model_parameters: data/Melbourne.yaml

city: Melbourne

assignment_parameters: data/assignment-Melbourne.yaml
```

6.1.2.1. *Modifying YAML Files*

When modifying a yaml file, it is good practice to copy an existing yaml file, and edit the copy instead.

Then, when you run the model, you can instruct Zenith to load your new yaml file by using the 'additional_yaml_includes' property on ZenithModelRun.

For example (in Ruby):

```
myZMR = ZenithModelRun.new

myZMR.additional_yaml_includes = ['data\my_new_parameters.yaml']

myZMR.execute
```

Step-by-step

1. Identify the yaml parameter you want to modify, and locate the yaml file which contains this parameter.
2. Make a copy of the yaml file, and give it a new name
3. Use the 'additional_yaml_includes' property of ZenithModelRun to include the new yaml in your model run (as shown above).

6.1.2.2. *YAML Parameters are Really Just ZenithModelRun Parameters*

We're getting a bit under-the-hood here, but bear with me – this is useful!

We're going to skip ahead a bit and look at how ZenithModelRun's are configured.

The Ruby script for a simple ZenithModelRun (that just does assignment) looks like this:

```
myZMR = ZenithModelRun.new

myZMR.do_assignment = true

myZMR.execute
```

We are constructing a ZenithModelRun object, and then setting the 'do_assignment' property on that object to true.

Behind the scenes, though, there are lots more properties being set on this ZenithModelRun object.

In fact, every yaml parameter is a parameter on ZenithModelRun.

For example, if we look inside an assignment.yaml file, we'll find a definition for fuel price that looks like this:

```
fuel_price: 140
```

Behind the scenes, this becomes:

```
myZMR.fuel_price = 140
```

So why is this important?

It's important because it means that you can override yaml parameters in your Ruby script. We can change the fuel price to 150 using the following Ruby script:

```
myZMR = ZenithModelRun.new

myZMR.do_assignment = true

myZMR.fuel_price = 150

myZMR.execute
```

Pretty easy, huh?

The same rule applies for any yaml parameter. They can all be overridden in your Ruby script.

6.2. Policy Variables

This section explains how to modify some of the most important policy variables input to the Zenith model.

The policy variables we will consider are:

- Fuel price
- Fare levels
- Parking charges
- Tolls

6.2.1. Fuel Price

The fuel price is defined as a property of ZenithModelRun.

A default fuel price is normally included in your assignment yaml file, and looks something like this:

```
fuel_price: 140
```

There are two ways to modify the fuel price for a model run:

a. Create a new yaml and include it in your model run (see Section 6.1.2.1 for details)
b. Override the fuel price in your ZenithModelRun Ruby script (see Section 6.1.2.2 for details)

For simple scenario tests, it is often easier to choose the latter option. The first option makes more sense when you have a package of policy changes that go together to make a scenario (such as a 2031 scenario).

6.2.2. Fares

Fares can be modified in a number of ways. The Zenith model supports multiple fare system types, the most commonly used is a Zone to Zone system or matrix based (stop type to stop type), similar to how fares are charged in Melbourne and Brisbane. More information can be found in the OmniTRANS manual about adding and modifying Transit Fare Systems.

Another option would be to make a general assumption about the perception factor around fares. This can be accessed much like the fuel price, via the assignment parameters yaml. The fare 'weight' can be found here :

```
assignment_cost_function:
  PT:
    Time: 20
    Wait: 0.85
    Distance: 0
    Fare: 0.85
```

6.2.3. Parking Charges

Parking charges are used to approximate both the actual cost and perception of car parking in different areas. This is most important in the inner areas of most cities but should also be included at large institutions in outer areas (ie Monash University in Melbourne's east).

Zenith uses a type called ParkingZone to define centroids that include various parking charge types.

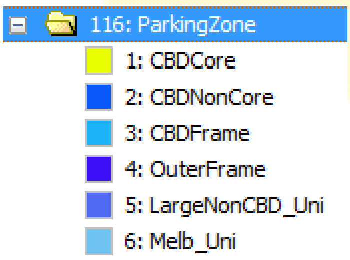


Figure 30 – The Parking Charge Types

These types are then accessed in the destination choice and mode choice parameters yaml files. These two values contain constants estimated from household travel surveys for each trip purpose and ParkingZone. Of particular note are two lines near the top of the mode choice parameters:

```yaml
mode_choice_model:
  BaseWithCycling:
    tree:
      Total:
        Mechanised:
          Car:
            - Car
          PT:
            - PT_Walk_Access
            - PT_Car_Access
        NonMechanised:
          - Walk
          - Bicycle
    scale_normalisation: bottom
    destination_constants_factor: 1.0
```

Similar can also be seen for BaseNoCycling. This destination constant factor allows us to factor the perception of every parking charge in the model for easy implementation of future parking charge increases or decreases.

Like with fuel it can be a good idea to set up a new yaml file overriding this value:

```yaml
mode_choice_model:
  BaseWithCycling:
    destination_constants_factor: 1.80
  BaseNoCycling:
    destination_constants_factor: 1.80
```

This type of modification would be the only suggested type of parking charge analysis that should be considered by a new modeller. Parking charges have considerable impact on Destination and Mode choice and should be treated with respect.

6.2.4. Tolls and Toll Caps

Tolls are an important part of the transport networks in most Australian cities. In Section 4.3 we learned how to code up tolls within the network editing features of OmniTRANS. Further to that we mentioned a concept of a 'Toll Cap'. This is commonly used in

6.3. Behavioural parameters

6.3.1. Public Transport Penalties

Public transport penalties are used to impact the varying perceptions that people have of different sub-modes. These are defined in the Transit Transfers tab of the Project Setup.

There are 3 types of penalty that are defined here;

- First access to a PT sub-mode
- Transfer from one PT sub-mode to another
- Last egress from a PT sub-mode

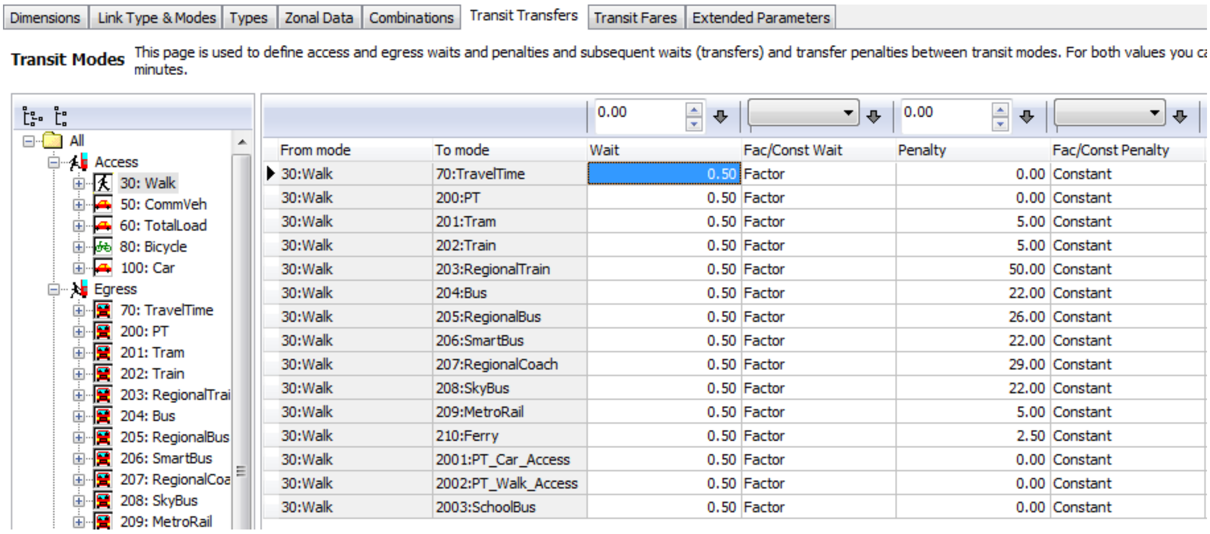


Figure 31 – Public Transport Penalties – Transit Transfers

6.3.2. Transit Crowding Cost Functions

Zenith features a sophisticated public transport crowding function that can be activated during the public transport assignment. This requires that every transitline in the network have specified a number of 'seats' per service and also the 'crush capacity' of each service. For this we need to specify 2 parameters in the model.

```
crowding_function: [[[0.0,0.0],[0.7,0.01],[1.0,0.1],[2.0,0.3]],[[1.0,0.4],[2.0,1.0],[10.0,20.0]]]

skim_crowding_function:[[0.0,0.0],[0.7,0.01],[1.0,0.1],[2.0,0.3]],[[1.0,0.4],[2.0,1.0],[10.0,20.0]]]
```

The crowding function is used to define the curve that generates the extra penalty for crowded conditions. Second is used to define the costs that are output as a 'crowding' penalty skim.

A graphical representation of these numbers can be seen in Figure 32

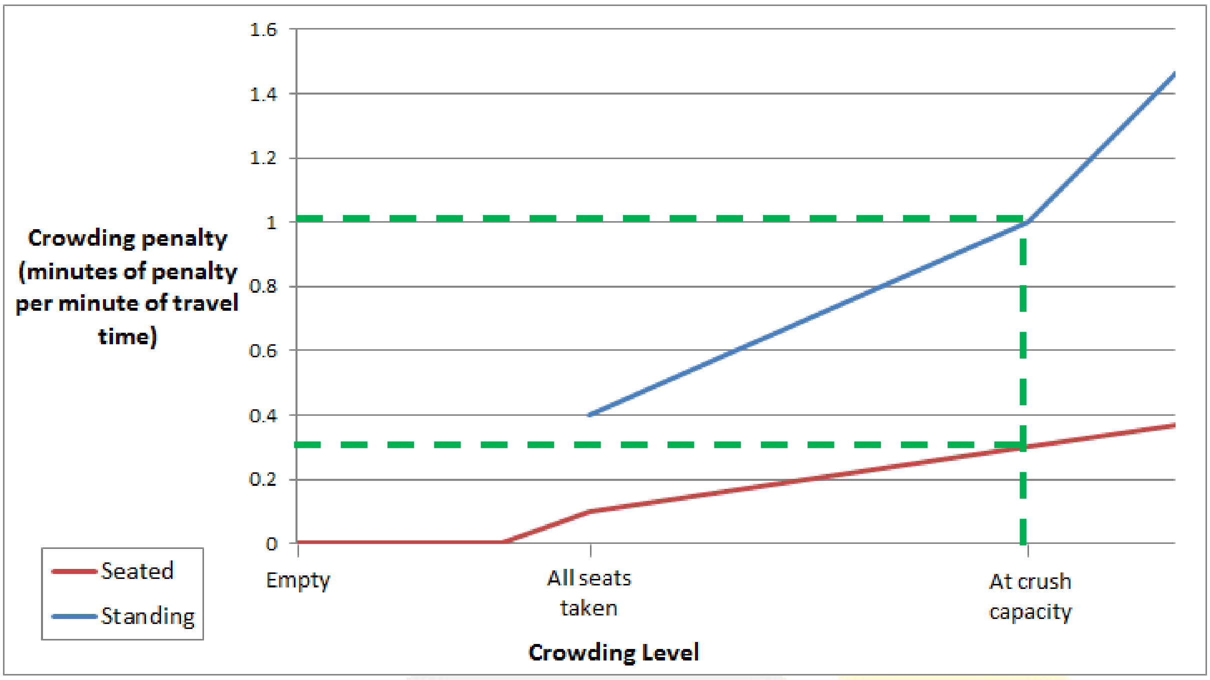


Figure 32 – Public Transport Constraint Crowding Functions

7. Running the Model

7.1. Basic Model Running

The Zenith model process is very complex with many options and configurations, thankfully most of this has been simplified for easy model running. The first step is to use the available user interface to create a script. Figure 33 and Figure 34 show where this interface can be accessed within OmniTRANS.

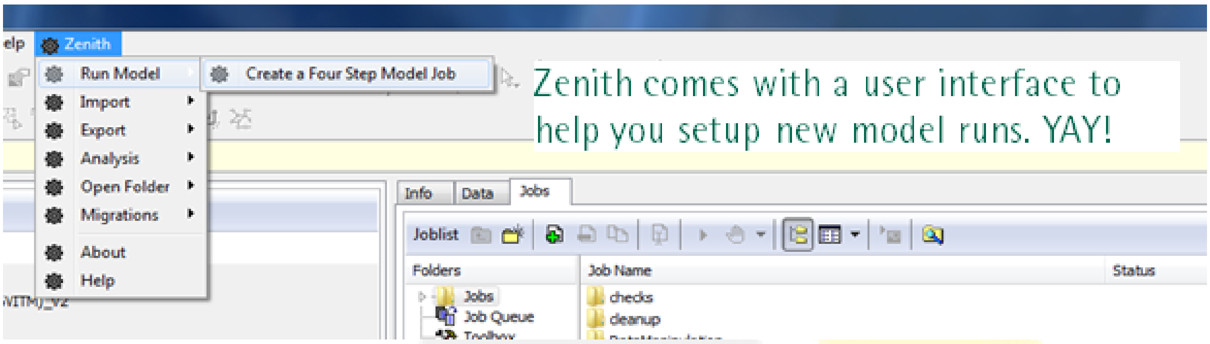


Figure 33 – Accessing the GUI creating model run scripts.

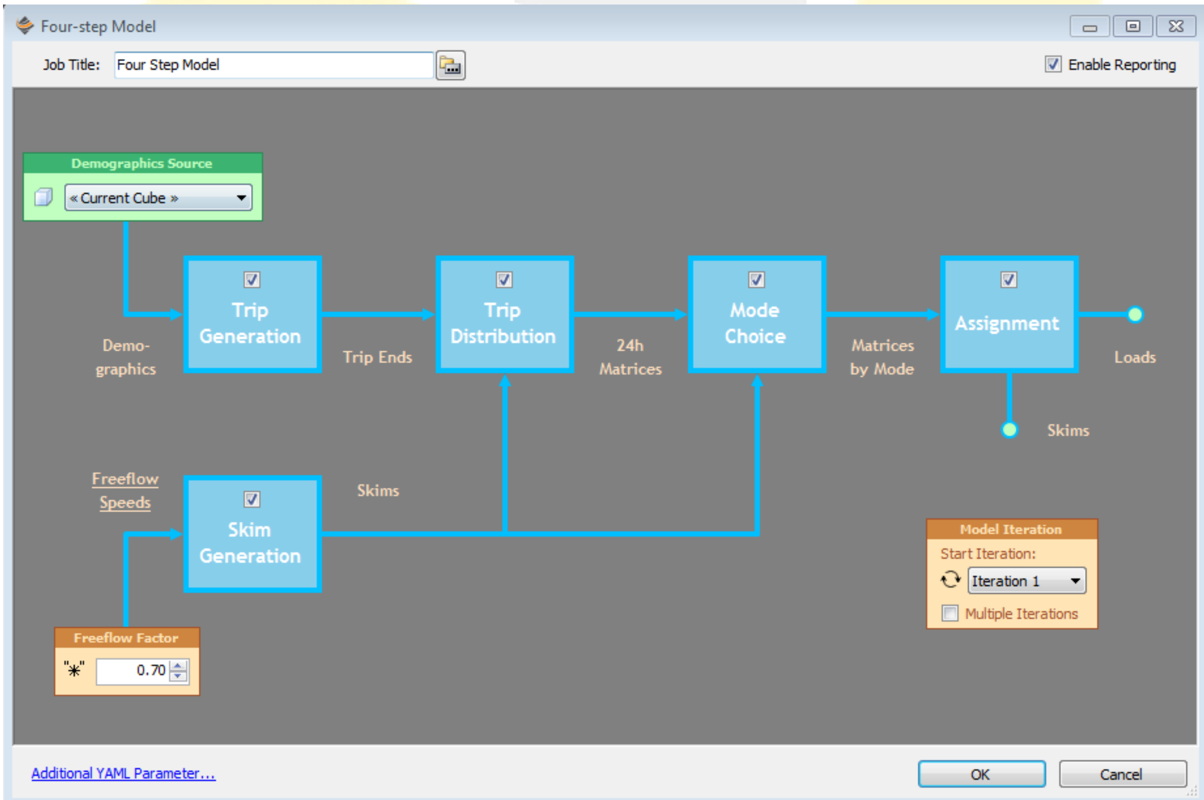


Figure 34 – The GUI for creating model run scripts

As you can see the creation of a run script is very easy, all that's required is a simple tick of a box for each step of the model run process. Certain steps have options for data input. These are shown in Green (input) or Brown (configuration) boxes.

Depending on the tick boxes and settings chosen a Ruby script is created in the location you choose at the top. An example of this can be seen

```
require 'zenith_model_run'

# Settings
myZMR = ZenithModelRun.new
myZMR.num_iterations = 1
myZMR.start_iteration = 1
myZMR.do_reporting = true

# Trip Generation
myZMR.do_trip_generation = true
myZMR.zonal_data_cube = $Ot.currentCube

# Skimming
myZMR.do_seed_skims = true
myZMR.seed_skims_method = :skim_free_flow_speeds
myZMR.free_flow_speed_factors = {
  'Car'      => 0.7,
  'CommVeh'  => 0.7
}

# Trip Distribution
myZMR.do_destination_choice = true

# Mode Split
myZMR.do_mode_choice = true

# Assignment
myZMR.do_assignment = true
myZMR.assign_traffic = true
myZMR.assign_pt = true
myZMR.assign_walk = true
myZMR.save_output_skims = true
myZMR.save_turning_movements = true

# Execute
myZMR.execute
```

Figure 35 – The run script produced by the GUI

As can be in the resulting run script, each tick box and input or option correlates to a simple line of Ruby code.

7.2. Advanced Model Running

As mentioned above there are a numerous advanced options within the model run process. The most common type of speciality run scripts are surrounding some of the more advanced assignment options including:

- Public Transport capacity constraint

- Toll diversion
- Select link analysis
- Stop to stop analysis
- Cordon creation

We'll cover a few of these options in this chapter.

```
require 'zenith_model_run'

# Settings
myZMR = ZenithModelRun.new
myZMR.num_iterations = 1
myZMR.start_iteration = 3
myZMR.periods_to_do = ["AM"]
myZMR.fix_distribution = true
myZMR.do_reporting = true

# Trip Generation
myZMR.do_trip_generation = false

# Skimming
myZMR.do_seed_skims = false
#~ myZMR.seed_skims_source = [$Ot.currentVariant.last, 1]

# Trip Distribution
myZMR.do_destination_choice = false
myZMR.mode_choice_trips_source = [$Ot.currentVariant.last, 1]

# Mode Split
myZMR.do_mode_choice = false
myZMR.assignable_trip_matrix_source = ['y2016_ReferenceCase',3]

# Assignment
myZMR.do_assignment = true
myZMR.assign_traffic = true
myZMR.assign_pt = true
myZMR.assign_walk = false
myZMR.assign_bicycle = false
myZMR.save_output_skims = true
myZMR.save_turning_movements = true
myZMR.save_component_pt_skims = false

myZMR.selected_link_sets       = [ ["SL_RichmondRail_IN","SL_RichmondRail_IN"]]
myZMR.selected_link_outputs    = [true,true]    # output loads and matrices
myZMR.selected_link_condition = :or

# Execute
myZMR.execute(true)
```

Figure 36 – A Select link assignment script

Figure 36 shows an example of a select link script. There are 3 options of note:

```
myZMR.selected_link_sets       = [ ["SL_RichmondRail_IN","SL_RichmondRail_IN"]]

myZMR.selected_link_outputs   = [true,true]   # output loads and matrices

myZMR.selected_link_condition = :or
```

selected_link_sets contains two arrays, each sub-array represents a select link (you can do multiple at once). Within each sub-array are 2 records, one is the name of the selection of links within your network and the other is the name of the Result you want your output stored in. This result needs to exist in the Project Setup -> Dimensions tab.

selected_link_outputs includes two Boolean options, the first as to whether or not you would like to save out the select link load, the second as to whether you require select link matrices. While it is easy to set these to true for any select link assignment you may wish to conduct, it can be worth setting only the appropriate option to true in the interests of saving hard disk space as select link matrices in particular contain a large amount of data.

selected_link_condition is the condition for the select link. If your 'selection' contains multiple links do you want to record people who go through any of the links (:or) or do you want to record people who go through all of your links (:and)

'

```
require 'zenith_model_run'

# Settings
myZMR = ZenithModelRun.new
myZMR.num_iterations = 1
myZMR.start_iteration = 3
myZMR.periods_to_do = ["AM"]
myZMR.fix_distribution = true
myZMR.do_reporting = true

# Trip Generation
myZMR.do_trip_generation = false

# Skimming
myZMR.do_seed_skims = false
#~ myZMR.seed_skims_source = [$Ot.currentVariant.last, 1]

# Trip Distribution
myZMR.do_destination_choice = false
myZMR.mode_choice_trips_source = [$Ot.currentVariant.last, 1]

# Mode Split
myZMR.do_mode_choice = false
myZMR.assignable_trip_matrix_source = ['y2016_ReferenceCase',3]

# Assignment
myZMR.do_assignment = true
myZMR.assign_traffic = false
myZMR.assign_pt = true
myZMR.assign_walk = false
myZMR.assign_bicycle = false
myZMR.save_output_skims = true
myZMR.save_turning_movements = true
myZMR.save_component_pt_skims = false

myZMR.stop_to_stop_matrix       = [$Ot.network.get_stops_for_mode(["Train"],false,true),'s2s_rail']


# Execute
myZMR.execute(true)
```

Figure 37 – A Stop to stop assignment script

8. Analysing Model Inputs and Outputs

The Zenith modelling framework provides a vast range of analysis tools to analyse model inputs and outputs. This section covers a broad selection of the available tools. The main categories are:

- Spreadsheet outputs
- OmniTRANS thematic maps
- OmniTRANS reports

The following sub-sections discuss each of these in turn.

8.1. Spreadsheet outputs

The Zenith model comes with built-in tools for producing standard Excel reports, and for creating new report templates.

This section covers a range of available spreadsheet outputs and concludes with an outline on how to create custom spreadsheets. The topics which will be covered in this section are:

- The Zenith Model Run Spreadsheet
- The Model Convergence Spreadsheet
- The Traffic Validation Spreadsheet
- The PT Validation Spreadsheet
- The Economics Spreadsheet

8.1.1. The Zenith Model Run Spreadsheet

Any time a Zenith Model Run has been performed, a wealth of Key Performance Indicators (KPI's) will become available. The Zenith Model Run Spreadsheet is able to gather these and present them in a concise manner using a pre-formatted spreadsheet. This spreadsheet may contain results from multiple runs, enabling the user to:

- Check a Model Run for potential problems;
- Compare a future year to a base year scenario;
- Compare options to each other.

It is possible that not all information is available from a Zenith Model Run. This usually happens when only a specific part of a Zenith Model Run has been performed (eg. an assignment only run). In these cases, some rows within the spreadsheet will be left blank.

Accessing the Zenith Model Run Spreadsheet

The Zenith Model Run spreadsheet is generated anytime a Zenith Model Run cycle is completed successfully. It is located in the OmniTRANS project, under 'Reports\Zenith Model Run Report.xlsx'.

Interpretting the Zenith Model Run Spreadsheet

The Zenith KPI's spreadsheet contains 8 main sheets which cover:

- Demographics
- Matrix totals
- Skim totals

- Crowding
- PT Summary
- Traffic Summary

The majority of these sheets store model data by time period.

Demographics
Even though demographics are an input to the traffic model, inclusion of this data allows the user to quickly look op zonal data and verify its validity for that particular model run. The demographic information provided shows:

- Total Population
- Total Households
- Total White Collar Workers
- Total Blue Collar Workers
- Total Dependants 0-17
- Total Dependants 18-64
- Total Dependants 65+
- Total Cars
- Average Household Size
- Average White Collar Workers per Household
- Average Blue Collar Workers per Household
- Average Workers per Household
- Average Dependants 0-17 per Household
- Average Dependants 18-64 per Household
- Average Dependants 65+ per Household
- Average Cars per Household
- Total Primary Enrolments
- Total Secondary Enrolments
- Total Tertiary Enrolments
- Total Employment

Matrix totals
A Zenith Model run produces a plethora of matrices. This section of the Zenith KPI's spreadsheet covers some of the main performance indices dealing with trip matrices.

Category	Subcategory
Total Person Trips	
Mode Share	% Person Cars % Public Transport % Walk/Cycle
Car Split	% CompanyCars % NonCompanyCars % AirportCars
Public Transport Split	Walk Access / Walk Egress Walk Access / Car Egress Car Access / Walk Egress
Person Car Car Commercial Vehicle	Trips Intrazonal Trips Non-Intrazonal Trips Average Vehicle Occupancy (Car only)
Public Transport Trips	Walk Access / Walk Egress Trips Walk Access / Car Egress Trips

	Car Access / Walk Egress Trips Total Trips
Walk / Cycle	Trips Intrazonal Trips

Table – Zenith Model Run Spreadsheet - Matrix Totals description

Skim totals

Skim totals provide an early indication on how a traffic model has performed during a model run in terms of cost by comparing the output skims to the input skims. For example, improving public transport will lower the total cost for using the public transport system. Because of the improved public transport system, demand will increase. It is likely that part of this demand are persons who used to drive but are using the public transport system instead after the improvement. The decrease in demand for cars will lessen the strain on the road network, lowering the car skims as well. This effect should be apparent using this sheet, if not, further investigation might be required.

Both input and output skims are available by time period in the Skim Totals sheet. They show skim totals for:

- Walk / Cycle
- Car
- Commercial Vehicle
- Public Transport (Walk Access / Walk Egress)
- Public Transport (Walk Access / Car Egress)
- Public Transport (Car Access / Walk Egress)

Crowding

The crowding sheet provides information about the amount of crowding on the network. This amount is expressed in total number of kilometres incurred in a specific crowding range.

Because private and commercial vehicles share the road, the volume over capacity ratio is calculated by adding the volumes for both vehicle classes and dividing it by the total capacity for vehicles per link.

Crowding on public transport is calculated by dividing the total amount of passengers on a section of a transitline by the crush capacity of the vehicle. NB This capacity is the absolute maximum a vehicle can carry, not necessarily the capacity it was designed for!

Not standard at the moment, but available if required, the amount of crowding can be expressed in hours as well instead of kilometres per crowding range.

Category	Crowding Ranges
Private Vehicle Commercial Vehicle Train Tram Bus	0 to 0.7 0.7 to 0.85 0.85 to 0.95 0.95 to 1.05 1.05 to 1.2 1.2 to 1.5 1.5 to 2 2 to 3 3+

PT Summary
This sheet contains a summary of Key Performance Indicators for public transport.

Category
Total Dwell Time
Total Service Stops
Total PT Trips
Interchanges
Boardings per trip
Waiting Hours
Walking Hours
Car Access/Egress Hours
PT Car Access Trips
PT Walk Access Trips

Category	Subcategory
Service Kilometres Total Boardings Mode of Access In Vehicle Passenger Kilometres In Vehicle Passenger Hours Access Hours	Metro Rail Suburban Rail Regional Rail Total Rail Bus Tram Total All Modes

Traffic Summary
The traffic summary sheet contains both input and output information about traffic indicators.

Category	Subcategory
Network (Input)	Total Kilometres Total Capacity (Vehicle Kilometres per Hour) Total Traversal Time
Cars Commercial Vehicles	Total Trips Total Kilometres Total Hours Average Assigned Speed Average Freeflow Speed Average Volume over Capacity Ratio
Person Car Hours	

Consumer Surplus
This sheet contains a wealth of consumer surplus information which can be used to understand the impacts of different schemes such as an infrastructure change, a transit service improvement, a change in land use or demographic patterns or a change in travel behaviour. This is done by preparing a Base Case which is used to represent "business as usual" or a "do nothing" scenario and a Project Case which contains changes from the Base Case which will be evaluated.

For more in depth information about consumer surplus, see the Consumer Surplus Technical Note which can be found in the OmniTRANS Help (Help -> Technical Documents -> Zenith -> Consumer Surplus Technical Note).

All benefits and disbenefits presented in this sheet are calculated in hours, except for the fares which are in 1996 dollars. See table below for an overview of the generated consumer surplus indicators.

Category	Subcategory
In-Vehicle time	Existing Users
Waiting Time	New Users
Transfer and Access Penalty	Lost Users
Walk Access	
Walk Transfer	
Walk Egress	
Car Access	
Car Egress	
Fares	
Car Time Savings	
Commercial Vehicle Time Savings	
Total Benefits	

8.1.2. The Model Convergence Spreadsheet

At the conclusion of a multiple-cycle four step model run, it is important to check that the model has reached an acceptable level of convergence. The level of convergence which is 'acceptable' depends on the application, and is typically 'most strict' in the cases where Consumer Surplus outputs are being generated for economic assessment.

A lack of model convergence introduces 'noise' into the model results, which can in some cases 'drown out' the impact of changes to the model's inputs made by the user. This makes comparing model results difficult, and potentially futile.

The Model Convergence Spreadsheet allows the user to quickly examine the level of convergence achieved by the model.

A broad rule of thumb followed by VLC is that the difference in cost, total kilometres and total hours for cars should differ by less than 0.5% between subsequent iterations.

Creating a Model Convergence Spreadsheet
The Model Convergence Spreadsheet contains all the basic information to analyse model convergence. The sheet contains only a single model run, with each column containing the results from a single model iteration.

To create a model convergence spreadsheet, run the **'Convergence Report.rb'** script which is located in *'Jobs\Reporting\Spreadsheets\'*.

There might be a need to change some settings in this script depending on the performed Zenith Model Run or required output information. Below is an extract of the Convergence Report script.

```
1.upto(4) do |i|
  reporter = NewConvergenceReporter.new
  reporter.writeErrorMessage = true
  reporter.execute({:iteration => i, :doMacro => true, :doMicro => false})
end
```

iterations: every column in the spreadsheet will contain convergence information for a single column. The software is able to output information for multiple iterations at once, i.e. there is no need to run the script multiple times in order to output multiple iterations. The required iterations can be controlled in the block **1.upto(4) do |i| … end**.

writeErrorMessages: error messages can be suppressed by setting the **writeErrorMessages** parameter to false.

doMacro: convergence information can be extracted at two detail levels. Currently, the higher level is fully supported, set **:doMacro** in the execute parameter to *true* to use it. The lower level will be featured in a future version of the Zenith Modelling software (:doMicro).

After a successful run of the Convergence Report script the resulting spreadsheet will be available in the Reports-directory of the project with the name 'Convergence_Results_<variantname>.xls'.

Contents of Convergence spreadsheet
Detailed model information is contained in the sheet 'Macro Data'. This information is displayed for each time period in the model (when available) by iteration in four sections. See table below for the contents.

Category	Subcategory
Skim Totals	Walk Car Commercial Vehicle PT Walk Access / Walk Egress PT Car Access / Walk Egress PT Walk Access / Car Egress
Matrix Totals	Walk Car Car Intrazonal Person Car Commercial Vehicles Commercial Vehicles Intrazonal PT Walk Access / Walk Egress PT Car Access / Walk Egress PT Walk Access / Car Egress All modes (excl CV)
Mode Split	% Walk % Person Car % Public Transport % Total
Other Data	Total Vehicle Kms Total Vehicle Hours Average Trip Length (Kms, excl intrazonals) Average Trip Time (Mins, excl intrazonals) Average Travel Speed (Kms/hr, excl intrazonals) Total Passenger Kms Total Passenger Hours Number of Boardings Number of Interchanges

A graphical representation of some of the data is available in the next four sheets. The more horizontal the lines, the more converged the model is. Displayed are:

- Car Matrix Totals
- Car Skim Totals
- Kilometres Travelled
    - Vehicles (Car / Commercial)
    - Public Transport
- Mode share
    - Walk
    - Person Car
    - Public Transport

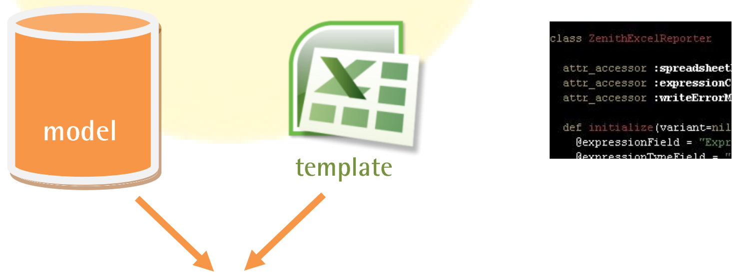
8.1.3. The Consumer Surplus Spreadsheet

8.1.4. Creating custom spreadsheets

Introduction
Besides having a number of pre-defined scripts the Zenith modelling framework provides a powerful tool to create custom automated spreadsheets called the Zenith Excel Reporter. This chapter explains how to use the tool and its advanced discusses some of the tools available and how to setup and run these spreadsheets.

Methodology
The Zenith modelling spreadsheets contain column by column information about a model run. A template spreadsheet is used which contain pre-formatted columns and code which is used by the Zenith Excel Reporter in order to evaluate expressions.

These columns are populated by the Zenith Excel Reporter which takes expressions to evaluate from a column in the template spreadsheet.



The Zenith Excel Reporter needs three elements to run:

- An OmniTRANS project containing model data
- A preformatted Zenith Spreadsheet Template
- A script

Template spreadsheet setup
The Zenith Excel Reporter will try to find two cells containing the names 'ExpressionType' and 'Expression'. Both fields are a requirement for the Zenith Excel Reporter to work.

The ExpressionType-column is used to determine whether an expression is used internally in the spreadsheet or if it requires external calculations. When set to 'Excel', the expression will be handled internally in excel, for example to add values from different cells. In case the ExpressionType is set to 'Ruby', the expression will be passed on to the Zenith Excel Reporter for further execution, like a calculation of total network kilometres by adding lengths of all links in the network together.

Native Excel Expressions
When the ExpressionType is set to Excel, the formula in the Expression column will be evaluated by Excel by copying the particular formula into the output column which is used to output data. Since it's a copy, all references to other cells are relative (unless absolute references are used, like '=A1'). The formula has to be a valid Excel formula in order to work, for example '=SUM(D30:D32)'. This assumes the Expression column is in column 'D'. Excel will try and evaluate this formula in the Expression column as well and might return an error in this actual column. This is no problem as only the formula is copied to the output column and not the actual value!

```
=SUM(D30:D32)
=D18/D27
=SUMPRODUCT(D46:D48,D18:D20)/D21
```

Model specific expressions
An Expression will be passed on to OmniTRANS and evaluated when the ExpressionType is set to 'Ruby'.

In the current implementation, local variables defined in a Zenith Excel Report script can be used in the Expression column. These local variables could for example be a skim matrix cube, a list of modes, an object etc.

```
sc_mats = OtSkimCube.new("y2006")
m_rail = ["Train","RegionalTrain","MetroRail"]
energy = ZenithEnergy.new
```

When for example 'sc_mats' have been made available through the run script, the Expression column could contain an expression like 'sc_mats.get([1,100,12,1,4,5]).sum' in order to calculate and output the sum of a car morning peak matrix for iteration 5.

It is considered good practice to define the iteration number as a local variable as well, since the iteration number could change in case outputs from a specific iteration are required. By creating a local variable instead of defining it in the Expression column, the only place where it needs replacing is inside the Zenith Excel Reporter run script.

```
iteration = 5
sc_mats = OtSkimCube.new("y2006")
zer = ZenithExcelReporter.new
  [...]
```

Some classes have so-called class methods. The advantage of these classes is that they don't need to be initialized as a local variable first using a constructor (i.e. energy = ZenithEnergy.new) and then in the Expression column have a method call on that variable (i.e. energy.fuelConsumption([1,100,12,1,13,iteration])). These classes can be called directly, like LinkCalc.calculate("SUM(length)").

The aforementioned LinkCalc class is part of a suite of tools to easily extract data from the model database. It does require some knowledge of the database fields, but removes the need to create full database queries. For more information on the ObjectCalc class of which the LinkCalc class is a part of, see the class documentation.

8.2. OmniTRANS Thematic Maps

OmniTRANS provides a wide array of visualisation tools, enabling the user to analyse model input and output data, and to produce report quality graphics.

OmniTRANS provides five types of thematic map, which can be used singly or in combination. These five types are listed in Table 3 below. In OmniTRANS, they can be found on the Graphics Toolbar, which usually runs down the left side of the OmniTRANS window.

Icon	Description
	Object colouring plots
	Histograms
	Pie Charts
	Link Bandwidths
	Desire Lines

Table 3 - OmniTRANS Thematic Map Types

8.2.1. Designs and Designers

For each type of thematic map, it is possible to save a graphic template – or as OmniTRANS calls it, **"a design"**. A design can be re-used to produce consistent graphic displays on any number of different variants.

By clicking on the tick beside a given type of thematic map (as seen in Figure 38 below), the user can choose from a range of pre-existing "designs", and configure various display settings associated with that design.

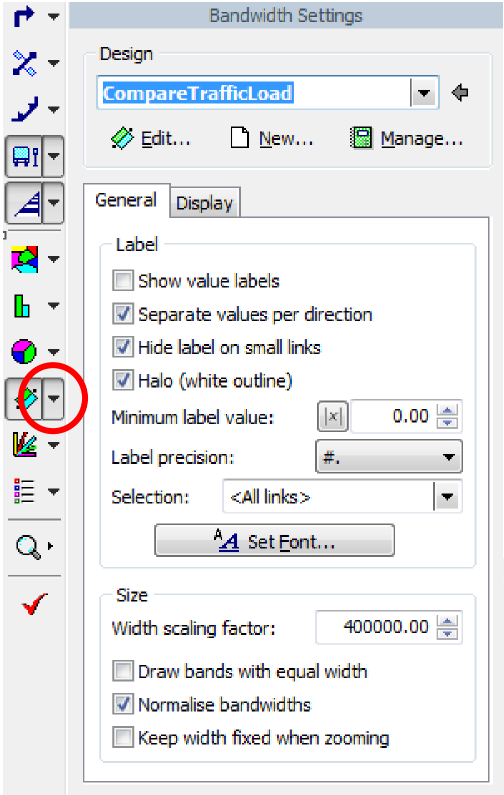


Figure 38 - Bandwidth Settings

A design can be edited by using its **"Designer"**. The designer is opened by clicking "edit" button.

The Designer typically allows the user to select what data is being displayed by a given design. For example, which purposes? which modes? which times? which model iteration? etc. An example is provided in Figure 39 below.

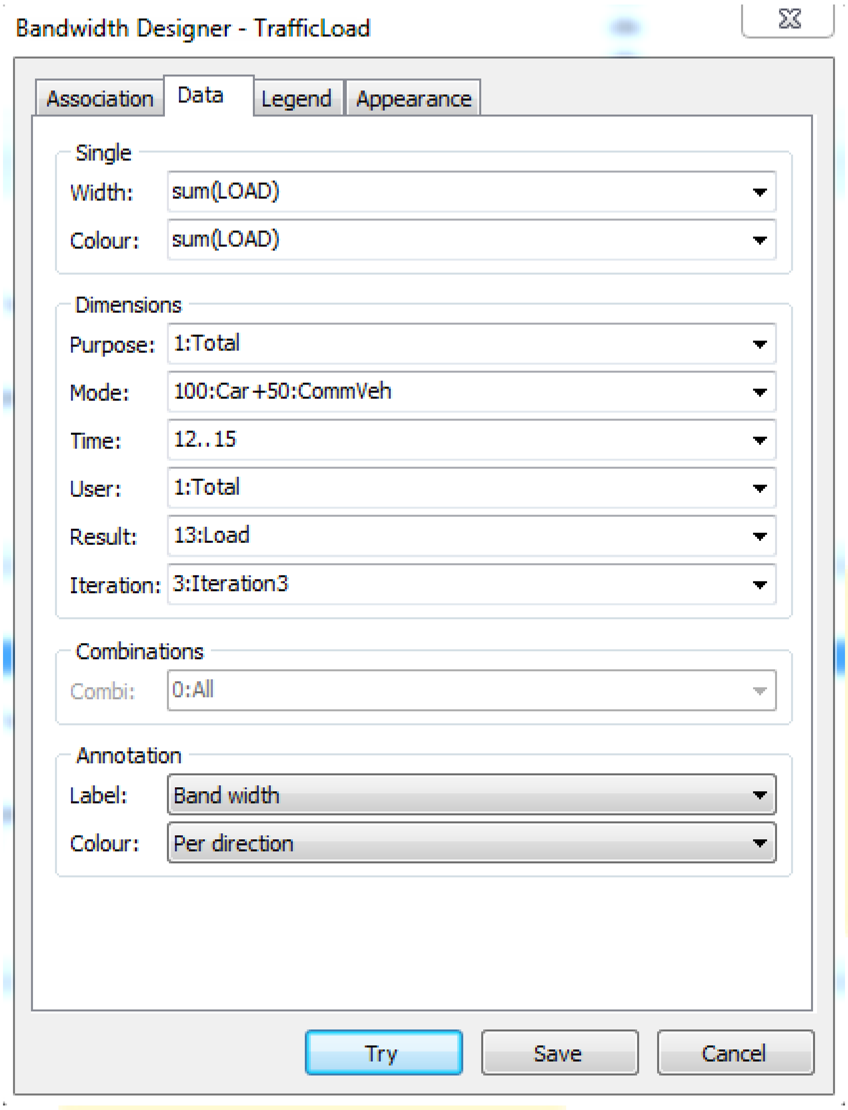


Figure 39 – The Bandwidth Designer

The designers are key to working with OmniTRANS, so is worth investing the effort required to master them. To learn more, go to the OmniTRANS Manual *[The User Interface -> Working with networks -> The Designers]*.

8.2.2. Standard Zenith Designs

The Zenith model comes with a number of commonly used "designs" which are useful for analysing Zenith inputs and outputs with a minimum of configuration.

The designs are letter coded and organised into folders to make it easier to find specific designs.

The letter codes are:

[ASG] Assignments, either single or multiple (stacked) assignments
[ASGi] As before, but using an interactive selection on that object.
[SL] Selected Link Assignment
[SZ] Selected Zone Assignment
[LU] Analysis of demographic (zonal) data

[MV] Model Validation (usually comparisons against countdata)

In many cases, the user will be required to slightly alter the pre-existing design; for example, to select outputs from the correct model iteration (the default value will always be iteration 1).

To help with this process, we have provided a series of step-by-step instructions in the sub-sections that follow.

Also keep in mind that while we have tried to make the set of preformatted designs as comprehensive as possible, we have barely scratched the surface of what is possible. With a bit of imagination, *almost* anything is possible.

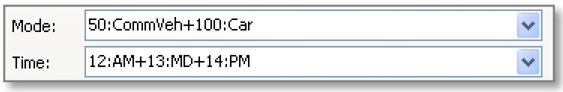
In the following sub-sections, we provide step-by-step instructions on:

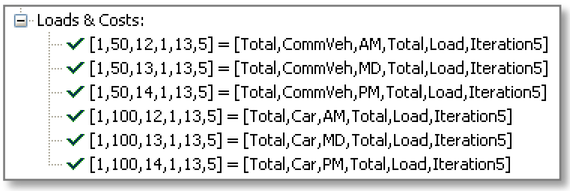
How to visualise Individual Assignments (Single)
How to visualise Multiple Assignments (Stacked)
How to visualise Volume over Capacity Ratios
How to visualise Selected Link Assignments
How to visualise a Comparison between Assignments
How to visualise a Comparison between an Assignment and Counts
How to visualise Stop Interchange Information
How to visualise First Boarding and Last Alighting Information

How to visualise Individual Assignments (Single)

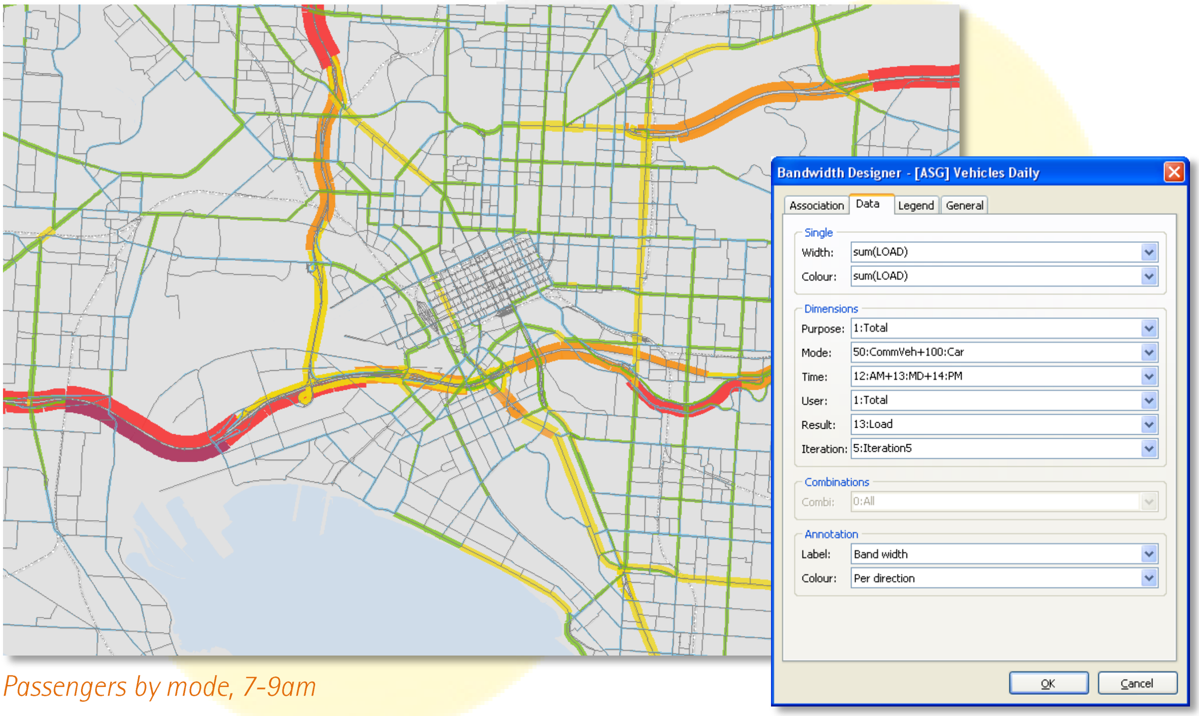
Description
Individual assignments allow visualisations of loads using ranges for colouring.

Considerations when using these designs
Multiple assignments can be shown at once, though the loads will be summarized. Add dimensions in order to show a summarized load. Make sure all requested loads are available. In case of for example car and commercial vehicles for three time periods to create a daily total vehicles visualisation means six loads need to be available!





Example



Passengers by mode, 7-9am

Available stacked designs
[ASG] Vehicles 7-9am
[ASG] Vehicles 4-6pm
[ASG] Vehicles daily

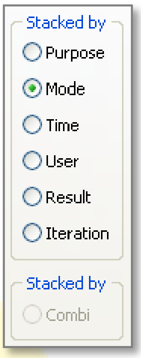
How to visualise Multiple Assignments (Stacked)

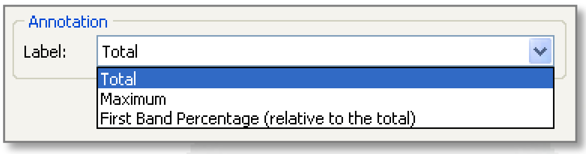
Description

Showing multiple assignments is extremely useful when information from each individual assignment needs to be retained within a visualisation of the total assignment. A percentage commercial vehicles in relation to the total number of vehicles on the network is an example of a stacked assignment. Passengers using different public transport modes is another example. Stacked assignments can be coloured individually.

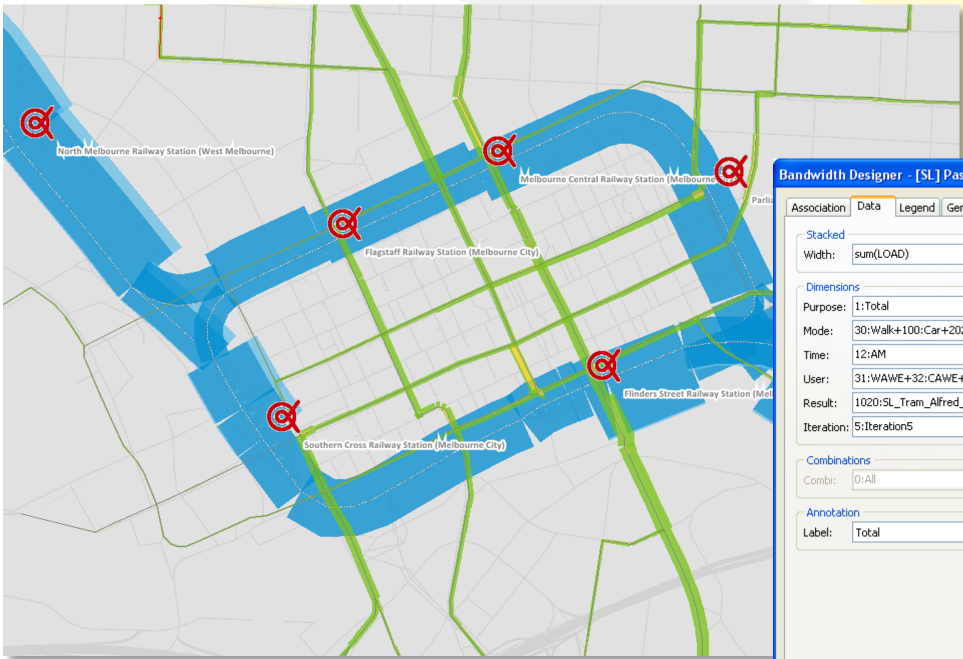
Considerations when using these designs

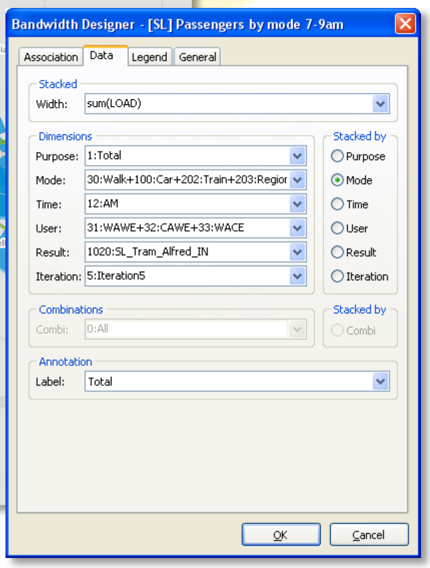
A common usage of these designs is to stack loads by mode. It is possible however to stack by different dimensions, like time periods or results. Use the 'Stacked by'-column to select the dimension to stack by. Also, a percentage to total labelling option is available for the first band.





Example





Passengers by mode, 7-9am

Available stacked designs

[ASG] Passengers by mode 7-9am
[ASG] Passengers by mode 4-6pm
[ASG] Passengers by mode daily

[ASG] Percentage CommVeh 7-9am
[ASG] Percentage CommVeh 4-6pm
[ASG] Percentage CommVeh daily

How to visualise Volume over Capacity Ratios

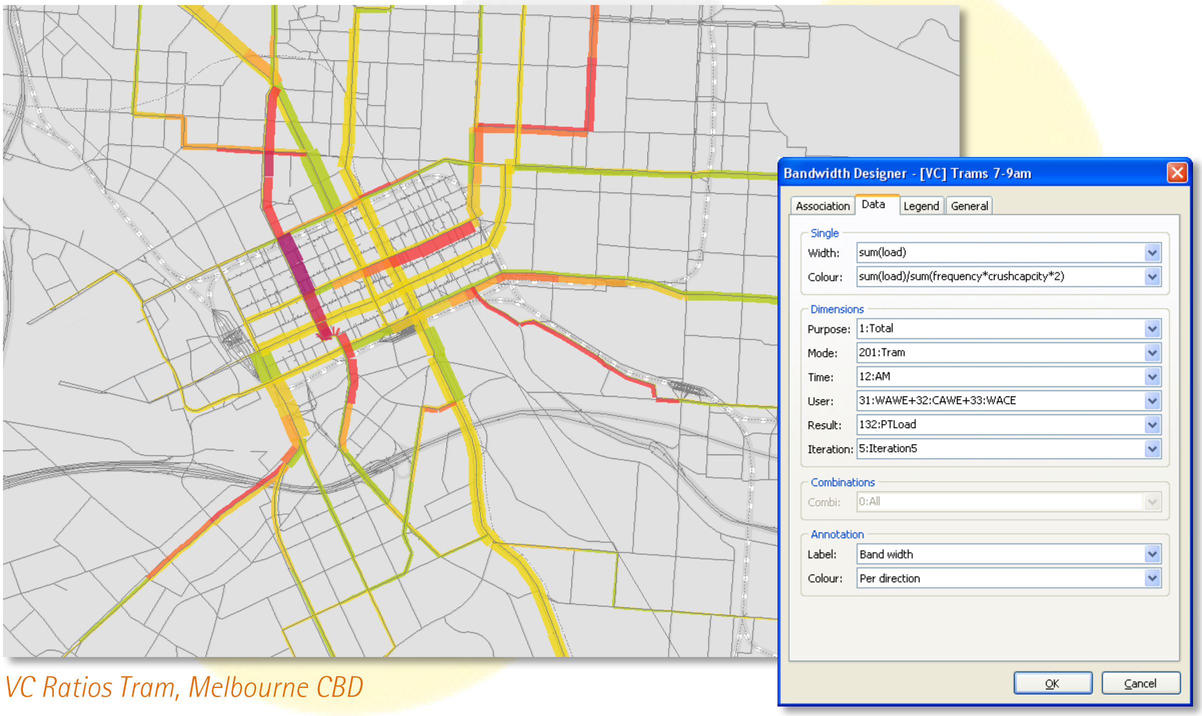
Description

Volume over capacity ratios are very useful in determining bottlenecks in the network. They can both be calculated for vehicles as well as public transport depending on the availability of the loads. Usually the width of the bandwidths indicate the number of vehicles / passengers and the colour of the bandwidths indicate the volume over capacity ratio.

Considerations when using these designs

When creating custom Volume over Capacity Ratio designs, remember that the capacities should match the duration of that particular time period. For example, a 7-9am morning peak means the hourly capacities need to be multiplied by a factor two in order to get capacities for the whole period. The expression would be: 'sum(load)/avg(capacity)*2'. As the previous expression demonstrates, use average capacity to determine the capacity of a link. Multiple modes and/or time periods would result in a summation of capacities when using sum(capacity).

Example



VC Ratios Tram, Melbourne CBD

Available VC Ratio designs

[VC] Trains 7-9am
[VC] Trams 7-9am
[VC] Vehicles 7-9am

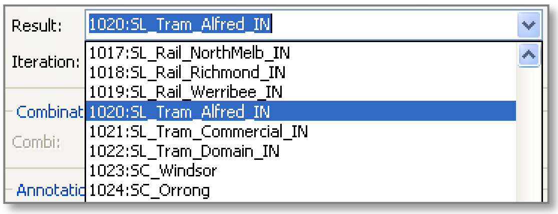
How to visualise Selected Link Assignments

Description

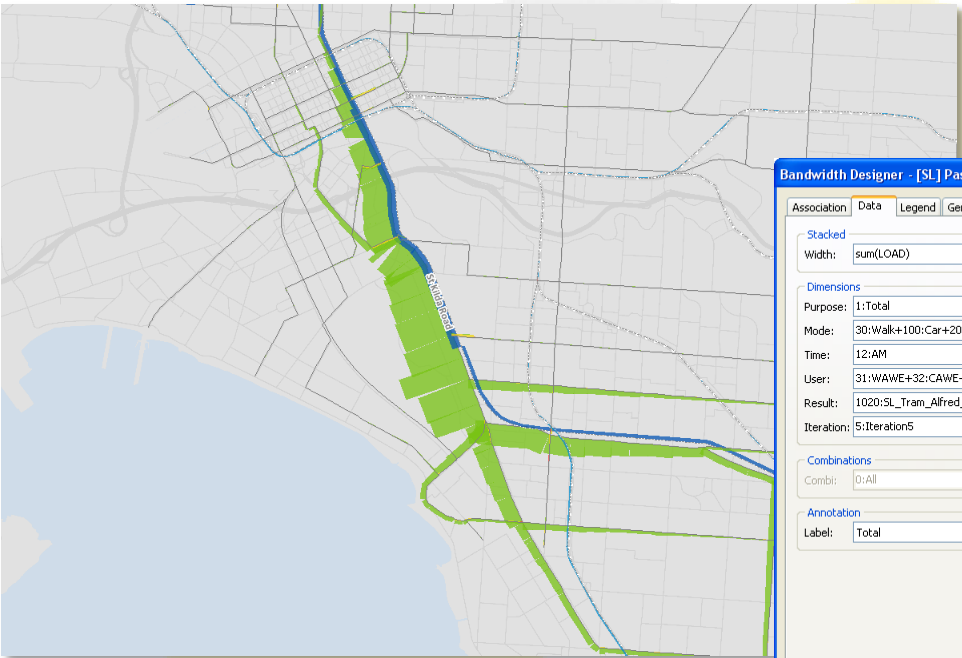
A selected link assignment is considered an advanced analysis tool. Outputs from these assignments are very powerful in determining which origin and destination pairs use the selected link and which routes they take. Selected links are available for vehicle assignments as well as public transport assignments.

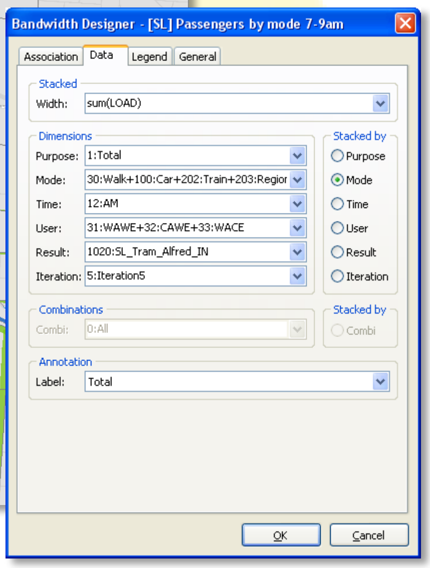
Considerations when using these designs

Make sure to edit the selected design to specify the required selected link in the result dimension of the 'Data'-tab. Usually, all selected links are stored in a result field from number 1000. The result will have been created prior to running the model (see Running a Select Link Assignment).



Example





Selected link Trams on St Kilda Road, inbound CBD

Available stacked designs

[SL] Passengers by mode 7-9am
[SL] Passengers by mode 4-6pm
[SL] Passengers by mode daily
[SL] Vehicles 7-9am
[SL] Vehicles 4-6pm
[SL] Vehicles daily

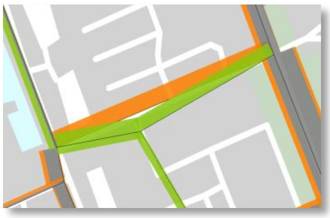
How to visualise a Comparison between Assignments

Description

In order to visualise changes in traffic or passenger flows between two variants, or even within a single variant, a comparison bandwidth visualisation can be used. This visualisation will show either an increase or decrease in traffic or passengers and the balance (or a 'shared' colour).

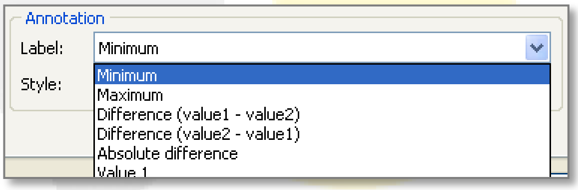
Considerations when using these designs

Select the variant to compare an assignment from the active variant with. It is possible to select the current variant here in order to compare two assignments within a variant, for example for a visualisation of traffic or passenger flows in different iterations.

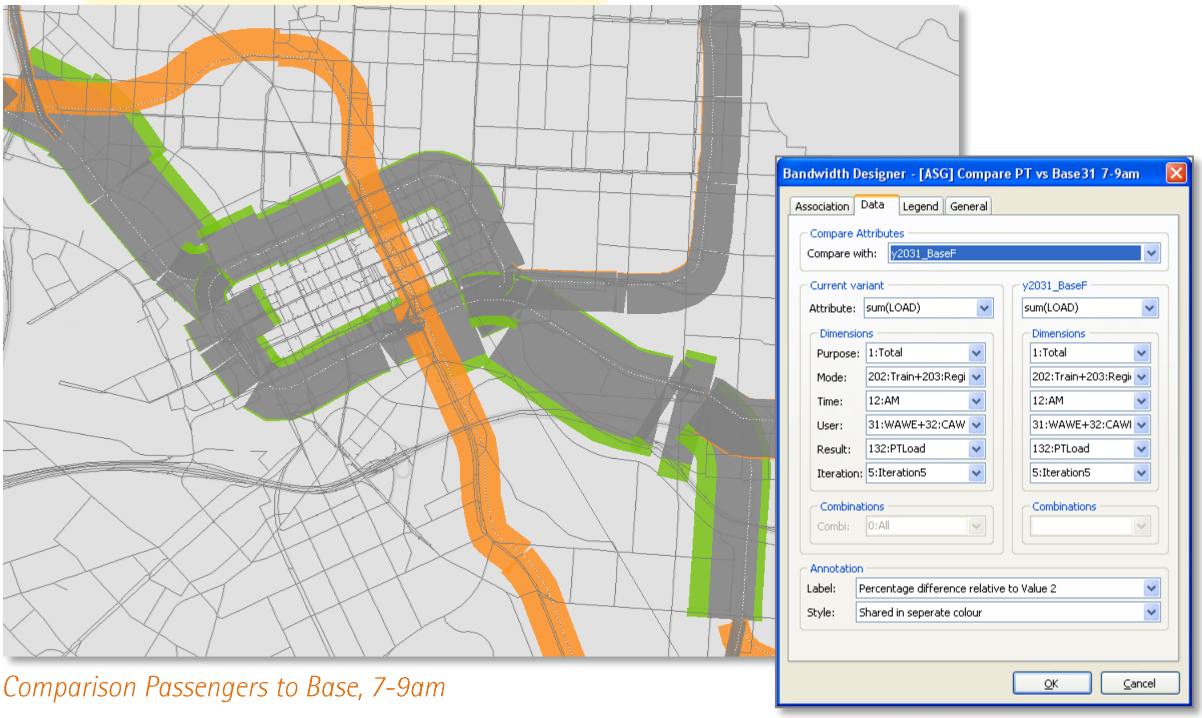


Comparisons between bandwidths are made on a link-by-link basis. This comes with the **side effect** of links showing either a full increase of full decrease colour whenever this link is not available in one of the variants and thus can't be compared. See the example to the right in which a link has been broken and moved slightly in one of the two variants which are compared to each other, green is an increase and orange is a decrease.

Also, when using these designs, a lot of **labelling options** are available. Aside from the usual absolute and percentual differences, a GEV value can be calculated and visualised.



Example



Comparison Passengers to Base, 7-9am

Available stacked designs

[MV] Trams 7-9am

	[MV] Trams daily [MV] Trains 7-9am [MV] Trains daily

How to visualise a Comparison between an Assignment and Counts

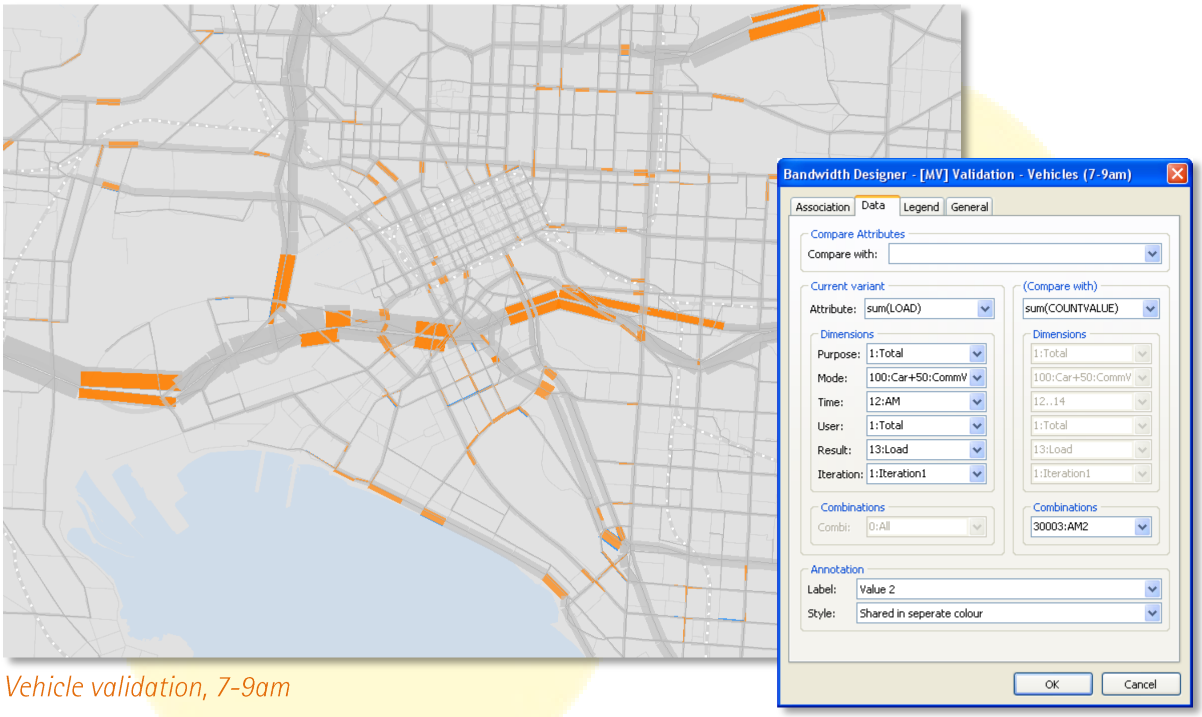
Description

A model represents real-world traffic flows within a simplified, but advanced, model. Amongst many validation tools, a heavily used visualisation tool is a comparison between an assignment and counts. This visualisation shows only validation on network level and should be used in conjunction with other tools, see also How to validate a Zenith Model <<TODO>>

Considerations when using these designs

Usually an assignment is compared against a countvalue. However, depending on the information available, some counts are saved as loads. In this case a regular bandwidth comparison would be used. See How to Visualise a Comparison Between Assignments.

Example



Vehicle validation, 7-9am

Available

[MV] Vehicles 7-9am
[MV] Vehicles 4-6pm
[MV] Vehicles daily

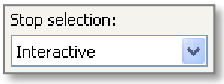
How to visualise Stop Interchange Information

Description
The Zenith Model stores detailed interchange information for at- and inbetween stop interchanging which can easily be visualised for usage in interchanging analysis.

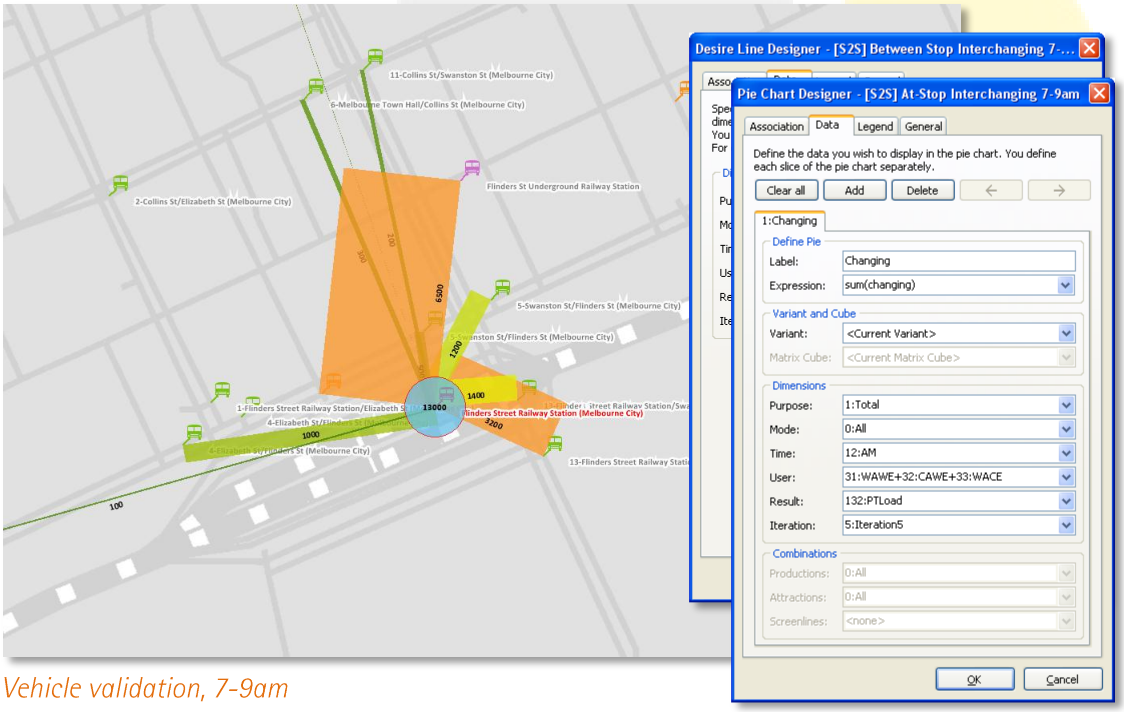
Considerations when using these designs


At stop interchanging is visualised using piecharts whereas interchanging between different stops is displayed using desire lines. In order to visualise all possible interchanges at and between stops, **both designs** for piecharts and desire lines need to be active!

Interchanges between stops is stored in the Zenith Model Database. In order to be able to view this data as desire lines, a station to station matrix has to be constructed using this database. In order to do this, **run the script** <<TODO>>



Instead of showing all interchanges within a single plot, it is possible to have this information displayed for a single only stop when it is selected. In order to achieve this, select **Interactive** from primary 'Stop selection'.

Example


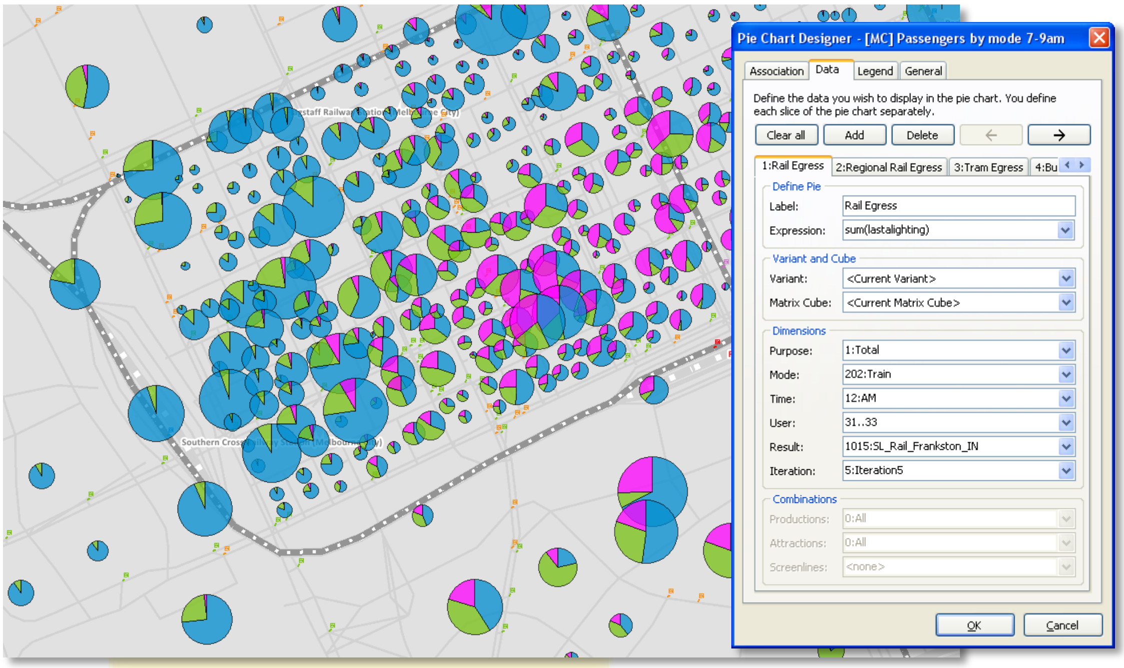
Vehicle validation, 7-9am

Available Stop Interchanging designs
[S2S] At-Stop Interchanging 7-9am
[S2Si] At-Stop Interchanging 7-9am
[S2S] Between Stop Interchanging, 7-9am
[S2Si] Between Stop Interchanging, 7-9am

How to visualise First or Last used Transit Modes

Description
Being able to visualise the first or last used transit modes allows for detailed analysis of transit mode choice either at the start or the end of a trip. This visualisation is especially powerful in combination with a selected link.

Considerations when using these designs
Additional modes or subdivisions of an existing group of modes can be made by altering the settings for the individual pies. The size of the pie represents the total number of passengers, the pies themselves show the mode share by transit mode.

Example


Passenger destinations by mode of last egress, 7-9am

Available Transit Mode Usage designs
[MC] Passengers by mode 7-9 am

9. Extra Information

The Zenith model is a four step travel model, which is developed and maintained by Veitch Lister Consulting. The Zenith model operates within the OmniTRANS transport planning software package, which is developed by Omnitrans International.

In order to successfully operate the Zenith model, the modeller needs a sound knowledge of both the OmniTRANS software and the Zenith model, and the relationship between them.

The remainder of this introduction will explain the relationship between Zenith and OmniTRANS, point you towards other useful sources of information, and explain how to use this manual.

1.1. The Relationship Between Zenith and OmniTRANS

OmniTRANS is a transport planning software package. It is a software platform within which transport models can be developed, stored, operated, and analysed. It provides a wide range of customisable tools, which enable the modeller to,

- store and edit modelling data, including transport networks, and demographic and land use data,
- manipulate and display model outputs such as trip matrices, skim matrices, select link matrices, traffic loads and passenger loads,
- execute modelling processes, such as trip generation, trip distribution, period choice, mode choice and traffic and transit assignments,
- create new and customised tools to perform any kind of modelling task

VLC has made use of these building blocks to build its Zenith model. Zenith is a specific implementation of a four step travel model, with a specifically designed set of model inputs, modelling processes, and model outputs. VLC has also developed a wide range of supplementary tools to assist the modeller in using the Zenith model, including tools to set up model run scripts, and tools for reporting, graphic display, import and export. These tools are integrated within OmniTRANS.

The relationship between OmniTRANS and Zenith is analogous to the relationship between Microsoft Excel and a complex spreadsheet. Excel is a software package (like OmniTRANS) which can be used to develop any type of spreadsheet. It comes with tools for developing spreadsheets, including formulas, graphing functionality, etc. In this sense, there are great similarities between Excel and OmniTRANS.

Zenith, on the other hand, is analogous to a very complex spreadsheet. Spreadsheets contain specific data, and specific formulas and graphs which manipulate that data in useful ways. The Zenith model is therefore analogous to a spreadsheet; it comprises an enormous set of useful data (transport networks, demographic and land use data, and behavioural parameters), and formulas which model human travel behaviour.

1.2. Other Sources of Information

There are a number of information sources available, which cater to users of different levels, and provide different types of information.

For information relating specifically to OmniTRANS there is,

- The **OmniTRANS Manual**, which provides information on all aspects of the software. This can be found under the OmniTRANS Help menu.
- **OmniTRANS Technical Documents**, found under the OmniTRANS Help Menu -> Technical Documents. Here you can find documentation on the OmniTRANS database, Ruby, etc.
- The **OmniTRANS forum**: http://forum.omnitrans-international.com/

For information relating specifically to Zenith there is,

- The **Zenith / OmniTRANS Training Course**, delivered by VLC
- **The Zenith Manual** (this document!) provides reference material for the Zenith model, and step by step instructions for a wide range of Zenith modelling tasks
- **The Zenith Technical Notes** provide detailed information about the development of the Zenith model, including its theoretical underpinnings and model validation. They can be found under the OmniTRANS Help menu, under Technical Documents -> Zenith.
- **The Zenith Model Assumptions Report** details the sources of information used to assemble base and future year models. This report can also be found under the OmniTRANS Help menu, under Technical Documents -> Zenith.

1.3. User Support

As a subscriber to Zenith User Support, you can receive help via email, telephone, and if necessary, on-site.

<u>Contact Details:</u>

Email: support@veitchlister.com.au

Telephone: (03) 9602 5200