Traxim

User Manual v5.0

August 26, 2026

Contents

1. Introduction to Traxim

1.1. Purpose of Traxim

Traxim is a rail network simulation program designed to increase the efficiency and accuracy of rail infrastructure planning and analysis.

The core of Traxim is the resolver. The resolver dynamically simulates trains given a network configuration and train plan, identifies occupancy conflicts between trains, and finds operationally feasible solutions to the conflicts. The end result is a fully deconflicted train timetable.

The automation of the timetable production process supports increased efficiency in four key network planning tasks:

- Strategic network analysis: Generate the data required for meaningful what-if analysis of complex rail networks.
- Master timetabling: Rapidly make master timetables with full wrapping at the end of the week (or other period).
- Daily timetabling: Replan the master timetable with daily cancellations, ad hoc additions, and temporary speed restrictions.
- Live run: Provide real time recommendations to train controllers, and project train arrival times for customers.

1.2. Structure of this Manual

Section 1 of this manual provides a primer on the principles and use of Traxim. Traxim was originally designed and built to support theoretical modelling. Using Traxim for theoretical modelling introduces many conceptual issues. This section therefore provides commentary on the uses and limitations of Traxim in network modelling. It also provides an explanation of the high-level functionality of Traxim, which is important for understanding the characteristics of the timetable it will generate. Finally, it offers an overview of creating and submitting a scenario for resolution.

Much of this discussion is theoretical. If you are eager to dive into the modelling task you can skim or skip this section and go straight to section 2.

Section 2 briefly describes setting up a Traxim simulation, covering the purpose and scope of the input files, and the required settings. While Traxim inputs are not especially onerous, there is a considerable amount of data required, and input formatting needs to be correct. This section provides an overview of the input files.

Section 3 explains the role and usage of the Network Editor, which is used to simplify the creation and management of the Infrastructure input file.

Section 4 describes the output files. This section might be most usefully covered once you have completed some simulations.

The Appendices provide straightforward descriptions of the structure and syntax of each of the input files. These are best used as a reference and should be read in conjunction with the process guidance in Section 2.

1.3. The Role of Theoretical Timetabling

Good decision making about rail infrastructure investment, or operating rules and practices, requires an analytical foundation, desirably including accurate quantification of the performance consequences of different options.

Many aspects of rail operations can be modelled using a spreadsheet and relatively simple formulas. The advantages of this approach are that it is simple, relatively quick, highly transparent, and it describes explicit relationships between assumptions, inputs, and outcomes that can be tested.

However, such “rule-of-thumb” tools necessarily risk oversimplifying complex problems. They struggle to deal with varying loop lengths, different train speeds and train priorities, peakiness, and the effects of following conflicts. They can only deal with relatively plain track – it is hard to assess junctions, intermediate signals, and so on. Some essential parameters, such as practical capacity utilisation, are arbitrary even though they may be derived from extensive practical experience.

They can only deal easily with a line section - there is limited ability to validate interactions between sections.

As such, they are best at allowing an analyst to gain a sense of how performance is likely to respond under different scenarios, and they allow rough “order of magnitude” quantification.

Another way of looking at this style of analysis is that it is essentially an inductive reasoning approach. It allows an analyst to build a hypothesis about how the infrastructure might perform under different configurations or operational patterns.

As with any hypothesis though, it is highly desirable that it be validated through a controlled experiment. Historically, this has been done by creating sample timetables based on the infrastructure / operational scenario.

Rail network analysts also sometimes adopt a deductive reasoning approach to infrastructure and operational issues. That is, they draw conclusions based on observation. This “observation” might be of current actual operations, or it might be insights gained in attempting to produce timetables for future scenarios. A good example is that an analyst might sum all crossing delays in a timetable to identify where on the network the greatest congestion is occurring.

Whether it is for validating a hypothesis or to gain insights by observation, the process of preparing and analysing timetables has two major shortcomings, particularly in the context of a rail network that doesn’t use a repeating-pattern timetable:

- It is highly labour intensive.
- It is usually impractical to generate enough sample timetables to provide statistical validity.

The primary objective of the Traxim simulation software is to mitigate these shortcomings by automating the timetable generation process. This eliminates the labour involved in timetabling, and by extension it allows a sufficiently large population of timetables to be created that it is practical to achieve statistical validity. It also creates a wealth of data that can be deployed to gain deeper insight into effects and their underlying causes.

1.4. Basic Principles and Philosophies of Traxim

1.4.1. Core Functional Elements

The core functional elements of Traxim are that it:

- Produces timetables that are reasonably optimised, but still realistic,
- Is based on a detailed (but not onerous) definition of the network, dynamically modelled trains, perturbed departure times, and true train priority, and
- Avoids deadlocking.

These concepts are explored in greater detail in the following sections.

Note that Traxim uses the terminology “train plan” to describe the set of train inputs, and “timetable” to describe that schedule once it is deconflicted.

1.4.2. Appropriate Uses and Limitations of Traxim

Traxim is most effective in analysing networks that are exclusively or predominantly for freight, or for passenger services that do not operate to a repeating ‘clockface’ pattern. While Traxim can preserve a preferred pattern of passenger operations by ascribing them absolute priority, a predominantly passenger rail network is likely to be better analysed in the context of an ‘intelligent design’ process and with software tools that are oriented toward such analysis.

1.4.3. Randomness

Traxim uses randomness as a mechanism to generate unique timetables.

The randomness is generated through perturbation of train departures. Train departure times are defined as a nominal network entry time and a tolerance, thereby creating a departure window. Actual modelled departure is randomly determined as a time within the window, applying a uniform distribution. The randomizer is seeded so that simulations can be replicated. Some trains can have a zero tolerance, and hence enter the network at a fixed time, if that is appropriate to the circumstances. This may be a modelling choice for passenger trains for instance. However, to create a population of unique timetables at least some trains require randomization. Generally, the butterfly effect is very strong, and even small amounts of randomization can create dramatically different final timetables, though this will largely be a function of the intensity of use of the network and the distance of travel of the train.

1.4.4. Optimisation

Traxim is not a timetable optimisation tool as such. It uses a complex system of weighting to make decisions as to which crossing solution to adopt from a menu of reasonable crossing options. While the algorithm is technically “greedy” in that it will always seek to minimize local delay in the first instance, it will in some cases hold an individual cross for longer than “necessary” when it encounters serious congestion, in the interests of finding, quickly, a feasible solution.

This is distinct from a pure optimization approach, which would seek to minimize aggregate delay for all trains looking across the entirety of their journey. Such an approach though, is generally considered to be computationally beyond current technology for large complex networks.

Since there is no way to create a perfectly optimized timetable to use as a baseline, the extent to which Traxim does optimize is unknowable. What is knowable is that differently perturbed timetables will achieve different levels of efficiency (in terms of less delay) than others with otherwise identical inputs. This isn’t exactly comparing like-for-like though – it only demonstrates that a scenario with different train start times can create a more efficient timetable, not that the crossing decisions themselves were better optimized.

What is certain though is that Traxim has advantages over a train controller, or even a timetable, in that it has highly accurate timing of train location, and the ability to consider large numbers of alternative solutions. This suggests that it is likely to typically outperform them in minimizing delay.

1.4.5. Relationship with Network Reliability

An important question for railway planners and operators is ‘how reliable will my performance be?’

The extent to which a railway is perceived to be unreliable will depend on the difference in delay between the planned timetable and the actual disrupted state. That is, if a railway is timetabled in a highly efficient way, with well-designed crosses and minimal delay, it would be expected to be relatively ‘unreliable’ when subjected to disruption. If the timetable is inefficient, with poorly designed crosses, lots of additional crossing allowances, and scheduled recovery time, it could be expected to be more “reliable”. Essentially, there is a direct trade-off between planned transit speed, and reliability.

Given the perturbation of train start times, Traxim is in a sense modelling a disrupted state.

In interpreting an analysis undertaken using Traxim, the typical timetabling practice of the modelled railway needs to be assessed. To the extent that timetabled delay is greater than predicted in Traxim, it would be expected that trains entering late, or experiencing problems enroute, would not incur any further delay and may even recover lost time. To the extent that the timetable has less delay than predicted in Traxim, trains that enter late or encounter problems are likely to, on average, lose further time.

It should be noted, in this context, that this makes a good case for the use of Traxim in the master and daily timetabling process, since the conditions and assumptions can then be controlled across actual and theoretical timetables, allowing reliability to be directly estimated.

1.4.6. Network Definition Detail

The objective of Traxim is to generate realistic timetables and it is therefore necessary to define the network in sufficient detail that the incidence and duration of conflicts is reasonably well represented.

However, it is intended primarily for high level planning, and for application to long distance, heterogeneous networks, as opposed to intensive commuter type networks, or complex signaling system design. The aim has been to define the inputs at a level that makes a sensible trade-off between accuracy and complexity. As rule of thumb, the inputs are simplified to the extent that the loss of precision is not likely to be material given normal natural variation in train performance. Technically, this can be described as a mesoscopic level of modelling detail.

1.4.7. Dynamically Modelled Trains

Dynamic train modelling is the application of physics to a defined train consist and track infrastructure / geometry to determine train speed and hence transit time.

To ensure performance accuracy and consistency within Traxim, trains are dynamically modelled based on their actual journey. This involves the dynamic simulation of the train when operating on its default path, and its resimulation whenever it needs to stop or slow to resolve a conflict, including when it is redirected off its default path, specifically to respect turnout speed limits.

Actual maximum train speed has been calibrated to be limited to 90% of maximum permissible speed once a train exceeds 40 km/h, which reflects observed real world practice. Speed on downhill gradients is also constrained based on a rule-of-thumb for acceptable downhill speed to avoid brake failure.

1.4.8. Nuanced Train Priority

In certain situations, one train, or a class of trains, usually passengers, will have absolute priority over all other trains. Traxim allows for the highest priority of train to be given absolute priority. When two absolute priority trains meet, the crossing decision will always be based on minimizing delay for that single conflict.

In other cases, classes of train will have a relative priority. The priority value used in Traxim is unbounded, allowing for high levels of discrimination between trains. Traxim multiplies predicted delay by train priority to get a weighted delay. It uses this weighted delay as the primary consideration in selecting a crossing option. Where there is congestion though, delay becomes interdependent between conflicts. Hence, the crossing option selected for any individual conflict might not minimize purely for that conflict – it will be aiming to minimize weighted delay across all interdependent crosses.

Train prioritization weighting is purely a matter for the modeler, but requires some care and consideration. The modeler may, for instance, want to give train B a fivefold priority over train A, so that train A will, for instance, be delayed up to 10 minutes to save train B a two-minute delay. If in turn, they want to give train C a fivefold priority over train B, it's important to be aware that this will give train C a 25-fold priority over train A. Hence, train A would wait up to 50 minutes to avoid a two-minute delay to train C, which may be excessive in practice. Getting the balance right may require generating and reviewing some test timetables to assess whether the prioritization schema is generating the intended outcomes.

1.4.9. Unresolved Scenarios

A fundamental problem of simulating train operations is the risk of deadlocking, that is, a situation arising where no train can move forward due to other trains blocking its path. Traxim is designed such that it will not get into a deadlock situation. It should never fail to resolve because of a deadlock.

In situations where the number of trains exceeds the capacity of the infrastructure, the software may not be able to find a solution. If the timetable is not being "wrapped" - that is, trains at the end of the week don't wrap back to the start of the week - the algorithm should find a solution, but with extensive congestion apparent at midnight on the last day of the simulation period. If wrapping is enabled, and the number of trains exceeds the network capacity, the algorithm will necessarily be unable to resolve the timetable and it will eventually time-out. In extreme cases, it may also time-out even if not wrapping, simply due to unmanageable congestion.

1.4.10. Simulation Completion Speed

Statistical theory suggests that to get a statistically valid answer requires a population of around 30 randomly generated timetables. The most simulation intensive task is the comparison of multiple competing infrastructure options, such as different combinations of new crossing loop locations. It would typically be reasonable to test up to four such options, and a "do nothing" base case. As such, a set of scenarios to analyse a network performance question might typically require the generation of up to 150 timetables (30 timetables each for 4 options and a base case).

Each timetable should ideally span a one-week period to get the full range of likely peaks and troughs, and the "run-in" period needs to be at least one day, and up to 3 days, for a total of 8 - 10 simulation days.

Traxim has been developed specifically to address this scale of analysis and generate results in a practically useful time period. The speed with which Traxim resolves a scenario will vary considerably depending on the size of the network and the intensity of operations. Experience suggests that a scenario can take anywhere between 5 seconds and 15 minutes to resolve. The elapsed time to complete a batch simulation will depend both on the complexity of the scenario, and on how many virtual machines are deployed to process the batch simulation. However, you should expect it to typically take somewhere between 10 minutes and two hours.

If you have specific requirements, or would like more detailed advice, please contact us.

1.5. Overview of the Traxim Modelling Process

1.5.1. Introduction

There are three essential elements to Traxim:

- Input data.
- The timetable resolution process.
- Simulation outputs and their interpretation.

1.5.2. Input Data

All of the data required to run a simulation is inputted in the form of CSV (Comma Separated Values) files.

All of the CSV input files are designed to be edited in Excel, or other spreadsheet program. A master version can be maintained in Excel format for convenience, but it needs to be saved using the CSV (comma delimited) option in the 'file format' drop-down when preparing it as the Traxim input file.

Chapter 2 describes each of the input files and their structures in overview. The appendices provide more detailed descriptions of all of the input fields. Sample files are available in the Resources section of the website.

To streamline editing of the Infrastructure file, a Network Editor is available as a support tool. This provides a graphical interface that allows visualization of the track configuration. Editing the Infrastructure CSV file directly can be efficient where bulk changes are required, such as renaming a region, but for most purposes it is highly recommended that the Network Editor be used.

1.5.3. The Timetable Resolution Process

Traxim is offered as a service.

The Traxim Controller Interface manages the process of defining and submitting a scenario. Validation and timetable resolution are undertaken using cloud based virtual machines. Aggregation (calculation of statistical averages) is initiated manually once a batch resolution is completed. Results are downloadable from the server as they are generated.

Full control of the timetable resolution process is available to AI agents using the MCP connector at https://tools.traximrail.com/api/agent/mcp A skill is also downloadable to assist the agent to understand the theory and practice of using Traxim.

In overview, the process is as follows:

1. Create, or update as appropriate, all of the input files to describe the scenario you wish to run. This User Manual has detailed instructions for each file type.

2. Once you have your scenario files, login to the Traxim Controller.

3. Upload your scenario input files and determine your settings in the Scenario Management panel on the left-hand side of the Traxim Controller. Save your scenario.

4. Select your scenario in the dropdown in the "Scenario File" field in the Simulation Management panel in the middle of the Controller. Click "Validation". The validation process will check for missing or logically inconsistent data. You can monitor progress in the Console Output field on the right-hand side of the Controller. Any input errors identified in the Validation process will be displayed in the Console Output. If there are errors, correct your input files, re-upload them, save the scenario, and rerun the Validation. Iterate until the Validation completes successfully.

5. Once you have a successful Validation, go to "Download Results" and download the unresolved timetable pdf. Please review that pdf to ensure that the scenario appears to be as you expected. If necessary, modify your inputs and rerun the Validation. Iterate until you are happy with the output.

6. When you are confident that your scenario is validating, and it is generating the expected results, submit the scenario for resolution. Click "Resolve (single)" to run one perturbation. Progress will be shown in the Console Output box. Again, review the resolved timetable to confirm that it is what you were expecting. For instance, you may want to check that your train priority

settings are giving reasonable crossing solutions. If appropriate, make changes to your scenario and regenerate the sample resolved timetable.

7. A single resolution run generates a train statistics csv file, and a timetable csv file, as well as the train graph. These can be used for multiple purposes. In particular, the timetable csv is a valid resolved timetable that can be used for train operations. The pass times in the timetable csv also provide the basic data required to calculate utilization per UIC-406.

8. If you require a statistically valid dataset, then once you are satisfied with your resolved timetable, submit the scenario for batch resolution. Select "Resolve (batch)". This will generate a batch of 30 perturbed timetables. Progress will be shown in the Console Output box.

9. Once the batch resolution completes, select "Aggregate Results". Aggregation progress will be shown in the Console Output. Once aggregation is complete, download the AggregatedResults:TrainStats file. This contains the key performance data to allow you to undertake statistical analysis.

This process is described in more detail in Section 2.

1.5.4. Simulation Outputs

Traxim generates and returns a slightly different set of files depending on the type of simulation as follows:

- Validation
  - TrainStats.csv
  - TrainGraph.pdf
  - TrainGraph.json.gz

- Resolve (single)
  - TrainStats.csv
  - TrainGraph.pdf
  - TrainGraph.json.gz
  - Timetable.csv

- Resolve (batch) plus Aggregate
  - Detail.detail for each perturbation
  - Aggregate.xml
  - AggregatedResults.TrainStats.csv
  - AggregatedResults.GoogleEarth.kml

The TrainStats files report key outputs for each train. They are reported unwindowed, and for the window period by region and in total. For the aggregated results, each of the statistics is the average of the population of perturbations. The metrics reported are:

- Delay (scheduled, unscheduled, and total)
- Raw transit time
- Total transit time (raw transit time plus delay)
- Total distance
- Number of conflicts won and lost, by type.
- Various energy statistics.

The Train Graph is available as a pdf, openable in any pdf viewer, and as a json.gz file. The json.gz version can be opened in the Train Graph Viewer. The Train Graph Viewer is able to hold proportions constant, and has controls for changing the sequence and direction of regions, allowing easier visualisation. It also has a pop-up that shows the train route through the network, allowing granular understanding of pathing.

The timetable.csv reports arrival / departure / pass times for nominated nodes.

The Detail.detail and Aggregate.xml files are intermediate outputs not designed to be used for analysis.

The GoogleEarth.kml file can be opened in Google Earth to show delay by location geographically.

Chapter 3 describes the outputs and the recommended approach to analysis in more detail.

1.6. Designing and Interpreting a Simulation

The process and considerations in designing a scenario to simulate vary depending on the nature of the issue.

1.6.1. Quantification of the benefit of competing scenarios

The benefit of competing scenarios can be assessed by running a series of simulations where everything is held constant other than the input you are wanting to test. Most obviously this would be the infrastructure configuration, but could also be differences in the train inputs, like priorities, power-to-weight ratios, or even network entry time patterns.

For each scenario, run a batch resolution to generate 30 perturbed timetables, and then the aggregator to produce the dataset of train statistics. More detail on the train statistics output file is provided in Section 4. It is recommended that you use a pivot table to analyse the data. The key metric to generate is delay per 100 km for the windowed period. This normalizes for variances that might cause trains to travel different aggregate distances in the window period. Using the window period ensures that you are looking at a like for like consistent steady state.

In designing your scenarios you may want to consider the following:

- The net effect of any change will vary depending on the train pattern. You may want to run multiple scenarios with different operating patterns, and different numbers of trains, as this may influence the relative outcome.
- A solution that is optimised for the immediate challenge may not be the best solution for the medium term. For instance, building a new loop in the middle of an existing section may solve a short-term problem. The next increment in capacity may, however, require two new loops to split each of the newly created sections. It may be preferable to have a medium-term plan to spilt a section in three. The first loop built under this approach may offer less benefit than splitting the section at the mid-point, but it gives the best medium-term outcome.

It is worth noting that in looking at competing scenarios, the issue is one of relative rather than absolute performance. Hence, data accuracy is not so critical, though obviously the better the data quality the more reliable the result will be.

1.6.2. Estimation of transit time

Traxim dynamically models the ‘raw’ running time for each train based on its characteristics, as well as calculating crossing delay and any other dwells, to give a total origin – destination transit time. All of the transit time elements are reported in the Train Statistics file.

It is recommended that to calculate an origin – destination transit time, you calculate average speed during the window period and multiply that speed by the route length. This corrects for the effect of the run-in and run-out period.

The key to accuracy in this calculation is the quality of the inputs, in particular speedboards and train configuration. If you have particular requirements around train performance, for instance to calibrate for driving style, please contact us.

As well as looking at the average transit time for any class of trains, you may also want to look at the range of transit times for each individual train. A high degree of variability in transit time suggests that the railway is operating at a relatively high level of utilisation and that the average transit time may need a relatively large buffer to ensure predictability in arrival time.

1.6.3. Assessment of capacity adequacy

There is no single correct answer as to capacity adequacy. UIC-406 provides a useful framework, the compression method, for how to calculate capacity utilization. It does not, however, provide an answer as to what the limit of capacity utilization should be.

Capacity utilization for a given scenario can be calculated from the arrive / depart / pass times in the Timetable output file generated with a single simulation.

The recommended approach to testing capacity adequacy using Traxim is essentially a stress test of the network, by running simulations with progressively increasing numbers of trains. Average delay per 100 train kilometres is the most useful quantifiable metric as an indicator of worsening congestion. A more brutal approach is to push the network to its limit. If timetable wrapping is off, utilization beyond the network capacity will manifest as a backlog of trains at midnight on the last simulation day. These trains will ultimately complete their journeys during the run-out period, but extensive dwell at the end of the period indicates a network beyond its capacity. If timetable wrapping is on, utilization beyond practical capacity will manifest as Traxim failing to be able to resolve. It is important to note that Traxim is designed to minimize delay, not maximise capacity, and in either case there is no way to be certain that a solution is optimal. As such, it isn't possible to guarantee that a failure to resolve a timetable represents an absolute limit of capacity: in particular, more intelligent design in the timetable inputs could extract additional capacity. It is, at best, an approximation, but nonetheless as reasonable one.

An important caveat here, is that, as discussed in Section 1.4.5, Traxim is likely to be able to outperform a network controller managing the network in real time. The capacity of the network is therefore also, in part, a function of the technology available for network control. Network capacity will be higher where an automated planning tool like Traxim is used for live run than where it is entirely dependent on manual processes.

The paper "NSW North Coast Line Capacity" on the Resources page of the Traxim website describes applying the stress test to a real world environment.

For capacity adequacy analysis it is important to ensure that all of the input data is as close to the real world as possible, since in this case the analysis is looking at absolute outcomes.

1.6.4. Identification of bottlenecks

The kml file aggregates and smooths delay by location for viewing in Google Earth. This allows you to instantly visualise where on your network you are incurring the greatest delay, which is likely to be where it makes most sense to focus your capacity enhancement or transit time reduction efforts. Google Earth allows you to open multiple KML files, so you can compare delay by location for a number of different infrastructure scenarios.

The key to analysing bottlenecks is to consider the train inputs carefully, in particular departure times. There is a risk that the choice of departure times can create artificial peaks that cause congestion around certain locations. By rethinking the train operating pattern, or simply giving greater latitude around the acceptable range of the departure windows, it may be that the apparent bottleneck is largely or completely eliminated.

Note that in the case of identifying bottlenecks, data accuracy is not so important since the key issue is one of relativity of delay across scenarios.

More detail on the kml file is given in Chapter 4.

2. Using the Traxim Controller

2.1. Introduction

This section provides a detailed description of the process of setting up and running a simulation. This is done through the "Traxim Control Interface" (just referred to as the Traxim Controller here) at https://tools.traximrail.com/traxim_controller.html.

Please note that to resolve a scenario you will be required to login to the Traxim Controller. Logging-in ensures the privacy and accessibility of your scenario submissions, results, and payment arrangements. We have tried to make account management as painless as possible. Simply providing your email address creates your account. Access to the account is via a "magic link", that is, just an email to the email address you use for logging-in with a link button to confirm ownership of the account. Once you have logged-in, you will be able to re-access, edit, and resubmit your scenarios as often as you need. Each login remains active for one week, after which time you will need to request a new magic link.

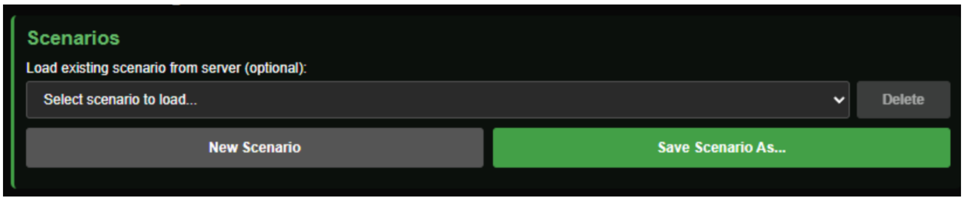
Traxim is currently free during testing and ongoing product development. Beyond this beta phase, simulations and the train graph outputs will remain free. Payment will be required to download statistical outputs. Pricing will be based on the number of conflicts resolved, to reflect the scale of the simulation task.

The controller is split into three panels, "Scenario Management", "Simulation Management", and "Console Output". This section will describe how to use each of these in turn.

2.2. Scenario Management

2.2.1. Scenarios

The scenarios sub-panel is as follows:



The basic input and output structure is built around a scenario (.scn) file. This simply records the locations of the input files, and the scenario settings. When a scenario file is saved, the system creates an associated folder where the input files are stored, and future outputs will be saved. All of the inputs and outputs are saved within your user directory, and only visible to you.

To modify a scenario, including loading an updated version of an existing input file, select the scenario from the dropdown. This will populate all of the scenario input fields. This feature can also be used to check the files and settings of a previously saved scenario.

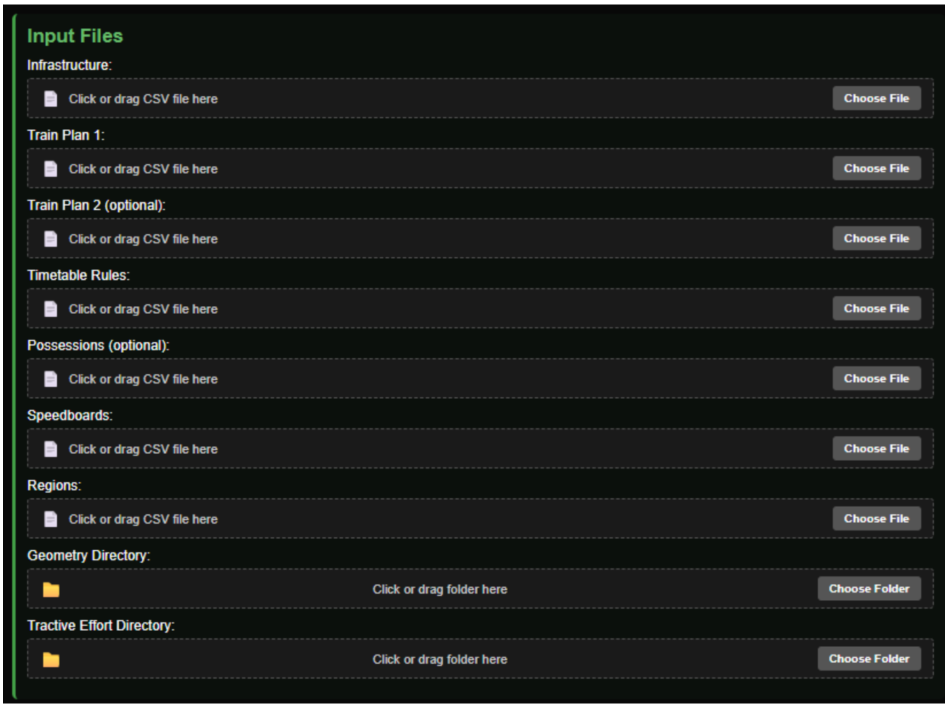
If you want to clean up your scenario files and folders, load a scenario from the dropdown and click "delete" to remove it. Note the scenario file will be deleted immediately. The folder will be deleted during the next clean-up cycle.

Clicking the "New Scenario" button will clear all the input fields.

Once you are happy with the inputs and settings for your scenario as per sections 2.2.2 and 2.2.3, return to this section and click the "Save Scenario As …" button. This will open a dialog window where you can enter a scenario name. Choose something that is meaningful to help you remember and locate it. If you choose the name of an existing scenario, the existing version will be overwritten. Note that overwriting an existing scenario will create a new folder for the files. The old folder will be deleted during the next clean-up cycle. If you have files that you want to keep, make sure you download them before you overwrite a scenario file.

2.2.2. Input Files

The input file panel is as follows:



This panel is used for uploading files, either by dragging or by selecting from the Windows picker box. The following is an overview of each of the files:

Infrastructure

The definition of the infrastructure is the most complex of the Traxim input files. The infrastructure input to the simulation is a CSV file but due to the need to be able to visualise the network to efficiently edit it, we provide a Network Editor tool that allows the network to be displayed graphically. The tool is accessible from the Resources page of the website. The use of this Network Editor is described in detail in Appendix 3.

The Traxim infrastructure configuration is primarily defined in terms of turnout nodes. There are also secondary node forms to represent intermediate signals, and end node points.

A turnout has three branches, “facing”, “through” and “diverge”. An intermediate signal implicitly has two branches, both “through” in nature. An end point has only one connection, to the facing branch.

Branches are connected to each other by links. In network systems theory, these are also called edges.

Each turnout is located in space by its kilometer posting and nominated track geometry section. That is, it is referenced to a latitude, longitude and elevation by matching its kilometrage to the corresponding kilometrage in the track geometry file for the region. Note that since it has three branches, it can reference up to the three track geometries and kilometrages.

Train Plan

The purpose of the Train Plan input file is to define the characteristics of the trains to be simulated, including their entry time, and entry and exit locations on the network.

The Train Plan file specifies a range of parameters including train name, length, weight, locomotive types and train priority.

A full description of the Train Plan input file is at Appendix 7.

Rules

It is frequently desirable to specify a series of events to occur along the journey of a train. The primary ones are that it must pass a specific location (eg a platform), that it must not pass a specific location (eg due to outline gauge limitations) or that it must dwell for a defined period / not depart until a specified time.

The Rules file works in conjunction with the Train Plan file to define these events. It is separate to the Train Plan file for ease of data management.

Detail on this file is provided in Appendix 6.

Possessions

The possessions file specifies a period during which a link, and the nodes at each end of the link, are unavailable. This input is optional.

Speedboards

The speedboards file is a list of speedboards by region and kilometrage. Speedboards need to be defined by direction. An unlimited number of speedboard categories (eg, freight, passenger) can be defined for each direction, allowing different classes of trains to operate to different maximum speed limits.

More detail is provided in Appendix 4.

Regions

The primary purpose of the regions file is to integrate the regions used in the infrastructure, speedboards and track geometry files. It also provides a filtering option for analysing results.

The Regions file:

- Designates a colour for each region, which is used in the pdf of the train graph (though not in the web viewer).
- Specifies which order the regions will appear in on the train graph.
- Specifies which direction of travel will be used on the train graph.
- Provides a mechanism to specify region specific transaction delays to be applied when there is a following or opposing conflict.
- Specifies the default values for super-elevation and cant deficiency when applying auto-speed limiting.

The regions input file is described in more detail in Appendix 1.

Geometry

Curves and gradients are defined by a track centreline file specifying latitude, longitude, elevation, and nominal kilometrage.

The track geometry file is used primarily for dynamically simulating trains. The latitude and longitude, rather than the kilometer points, define the distance to be travelled by the train. Elevation data is used to generate the gradient. The nominal kilometrage cross-references the infrastructure file so that nodes can be positioned relative to the track geometry.

The track geometry file is described in detail in Appendix 2.

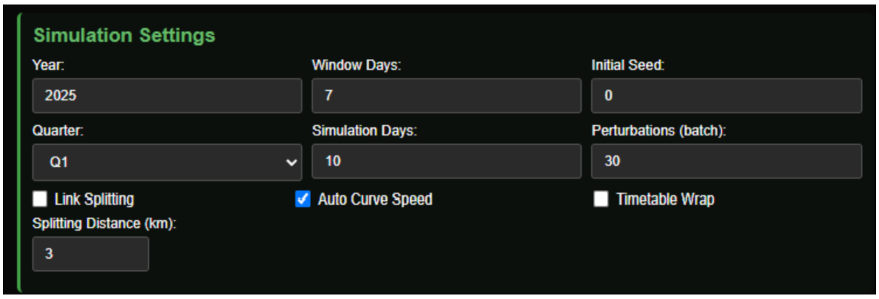
Tractive Effort

Each locomotive type used in the train plan input file needs a number of fundamental performance characteristics to be defined. This information is held in the tractive effort file for that locomotive type.

The tractive effort file is described in Appendix 6.

2.2.3. Simulation Settings

The simulation setting panel is as follows:



The purpose of each of the settings is:

Simulation Year and Simulation Quarter

Traxim allows inputs across time to be simplified by specifying within the Train Plan what time periods each train is active for. For each train, the Train Plan has required cells of Start Quarter and Start Year. There are optional cells for End Quarter and End Year.

The Simulation Year and Simulation Quarter defines the point in time that is going to be simulated. Only trains that are active at the simulation date, as defined in the Train Plan input file, will be simulated.

This structure simplifies management of the Train Plan input file where you have expectations of additional or changed services over time, by allowing all trains to be kept in a single file.

Window Days and Simulation Days

At the start and end of the simulation there is a run-in and a run-out period.

The run-in period arises because in Traxim all trains start from their origin rather than at a point on their journey. Accordingly, there is a period when they will not be encountering opposing trains and they will be running earlier than they would in a steady state environment. The period it takes to reach a steady state depends on the frequency of trains and the length of journeys. Most networks will require a run-in period of between 1 and 3 days.

Likewise, during the run-out period, trains are completing their journeys, but no new trains are entering the network, so the number of conflicts is reduced relative to the steady state environment.

Data generated during the run-in and run-out periods is not statistically valid for any analysis where absolute performance is important. If the analysis is only to compare options, inclusion of the run-in and run-out data is usually neutral.

To allow only steady-state data to be included in the statistical outputs, Traxim allows for specification of a window period.

The number of simulation days defines the time period that Traxim will generate train departures. Window days defines the period that Traxim will use for calculation of performance results.

The difference between “simulation days” and “window days” defines the size of the run-in period, with the window period ending at midnight on the last simulation day and windowing data for the specified number of days prior. For instance, if the window days is set at seven days, and a three-day run-in period is required, the simulation days would need to be set to 10.

For most scenarios a one-week period is appropriate as it will capture a full cycle of trains. For some more intensively used networks that have little variation in train numbers and relatively short journeys, a shorter window period may be adequate.

If your analysis is dealing in comparative data only, you may want to set the window and simulation days to be the same. This can remove the need to normalize data, which is discussed in more detail in Chapter 4. However, there is little upside in doing this.

Note that Traxim also has a timetable wrapping option, discussed below. In this case the Window Days determines the timetable period and the Simulation Days is overwritten to match.

Initial Seed

When creating a population of timetables, the train entry times are randomly perturbed within the start window using a random number generator. The initial seed sets the seed value for starting the random number sequence. This allows replication of simulations. Circumstances where you might want to change the initial seed are rare, but can arise.

Perturbations
The number of perturbations is the size of the population of timetables that you want to generate. The default is 30, which in most cases is a sufficient number to give reasonable statistical validity. At the Scenario Submission point, there is an option to choose either a single or batch simulation. Choosing single simulation will override the number of perturbations – it isn't necessary to enter 1 here.

Link Splitting
The purpose of link splitting is to emulate virtual block signalling. When on, Traxim will split each section between nodes into virtual blocks equal to the link splitting distance.

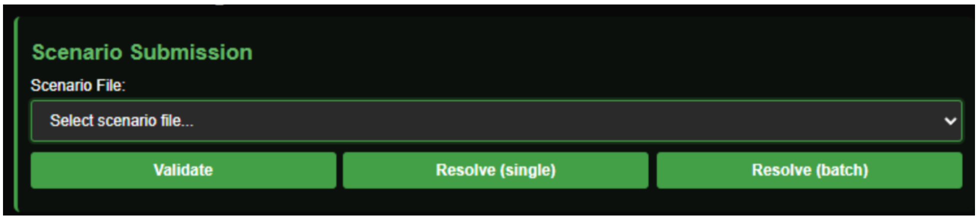
Auto Curve Speed
Traxim calculates curve radius, and when combined with a nominal super-elevation and nominal cant deficiency as specified in the Regions input file, will calculate a maximum safe speed at each geometry point. Turning this setting on automatically limits train speed based on this calculation. This is a useful functionality, especially for new conceptual alignments.

Timetable Wrap
Traxim has the capability to wrap a timetable at the end of the week (or other nominated period). That is, it will timetable trains that are running at midnight on the last day to continue their journeys at 00:01 on the first day. It should be noted though that this is technically challenging, and some compromises need to be made on train priority enforcement. To generate master or daily train plans it works very well. For statistical analysis it is probably better to not wrap the timetable.

2.3. Simulation Management

2.3.1. Scenario Submission

The scenario submission panel is as follows:



Scenario File

As a first step, select the scenario that you want to run using the drop-down. Note that this acts independently of the scenario dropdown in the Scenarios panel.

Validate

The validation process will check for missing or logically inconsistent data. If there are problems, they will be described in the Console Output.

In some cases, it can be difficult to identify the precise source of the error. For instance, if there is no pathway from a train's origin to its destination, it is unlikely to be immediately apparent how to fix that problem. If you are unable to identify the source of the problem, please contact us and we may be able to investigate further.

Note that the validation process steps through the inputs in a logical sequence and will generate an error report after each stage of the validation process. Since in many cases it can't proceed to the next stage until the problems are resolved, the validation will terminate at the end of a stage that has generated errors. Accordingly, you may need to iterate through a number of rounds of validation before all errors are identified and corrected.

Once you have a successful validation, an unresolved timetable pdf and json.gz will be available in your outputs folder. Please review that train graph to ensure that the scenario appears to be as you expected. Note that what may be an input error in terms of the objectives of the simulation won't necessarily cause a failure to resolve.

A detailed explanation of the train graph is given in Chapter 4.

Note that the resolve process won't initiate unless the validation process has been completed successfully. Any change to the inputs will reset the validation flag, so even if you are making a minor change after a successful validation it will need to be rerun. Validation is, however, generally very fast.

Resolve (single)

When you are confident that your scenario is generating the expected results, submit the scenario for resolution.

It is best to run a single perturbation before starting a batch perturbation so you can review the resolved timetable to confirm that it is what you were expecting. For instance, you may want to check that your train priority settings are giving reasonable crossing solutions. Trains overtaking, or not overtaking, are often worth considering to assess whether your priority schema is appropriate. To the extent you can, check that trains are using, or not using, the infrastructure as intended. Common errors are loops not being used (or being used when they shouldn't be) because they are the wrong length, and trains not preferencing the desired route on multi-track sections, which is hard to identify, but often results in delays at unexpected locations. A pdf and a json.gz of the Train Graph will be available in your downloads folder.

If appropriate, make changes to your scenario, revalidate, and regenerate.

A single perturbation will generate a timetable and train statistics output files.

Note that the validation process will calculate the number of conflicts for billing purposes. A conflict is defined as two opposing trains seeking to occupy the same section of track at the same time. A conflict ultimately manifests itself as a dwell. The number of conflicts will be a function of both the number of trains, and the intensity of section utilisation.

Resolve (batch)

Once you are satisfied with your resolved timetable, and if you are wanting a statistically valid data set, submit the scenario for batch resolution.

A batch resolution will generate 30 perturbed timetables. Note that it won't preduce individual train graphs are timetable output files.

2.3.2. Simulation Outputs

The simulation outputs panel is as follows:

Outputs Folder

The outputs folder populates from the selected scenario file. It shows the directory path so you can, if it's helpful, keep track of the location of files. It doesn't though have a functional purpose.

Aggregate Results

Aggregation is a key feature of the statistical approach to scenario analysis. Clicking "Aggregate Results" initiates the aggregation process. Progress will be shown in the Console Output.

Download Results

Clicking this button calls a folder window to allow selection of results files to be downloaded. Note that the file naming convention for results files is as follows: unique scenario hash_scenario name_seed_iteration count.file type. For the validation process, seed) is used and, as it is not resolved, the iteration count will also be 0.

3. Using the Network Editor

3.1. Introduction

The 'Infrastructure' file is a csv file that describes the physical layout of the network to be simulated. The way the network is necessarily described in the csv is non-intuitive. For this reason, the Network Editor has been developed to allow straightforward visual network editing.

The way the infrastructure is configured drives a number of the decision-making processes for the simulation. This section therefore provides some detail on how the Traxim algorithm makes decisions and how choices around the infrastructure configuration will influence the simulation outcomes.

This section describes the structure of the Network Editor and the various properties of nodes. Section 5 provides detail on the syntax of the Infrastructure input file itself.

The Network Editor has a short summary of the functionality of the tool, revealed by clicking the "How to" button.

3.2. Basic Principles

The infrastructure in Traxim is described in terms of nodes and links. Nodes represent:

- Turnouts, including an assumed set of associated signalling.
- Intermediate signals between turnouts.
- End points, being the end of a track.

From each node radiates 3 links in the case of a turnout, 2 links in the case of an intermediate signal, and a single link for an end point.

Each node is identified by:

- A unique name.
- The region (up to three, one for each branch of the node).
- Its kilometrage within the relevant region.

The primary purpose of the regions is to attach a location to a track geometry and speed limit. Where the node represents a junction between multiple regions it will have a kilometrage for each region. A single node can therefore have up to 3 kilometrages.

Where the node represents a turnout, it has 3 connection points. These are designated:

- F – For the facing branch of the turnout.
- T – For the trailing or through (non-diverge) branch of the turnout.
- D – For the diverge branch of the turnout.

In the case of a node representing an intermediate signal, the T and F connection points are used, though these descriptors have limited practical effect.

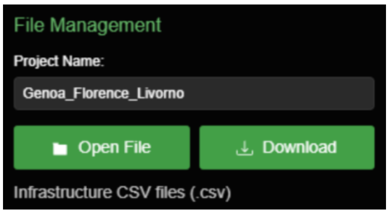
In the case of an end point, only the F branch can be connected.

A link joins the connection points on the branches of a node. When established, a link defaults to being bi-directional. It can then be modified to be uni-directional in either direction.

3.3. Controls Panel

3.3.1. File Management

The file management section of the controls panel is as follows:



Project Name

Project name loads automatically when you open an existing file. When a file is downloaded, it is used to write the file name. If you want to change the name of your project, you can overwrite the name in the Project Name text box and it will update the next download of the file.

Open File

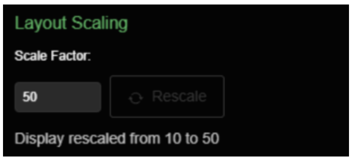
Open file will open a Windows picker box. Navigate to where your infrastructure file is located and click to open it. The infrastructure should appear in the main screen within a few seconds.

Download

For security reasons it isn't possible to save a file directly to your local drive. Selecting Download will download an infrastructure.csv file to the downloads folder on your C: drive. The file name will be project name_date(yyyy-mm-dd)_time(hhmmss).csv

3.3.2. Layout Scaling

The layout scaling control panel is as follows:



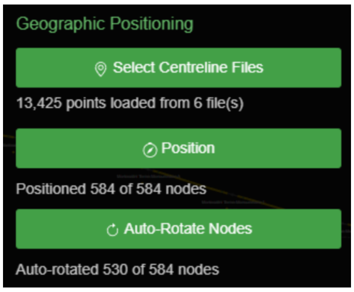
The scaling of the network is necessarily a compromise. If the scale is too small, nodes, which are a fixed size, will overlap and be indecipherable. If the scale is too large, nodes that are a long way apart won't appear on the screen at a sufficiently large size for them to be able to be linked.

On opening, the scale factor is set to 10. This is a good compromise, though preferenced toward long sparse networks. If your network has a lot of closely spaced nodes, you will need to increase the scaling factor. A scale factor of 100 is often required for dense areas like yards. At the other extreme, if you have a large network with loops a long distance apart, you may need to reduce the scaling factor to around 3 or 4.

To rescale, enter the Scale Factor value you want to use and click rescale.

3.3.3. Geographic Positioning

The purpose of the geographic positioning tools is to place nodes at their correct geoposition and rotate them to an orientation that aligns them well with adjacent nodes. The geographic positioning controls panel is as follows:



The mechanics of the geopositioning are that the kilometrage value of the node is interpolated between the latitude and longitude of the track geometry points either side of it. Hence, it is necessary to load all of the track geometry files to initiate the process. To do this, click “Select Centreline Files”. This will open a Windows picker box. Select the relevant files and click “Open” in the picker.

Once the files are loaded, click on “Position”. The nodes will then position themselves to their coordinates.

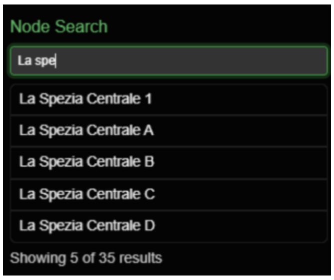
The third step is to orient them. Click on “Auto-rotate Node”. This will rotate the nodes so that the angle of the T and F branches to their adjacent connected nodes is minimized. The angle is weighted by the distance to the next node so that if one node is close by and the other is a long way away, the angle to the close node should be small, allowing for good, intuitive, alignment.

While this mechanism provides a reasonable alignment of nodes to their real world orientation, this does not always necessarily follow. This is particularly true there is significant curvature between a node and its next connected node, and also sometimes where node spacing is very close.

The intent of the geographic positioning function is to provide a starting position. Manual adjustment of both absolute position, and orientation, is often desirable to achieve a good representation of the actual track configuration. Note that neither setting is functional in the timetable generation process – they exist for visualization only.

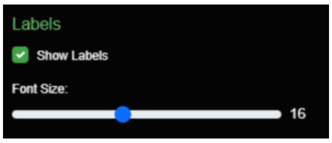
3.3.4. Node Search

Nodes can be found by searching for them by name. Enter part or all of the node name in the search box and the first five matches will appear. Click on the node to center the screen on that node. The search is not case sensitive. The Node Search panel is as follows:



3.3.5. Labels

Node labels, including kilometrage by region, are turned on by default. To make viewing of complex junctions easier, they can be turned off. Font size can also be changed using a slider. The Labels panel is as follows:

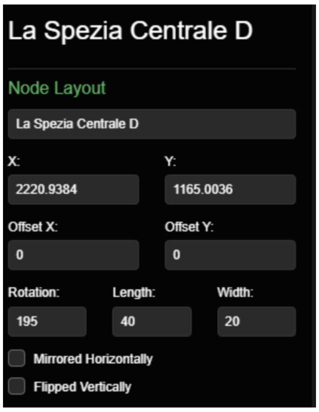


3.4. Node Properties Panel

The nodes area is where the Infrastructure Editor mainly functions from. The function of the fields within this panel are as follows;

3.4.1. Node Layout

The Node Layout section of the panel deals with the visual characteristics of the node, that is, its name, position, and size. The section appears as follows:



Name

The node name is a unique name allocated to the node. The node name will turn red if it duplicates another node name. There are no constraints on node name choice.

X and Y Position

These values describe the position of the node on the canvas as a conversion of the actual latitude and longitude of the node based on its kilometrage. Although they are editable, they are mainly shown for reference. They are best populated using the Geographic Positioning tool. It should be noted that latitude and longitude are derived from the track centerline. This means that all nodes, when initially positioned, are on a single linear string. In reality nodes will be offset perpendicular to the centerline, which is addressed with offsets below.

Offset X and Offset Y

These values represent the difference between where the node is actually sitting on the canvas, and its latitude and longitude based on kilometrage. When you drag a node, it will change the offsets rather than the X and Y coordinates. To move a node, simply click and drag. The offset values can also be edited manually.

Rotation

This field reflects the rotation of the node, in degrees, as it appears on the canvas. A node can be rotated by clicking on it and dragging the rotation handle that appears as a green circle. The value of rotation will auto-populate as the node is rotated. The rotation can also be set manually in this box if you want to get a precise rotation for the node such as horizontal or vertical.

Length

Length specifies the longitudinal size of the node drawing box. The default is 40, which roughly equates to 40 meters. This value has no meaning in the resolver algorithm – it is only for the purpose of optimizing the display in the Network Editor and the Train Graph Viewer.

Width

This value determines the lateral size of the node drawing box, which is effectively the separation between the T and D branches of the node. As for length, this is for visualization purposes only. The value defaults to 20. This approximately equates to 20 meters, which is unrealistic in reality, but is a comfortable setting for loop turnouts. For crossovers and complex yards, a smaller value can sometimes make for a clearer visual representation. Note that this value is also used in the Train Graph Viewer, and getting a good representation of the network in that tool requires care when setting the width values here.

Mirrored Horizontally

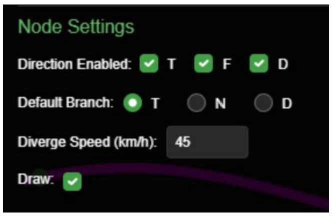
This is the equivalent of flipping the node (that is, swapping which side the diverge leg angles toward) and rotating 180 degrees.

Flipped Vertically

This swaps which side the diverge leg angles toward.

3.4.2. Node Settings

The Node Settings section of the panel deals with the functional characteristics of the node, that is, key inputs to the algorithm. The panel is as follows:



Direction Enabled

Whether a direction is enabled determines whether a train can enter a link from that branch of the turnout. This essentially sets the directionality of the track. The setting is represented on the canvas by a green or red dot positioned immediately beyond the node, and centered on the relevant link. The settings here can also be toggled by clicking on that dot. All directions are enabled by default.

Default Branch

Each turnout needs to be assigned a default branch, that is, the leg of the turnout (through or diverge) that trains will choose by default if there is a choice.

Traxim uses a concept of a default path to assist with efficient simulation and realistic replication of actual rail network operations. When the simulation initialises, a default path is identified for each train. This is the path that gets the train from its origin to its destination using the least number of non-default turnout branches. To take a simple case, the default branch of a turnout to a loop will be the through leg. This means that the train will normally take the mainline rather than the loop line as its normal route.

However, in the case of the start of a double track section where the normal direction of travel requires a train to take the diverge branch, the diverge branch should be set as the default. Similarly, if the double track was bi-directional with cross-overs, then it would be appropriate to set the divergent leg of intermediate trailing cross-overs to the default so that trains that find themselves travelling in the non-standard direction prefer to cross-over back to the “normal” direction track.

Note that in the timetable resolution process, trains will attempt to use the default track in the first instance, but if they encounter a conflict they will then try the non-default turnout leg (assuming that the directionality permits this).

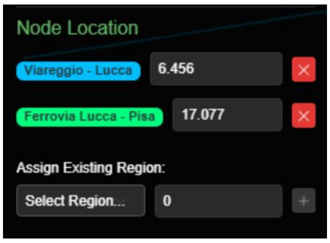
There is also an option to set a turnout to neutral, though the circumstances in which this is appropriately used are uncommon. With a neutral node, Traxim will be indifferent between the two route options. The default path will use the through branch but no penalty is imposed when the diverge branch is considered for resolving a conflict. It should be used where trains have a choice of track and in real life there is no preference between them. A good example might be where there are multiple platforms at a station and you want your train to automatically select an unoccupied platform. It is also good for multiple track sections where more than one track is available for trains travelling in the same direction. If a following train catches-up to a train in front, a neutral node at the diverge will tell the algorithm that the following train should take the diverge route without any penalty.

Draw

This tick box flags that the node should be drawn on the train graph. It will also set it as a timing point when exporting a csv of the resolved timetable.

3.4.3. Node Location

The Node Location section deals with the relationship of the node to the track geometry. It is where a Region is assigned and an associated kilometrage within that region. As each node has three branches, up to three Regions and their associated kilometrages can be assigned. The panel is as follows:



To add a region to the node, select a pre-defined Region by clicking the “Select Region” box. This will bring up a list of available regions. Click the region you want to add. Then enter the appropriate kilometrage in the kilometrage box. Click on the + to add the details to the node.

If you want to change an existing kilometrage, simple enter the value and click outside the box.

Any kilometrage value can be entered into the kilometrage box. If the kilometrage entered is logically inconsistent with the kilometrages of connected nodes, the links to those nodes will turn red and an error will appear in the error warnings box in the bottom right of the screen.

If the kilometrage falls outside the range of the track geometry file for the Region, and you have loaded the track centerline files on the Geographic Positioning section, this will generate an error in the error warnings box. It will also cause an error during the validation process.

Traxim is intended to be a realistic but simplified version of reality and to this end each node is given a single distance for the relevant region. Determining the kilometrage to apply to a node has numerous consequences.

Traxim will locate each node relative to the kilometrage values in the track geometry file, interpolating between kilometrage values as required to determine the latitude and longitude of the node.

Where a node has a position within more than one region, those distance reference points within the track geometry files are implicitly at identical latitude and longitude. However, if there is a variance, Traxim will tolerate that. The simulation will effectively jump between the pair of latitude / longitude points. Small discontinuities are unlikely to have a material effect on the simulation.

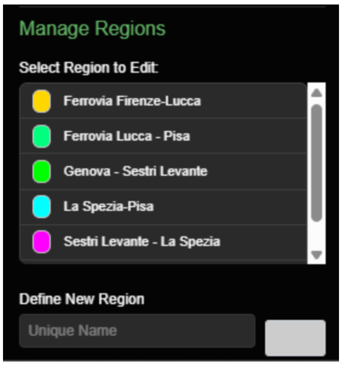
Consistent with the approach of a simplified infrastructure, the turnout and all associated signals are assumed to be at the single kilometrage point. This approach raises some challenges when determining the most appropriate kilometrage to allocate to the node, particularly in complex junction areas. The recommended approach is:

- Where a node is a turnout at one end of a loop, use a kilometrage that reflects its clearance point within the loop. This ensures that it correctly determines whether a train of a given length will fit within a loop.
- Where a node is one end of a crossover on double track, use a kilometrage that reflects the signal location in the facing direction.
- Where a node is a junction, give it a kilometrage that reflects the midpoint of the turnout itself. Use this approach even if there are multiple regions joining at the turnout. As noted above, the simulation will simply jump the train between latitude / longitude points if there is a difference between them in the different track geometries. As such, nothing is gained by intentionally using inconsistent kilometrages across track geometries.
- Where a node is an intermediate signal on a single-track CTC or track block system, use the physical signal location. This will slightly underestimate the time it takes for the train to clear the section, since the track circuiting would normally be some distance into the section. However, it isn’t possible to mitigate this without exacerbating the effect for trains travelling in the reverse direction.
- Where a node is an intermediate signal on a double-track uni-directional CTC system, giving the node a kilometrage that reflects the track circuiting will most accurately emulate the section clearance process.
- Where a node is an intermediate signal on a double-track bi-directional CTC system, or it is a signal that only applies in one direction on single-track, the best choice of kilometrage will depend on the circumstances. Note that it isn’t possible with Traxim to model a level of detail where there are different signal arrangements for the different directions on a single track. It will treat each node as a signal that applies equally in both directions. If the track has a “normal” direction it may be most advantageous to model the intermediate signals as if it was uni-directional. If there is no “normal” direction it may be best to approximately average the signal locations, or to position the nodes to best emulate the direction that is the most demanding. For instance,

where a single track is on a steep grade it may be preferable to represent the signal configuration for trains travelling uphill.

3.4.4. Manage Regions

The Manage Regions section of the panel allows you to add, delete, rename, and assign colours to regions. The panel is as follows:



To add a region, simply type its name in the “Define New Region” box and select a colour from the pop-up by clicking on the colour box. To edit the name or colour of a region, click on it in the list. To delete a region, click on the X adjacent to its name when it is selected on the list.

3.5. Network Editing Actions

The following actions are available to edit the network within the Network Editor window:

Zoom in and out: Use the scroll wheel of the mouse.

Select a node: Left click on the node.

Unselect a node: Click on another node, or on a blank area.

Move a node: Left click and hold on the centre of the node, then drag.

Rotate a node: Left click and hold the green dot, then drag.

Link a node to another node: Left click and hold on the anchor circle of the node branch you want to connect from, then drag and release on the node branch anchor you want to connect to.

Add a new node: Left click and hold on the node branch anchor you want to extend from, drag away and then release.

Delete a node: Select the node and hit the delete button on the keyboard.

Break a link: Click on the relevant node branch anchor point, or click on the red X that appears when hovering over the link.

Enable / disable link directionality: Click the red / green circle at the end of the link that you want to enable / disable. Clicking the circle toggles between on and off, that is, green and red.

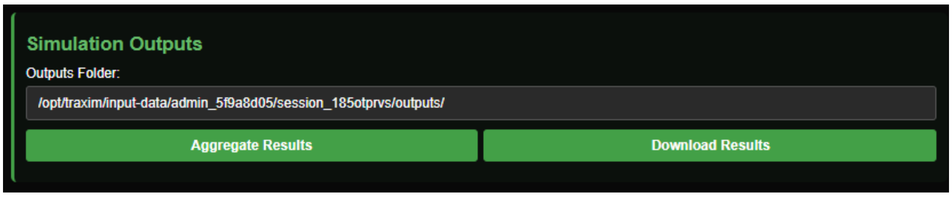
4. Simulation Results

4.1. Introduction

Traxim generates four output files:

- The train graph, which is a representation of the timetable in visual format, outputted as a pdf, and as a zipped json viewable in the Train Graph Viewer.
- The Train Stats csv file, which gives data on train performance for each individual instance of a train, focused on transit time and delay.
- The TrainTimetable csv file, which is a listing of arrival / departure / pass times for every train for nominated locations.
- The Google Earth kml, which provides a visualization of delay viewable in Google Earth.

Once a batch simulation is finished the aggregation process for the Train Stats csv and the Google Earth kml can be initiated by simply clicking on the “Aggregate Results” button in the Simulation Outputs section of the Traxim Controller as shown below. All output files are made available for download by clicking on the “Download Results” button. More detail is provided in Section 2.

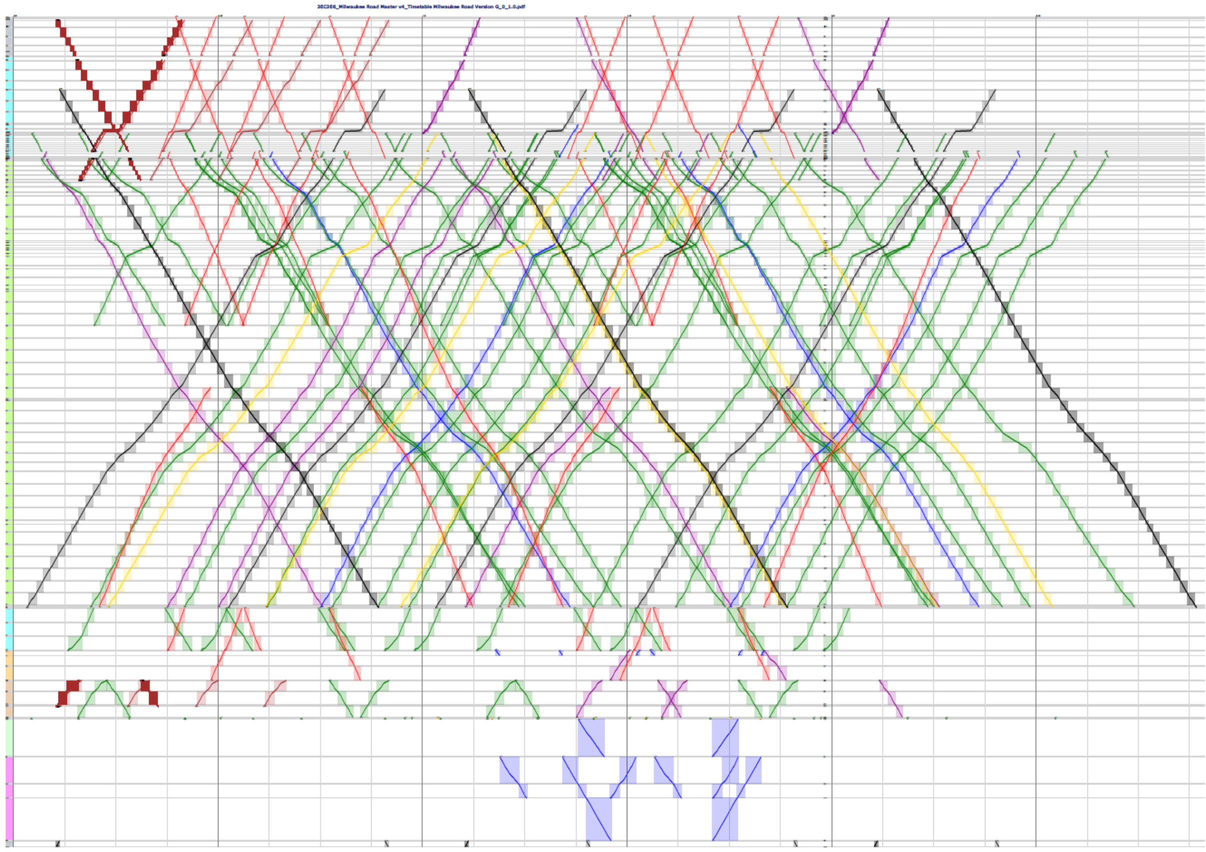


4.2. Train Graph

The validation and resolution processes each generate a train graph. As discussed in Chapter 2, you should review the unresolved train graph after the validation process, and the resolved train graph for a single perturbation, to make sure they look sensible.

For the batch resolution, each of the train graphs will be different, but since the purpose is to create a population of timetables, individual differences should not be of consequence. Hence, we do not provide each of the individual train graphs as part of the batch resolution.

The unresolved train plan pdf will look similar to this:



Key points about the train graph are as follows:

- A grey horizontal line represents a node. Note that whether a node is shown or not is user defined in the Infrastructure file. Hence not all nodes will necessarily appear on the train graph.
- Grey vertical lines are time, with the darker lines at 12-hour intervals, and the lighter lines at 3-hour intervals.
- The solid line represents the front of the train and the more lightly coloured box is the section occupation.
- Nodes are spaced by distance. Hence the train path will be more vertical when the train is simulated to be travelling faster and less vertical when it is travelling slower.
- There is no train linking in the unresolved state. Hence all linked trains will appear as departing at their nominal departure time. If this is set to zero they will all be shown as departing simultaneously, as with the maroon trains in the sample above.
- If there are multiple routes for a train between an origin and a destination, as opposed to multiple tracks that it can switch between on an individual route, check that the train is following the expected route. This will be apparent from the regions that it appears in on the train graph. Traxim will find the shortest route by distance of all possible routes. If the train should, in practice use a different route, force this by using the train rules to either exclude nodes or mandatorily require nodes to be passed.
- The height and width of the train graph is dictated by the number of simulation days and the length of the network. Depending on the interplay of these two factors, the train paths will be more or less vertical than shown in the sample above. If using the Train Graph Viewer, the graph is kept at a constant ratio that is consistently readable.
- Train names appear next to the train path where a train enters or leaves a section. These appear in very small type. You may need to zoom to around 1000% to be able to read them.
- The colour density of an occupation reflects the sum of the trains occupying the section. Be alert for darker occupations along the full length of a train path. This is likely to indicate two identical trains scheduled to depart at the same time.
- You may wish to apply additional intelligent design to the timetable at this point to reduce the bunching of trains. Note though that once crosses are resolved, most trains will shift progressively to the right and that what may appear as bunching in the unresolved timetable will be less intensive in the resolved state. That said, having multiple following trains leaving in a cluster is likely to generate congestion, especially if they have similar transit times or if slower trains are scheduled to leave ahead of faster trains.

4.3. Train Stats

The Train Stats output provides extensive statistical information on each train, reported for each region it passes through and in aggregate. Other than for the aggregated “All unwindowed” result, the data is trimmed to the analysis window as specified in the

Scenario inputs, that is, the time period defined by the Simulation Days less the Window Days. Accordingly, the performance metrics will vary where trains that are otherwise identical have different proportions of their journey within the analysis window. Care should be taken to ensure that all metrics are normalized to account for this. Converting metrics to a “per 100 km” basis is usually the most effective method to achieve this.

The following table describes the meaning of each of the columns in the Train Stats output:

Train Number	The unique identifier for the train, as set in the Train Plan, appended with the departure day of the week, and the week number.
Region	The region (or defined aggregate) that the data is for.
Colour Group	The colour group for the train as set in the Train Plan.
Priority	The priority specified for the train in the Train Plan (note that “absolute” appears as 1,000,000).
Locos	The locomotives (or self-propelled unit) specified in the Train Plan.
Length (m)	The train length (including locomotives) as specified in the Train Plan.
Total Weight (tonnes)	Total train weight (that is, including locomotives) as specified in the Train Plan.
Entry Node	The originating node, as set in the Train Plan.
Exit Node	The route end point node, as set in the Train Plan.
Entry Arrival Time	The starting time of the train, derived from the nominal entry time plus randomization within the defined perturbation window, in minutes from midnight on day 1. Note that the train may be delayed at its entry node, so this time can be different to the time it departs the start node. When aggregated, this is the average of all entry arrival times.
Exit Arrival Time	The time that the train arrives at its destination node. When aggregated, this is the average of all exit arrival times.
Distance (km)	Actual distance travelled. When aggregated this is the average distance travelled within the window. Hence, it may vary between otherwise identical trains if the train journey is partly outside the window time.
Raw Transit Time (m)	The dynamically simulated transit time, averaged across perturbations, and windowed. Note that as this is drawn from the resolved timetables, it includes acceleration and deceleration due to deconflicting dwells. Accordingly, trains that are identical as defined in the Train Plan will still have different raw transit times even if they operated fully within the window period. If raw undelayed transit time is required, this is available in the Train Stats file for the validation output.
Scheduled Dwell (m)	Dwell that has been specified for the train in the Rules input as “DwellAt” (windowed as appropriate).
Following Deconflicting Dwell (m)	Dwell incurred for a cross that was triggered by a following conflict. This includes overtaking (windowed as appropriate).
Opposing Deconflicting Dwell (m)	Dwell incurred for a cross that that was triggered by an opposing conflict (windowed as appropriate).
Total Deconflicting Dwell (m)	The sum of the following and opposing dwell minutes.
Total Transit (m)	The sum of scheduled dwell, deconflicting dwell, and raw transit time, also equal to the time between initial train departure and final train arrival.
Opposing Conflicts Lost	The number of opposing conflicts where the train was required to deviate from its unconflicted path. A deviation includes both deviations in time (dwells), and routes (diverging to a different track), or both.
Opposing Conflicts Won	The number of opposing conflicts where the train forced another train to deviate from its unconflicted path.
Following Conflicts Lost	The number of following conflicts where the train was required to deviate from its unconflicted path.
Following Conflicts Won	The number of following conflicts where the train forced another train to deviate from its unconflicted path.
Traction Energy (MWh)	An estimate of the amount of energy consumed by the train.

Regenerative Energy (MWh)	An estimate of the amount of energy generated by braking. This energy would be returned to the system where electrified, a battery in a hybrid locomotive set-up, or vented as heat in a conventional dynamically braked diesel locomotive.
Time Powering (m)	The amount of time the train is modelled as being under power.
Time Braking (m)	The amount of time the train is modelled as being under braking resistance.
Time Idling (m)	The amount of time the train is modelled as being neither under power nor braking.

It is possible that in some perturbations a train will both begin and end its journey in the timetable “run-in” period (ie before the window period) and in other perturbations it won’t. Accordingly, it’s possible, though rare, to record a train count of less than 30. Similarly, trains that are only partially in the window period will have their data aggregated for different amounts of time and distance across perturbations. For this reason, it can be unsafe to use the train count result when analysing data.

Given the effect of windowing, it is recommended that all metrics be normalised on a per km basis. For instance, if the aim is to calculate the transit time for a specific train, it is most correct to divide total transit time by total distance (for the train or train group) to arrive at an average speed, and then multiply it by the origin to destination distance.

The data in the train detail file is best analysed using a pivot table in Excel. Using the pivot table you can generate averages and other statistics for the metrics you specify. You can filter by any combination of train types (as specified in the timetable input file) and any combination of regions (as specified in the regions input file).

4.4. Train Timetable

The primary purpose of the Train Timetable when used for statistical analysis is to support calculation of section utilization. Section utilization is the key metric for capacity analysis.

Although statistics can be calculated directly for section utilization, there is a conceptually difficult decision that needs to be made in defining what constitutes a section for utilization purposes. This is particularly true on single track lines, especially where there is a mix of train lengths and loop lengths. Intermediate signals, and mid-section junctions add to the complexity.

To allow an analyst flexibility, the approach has been taken of providing a resolved Train Timetable as the key data source for utilization calculation. Importantly, the arrival /departure times are reported for nodes selected to be drawn on the train graph. Hence, to make sure that all of the data for the sections to be analysed is available, care should be taken in setting the “draw” status of nodes. Also note that the times reported are for the front of the train. As section utilization doesn’t end until the rear of the train has cleared the section, the analyst should take care to also select locations that reflect when the front of the train arrives it means the rear of the train is clear.

It is worth commenting here that there is no objective or correct limit for utilization, or a correct time period over which it should be measured.

For single track, utilization in the order of 60% - 70% is likely to be the top-end of what is realistically achievable without high risk.

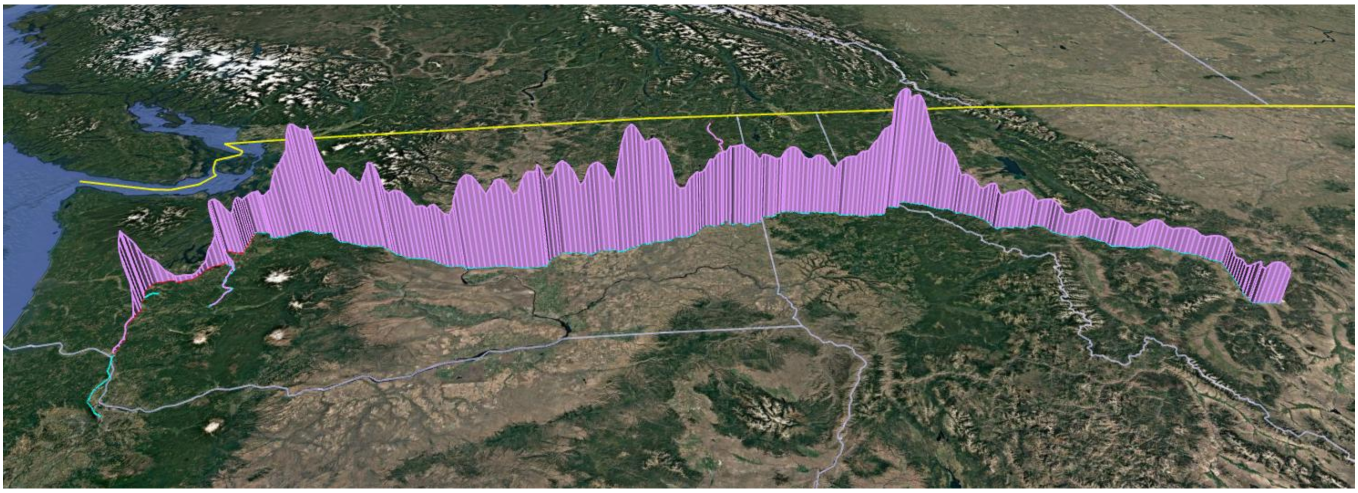
In terms of time-periods for defining utilization, if a traffic has high levels of flexibility over train departure times, such as on a largely bulk network, it is likely to be more appropriate to use a 24-hour utilization rate. If the traffics have relatively inflexible departure windows, such as for some intermodal traffics, a shorter window for utilization may be useful. Either way, peaks and troughs in train numbers may create artificial congestion. This might manifest as high levels of delay, but not necessarily lead to high utilization when calculated over a 24-hour period. In this case, relaxing the constraints around departure windows may be as effective as infrastructure enhancements.

Acceptable levels of section utilization are also influenced by the utilization of nearby sections. If the surrounding sections have relatively low utilization rates, a single section with a high utilization rate, like 80% over 24 hours, may be tolerable, whereas a series of sections all with utilization rates of around 60% may create unsustainable congestion.

4.5. Delay by Location in Google Earth

Traxim generates a Google Earth kml file that shows delay by location when opened in Google Earth. This view allows you to instantly visualise where on your network you are incurring the greatest delay, which is likely to be where you want to focus any capacity enhancement efforts. Google Earth allows you to open multiple kml files, so you can compare delay by location for different Infrastructure or Train Plan inputs.

When opened, your kml file should look similar to the following image showing delay for the Milwaukee Road Example Scenario:



When the kml file is being generated it's necessary to make an assumption about the amount of smoothing to apply. At one extreme, delay can be shown for each individual node, that is, with no smoothing. However, this fine level of detail can sometimes hide important information. In particular, if the network has a range of train lengths and loop lengths it's likely that the resultant graph will be very peaky. This tends to be distracting when trying to identify where on a large network the greatest levels of delay are occurring.

At the other extreme, delay could be averaged across all nodes, which would be meaningless.

The level of smoothing defaults to a distance of 25 km, which typically generates a meaningful result for most networks. If you feel that this smoothing factor isn't giving you adequate insight, please contact us and we can create alternative versions (at no charge).

Appendix 1 – The Infrastructure Input File

The Infrastructure file is usually best edited using the Network Editor tool. However, there are circumstances where working directly with the csv file can be effective, for instance to rename a region, or to identify links where directionality is disabled.

The Infrastructure input file format is as follows:

#6	Traxim v6 2026-01-28 17:06:11 nodes 179 connections 240										
Milwaukee Road											
#Name	F Enabled	T Enabled	D Enabled	F Node	F on branch	T Node	T on branch	D Node	D on branch	Region1	Km1
Seattle Station Jct	TRUE	TRUE	TRUE	Seattle Station Branch Jct	D	Seattle Union Station Platform 1	F	Seattle Union Station Platform 2	F	Longview WA - Seattle WA	236.43
Seattle Union Station Platform 2	TRUE	TRUE	TRUE	Seattle Station Jct	D					Longview WA - Seattle WA	236.63
Seattle Union Station Platform 1	TRUE	TRUE	TRUE	Seattle Station Jct	T					Longview WA - Seattle WA	236.63
Seattle Station Branch Jct	TRUE	TRUE	TRUE	Seattle Pass Crossover W	F	Seattle Intermodal South End	T	Seattle Station Jct	F	Longview WA - Seattle WA	231.55
Seattle Intermodal South End	TRUE	TRUE	TRUE	Seattle Intermodal North End	F	Seattle Station Branch Jct	T	Seattle Pass Crossover E	T	Longview WA - Seattle WA	231.75
Seattle Intermodal North End	TRUE	TRUE	TRUE	Seattle Intermodal South End	F					Longview WA - Seattle WA	234
Seattle Pass Crossover W	TRUE	TRUE	TRUE	Seattle Station Branch Jct	F	Duwamish West Track	F	Seattle Pass Crossover E	D	Longview WA - Seattle WA	231.35
Seattle Pass Crossover E	TRUE	TRUE	TRUE	Duwamish East Track	F	Seattle Intermodal South End	D	Seattle Pass Crossover W	D	Longview WA - Seattle WA	231.15
Duwamish East Track	TRUE	TRUE	TRUE	Seattle Pass Crossover E	F	Black River Jct X2	F			Longview WA - Seattle WA	226.65
Duwamish West Track	TRUE	TRUE	TRUE	Seattle Pass Crossover W	T	Black River Jct X1	T			Longview WA - Seattle WA	226.65
Black River Jct X1	TRUE	TRUE	TRUE	Black River North Jct	F	Duwamish West Track	T	Black River Jct X2	D	Longview WA - Seattle WA	221.55
Black River East Jct	TRUE	TRUE	TRUE	Renton Loop W	F	Black River North Jct	D	Black River South Jct	D	Black River Jct WA - Butte MT	15.79
Black River North Jct	TRUE	TRUE	TRUE	Black River Jct X1	F	Black River South Jct	T	Black River East Jct	T	Black River Jct WA - Butte MT	15.29
Black River South Jct	TRUE	TRUE	TRUE	Black River Jct X3	F	Black River North Jct	T	Black River East Jct	D	Longview WA - Seattle WA	220.65
Renton Loop W	TRUE	TRUE	TRUE	Black River East Jct	F	Renton Loop E	T	Renton Loop E	D	Black River Jct WA - Butte MT	16.4
Orillia Yard North End East Track	TRUE	TRUE	TRUE	Black River Jct X3	T	Orillia Yard South End East Track	T	Orillia Yard South End East Track	D	Longview WA - Seattle WA	217
Orillia Yard South End East Track	TRUE	TRUE	TRUE	Orillia Yard South End X1	F	Orillia Yard North End East Track	T	Orillia Yard North End East Track	D	Longview WA - Seattle WA	214.75
210km Signal East Track	TRUE	TRUE	TRUE	Orillia Yard South End X1	T	205 km Signal East Track	T			Longview WA - Seattle WA	210
205 km Signal East Track	TRUE	TRUE	TRUE	200 km Signal East Track	F	210km Signal East Track	T			Longview WA - Seattle WA	205
200 km Signal East Track	TRUE	TRUE	TRUE	205 km Signal East Track	F	195 km Signal East Track	F			Longview WA - Seattle WA	200

#Name	Region2	Km2	Region3	Km3	Default branch	PosX	PosY	Length	Width	Rotation	Flip	Mirror	Draw	OffsetX	OffsetY	Latitude	Longitude
Seattle Station Jct					T	1537.8371	6.4803	40	20	275	FALSE	FALSE	TRUE	0	0	47.59642	-122.329
Seattle Union Station Platform 2					T	1537.7583	0	20	20	265	FALSE	FALSE	TRUE	-4.2545	-26.842	47.59836	-122.329
Seattle Union Station Platform 1					T	1537.7583	0	20	20	280	FALSE	FALSE	TRUE	5.4088	-33.5065	47.59836	-122.329
Seattle Station Branch Jct					T	1557.3562	146.2167	40	20	225	TRUE	FALSE	TRUE	9.9649	0.7185	47.55457	-122.321
Seattle Intermodal South End					T	1553.9679	141.2331	40	20	15	TRUE	FALSE	TRUE	-1.7593	-0.1851	47.55607	-122.322
Seattle Intermodal North End					T	1540.9781	78.7928	20	20	180	FALSE	FALSE	TRUE	-285.238	54.3253	47.57476	-122.328
Seattle Pass Crossover W					T	1560.7165	151.2193	40	10	70	TRUE	FALSE	TRUE	9.7697	2.0083	47.55307	-122.319
Seattle Pass Crossover E					T	1564.1658	156.1609	40	10	250	TRUE	FALSE	TRUE	7.4243	1.8211	47.5516	-122.318
Duwamish East Track					T	1629.5836	272.6946	40	20	60	FALSE	FALSE	TRUE	-3.7399	1.2212	47.5167	-122.289
Duwamish West Track					T	1629.5836	272.6946	40	20	60	TRUE	FALSE	TRUE	-2.446	0.75	47.5167	-122.289
Black River Jct X1					T	1711.1261	401.0516	40	10	220	FALSE	FALSE	TRUE	-8.568	-1.6969	47.47827	-122.254
Black River East Jct	Renton WA Triangle	0.535			T	1726.7821	409.4215	40	20	170	FALSE	FALSE	TRUE	5.5269	1.2502	47.47576	-122.247
Black River North Jct	Longview WA - Seattle WA	221.348			T	1713.7212	405.6431	40	20	40	FALSE	FALSE	TRUE	2.3452	3.8242	47.47689	-122.252
Black River South Jct	Renton WA Triangle	0			T	1725.365	423.832	40	20	250	TRUE	FALSE	TRUE	2.1689	-2.1497	47.47144	-122.247
Renton Loop W					T	1741.1205	399.1316	40	20	15	TRUE	FALSE	TRUE	122.4673	-15.4925	47.47884	-122.24
Orillia Yard North End East Track					T	1739.6637	529.9234	40	20	85	FALSE	FALSE	TRUE	1.6193	0.3534	47.43968	-122.241
Orillia Yard South End East Track					T	1739.7922	597.5033	40	20	270	TRUE	FALSE	TRUE	3.4292	-0.9627	47.41944	-122.242
210km Signal East Track					T	1742.9651	740.0778	40	20	90	FALSE	FALSE	TRUE	-3.6205	0.4054	47.37675	-122.241
205 km Signal East Track					T	1748.9372	890.0123	40	20	270	TRUE	FALSE	TRUE	0.3146	0.8747	47.33185	-122.239
200 km Signal East Track					T	1738.0448	1037.7161	40	20	95	FALSE	FALSE	TRUE	-1.5719	1.7081	47.28762	-122.244

Each row represents a node. There are three types of node:

- *Turnout*: 3 branches, designated:
    - F – For the facing branch.
    - T – For the trailing or through (non-diverge) branch.
    - D – For the divergent branch.
- *Signal*: Two branches connected. These must be the T and F branches.
- *Track end point*: Only one branch connected. This must be the F branch.

Name is the name of the node.

Enabled (of types F, T and D) is a bool that specifies whether a train can enter the link connected to that branch (that is, whether the link is signaled to allow for that movement).

FNode is the node that the facing branch connects to.

F on branch is the branch (F, T or D) of the connected node that the facing branch of the node connects to.

TNode is the node that the through branch connects to.

T on branch is the branch (F, T or D) of the connected node that the through branch of the node connects to.

DNode is the node that the diverge branch connects to.

D on branch is the branch (F, T or D) of the connected node that the diverge branch of the node connects to.

Region1 is the name of the first region (track geometry string) that node exists within. A node must always be within at least one region.

Km1 is the reference kilometrage of the node for the first Region it exists within. The kilometrage references the nominal kilometrage as specified in the geometry input file for the region.

Region2 is the name of the second region that the node exists within. This is only populated if the node is at the interface of two regions.

Km2 is the reference kilometrage of the node for the second Region it exists within.

Region3 is the name of the third region that the node exists within. This is only populated if the node is at the interface of three regions.

Km3 is the reference kilometrage of the node for the third Region it exists within.

Default branch is the branch (T or D) that it is preferred a train entering from the facing end will take. When the default path of the train is being generated, the train will take the default branch provided it is physically possible to reach the destination node via that branch, and the path to the destination is no more than 10 kilometers longer than a route using the non-default branch. When crossing options are being assessed, the non-default branch will always be penalized relative to the default branch. The default branch can also be set to N (for neutral) if there is no preference between branches. In this case, the default path for the train will prefer the through branch. When crossing decisions are being taken, there will be no penalization of either branch.

PosX is the horizontal position of the node on the Network Editor grid. This represents its geoposition as defined by its longitude.

PosY is the vertical position of the node on the Network Editor grid. This represents its geoposition as defined by its latitude.

Length is the length of the node as it appears in the Network Editor. This variable is not used by the algorithm. It is for visualisation purposes only.

Width is the gap between the T and D branches of the node as it appears in the Network Editor. This variable is not used by the algorithm. It is for visualisation purposes only.

Rotation is the angle within a 360 degree rotation that the node renders in the Network Editor. This variable is not used by the algorithm. It is for visualisation purposes only.

Flip determines which side of the node the diverge end appears within the Network Editor. This variable is not used by the algorithm. It is for visualisation purposes only.

Mirror determines the direction of the F and T branches. This variable is not used by the algorithm. It is for visualisation purposes only. It is functionally equivalent to rotating the node 180 degrees.

Draw determines whether the node is rendered on the Train Graph, and whether it is reported as a timing point in the timetable output file.

OffsetX is an offset applied to the horizontal position of the node on the Network Editor grid, relative to PosX. This allows the node to be placed in a position offset from its actual geoposition.

OffsetY is an offset applied to the vertical position of the node on the Network Editor grid, relative to PosY. This allows the node to be placed in a position offset from its actual geoposition.

Latitiude is the latitude of the node interpolated from the latitude, longitude, and nominal kilometrages in the geometry input file. It is for information only, and not used as a variable in the resolver.

Longitude is the longitude of the node interpolated from the latitude, longitude, and nominal kilometrages in the geometry input file. It is for information only, and not used as a variable in the resolver.

Appendix 2 – The Train Plan Input File

Traxim uses the train plan input file to define the characteristics of the trains to be simulated.

Note that the file to be used must be in csv format. However, included in the Traxim Resources is an Excel version of the train plan input spreadsheet, which includes a number of formulas to simplify the train plan file creation process. The train plan input can be set-up in the Excel master spreadsheet and then the Excel spreadsheet saved as a csv.

The train plan input file format is as follows:

#3	Milwaukee Road Example Scenario - Train Plan Input																									
#colour descriptions																										
colour pairs	Blue	General Freight																								
colour pairs	Brown	Minerals																								
colour pairs	Gold	Grain																								
colour pairs	Red	Passenger																								
colour pairs	SteelBlue	Steel																								
colour pairs	DarkMagenta	Express freight																								
colour pairs	Indigo	Oil																								
colour pairs	Green	IMEX containers																								
colour pairs	DarkBlue	Automobiles																								
colour pairs	Black	Coal																								
colour pairs	Brown	Iron Ore																								
#						1	2	3	4	5	6	7														
#Name	Description	Start Quarter	Start Year	End Quarter	End Year	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour	Starttime (mins)	Max Entry Variation	Colour	Start	End	Rules	Linked Train	Trailing Weight	Length	Priority	Max Km/H	Speed Class	Traindetails
~SECH1	~SECH1 - Seattle to Chicago Express	1	2025					1	1	1	1	1	30	30	60	DarkMagenta	Seattle Intermodal North End	Butte Jct W			4456	1750	8	115	Normal	SD70ACe;SD70ACe;SD70ACe;SD70ACe
~SECH2	~SECH2 - Seattle to Chicago Express	1	2025			1	1	1	1	1	1	1	145	105	60	DarkMagenta	Seattle Intermodal North End	Butte Jct W			4456	1750	8	115	Normal	SD70ACe;SD70ACe;SD70ACe;SD70ACe
~SECH3	~SECH5 - Seattle to Chicago Merchandise	1	2027			1	1	1	1	1	1	1	1500	900	60	Blue	Seattle Intermodal North End	Butte Jct W			4456	1750	7	115	Normal	SD70ACe;SD70ACe;SD70ACe;SD70ACe
~CHSE1	~CHSE2 - Chicago to Seattle Express	1	2025			1			1	1	1	1	900	540	60	DarkMagenta	Butte Jct W	Seattle Intermodal North End			4456	1750	8	115	Normal	SD70ACe;SD70ACe;SD70ACe;SD70ACe
~CHSE2	~CHSE5 - Chicago to Seattle Merchandise	1	2027			1	1	1	1	1	1	1	1800	1080	60	Blue	Butte Jct W	Seattle Intermodal North End			4456	1750	7	115	Normal	SD70ACe;SD70ACe;SD70ACe;SD70ACe
~CHSE3	~CHSE7 - Chicago to Seattle IMEX	1	2025			1	1	1	1	1	1	1	2310	1390	60	Green	Butte Jct W	Seattle Intermodal North End			4456	1750	5	115	Normal	SD70ACe;SD70ACe;SD70ACe;SD70ACe
~CHSP1	~CHSP1 - Chicago to Spokane Express	1	2025				1	1	1	1	1		230	150	60	DarkMagenta	Butte Jct W	Plummer Jct			4456	1750	8	115	Normal	AC4400CW;AC4400CW;AC4400CW;AC4400CW
~SPCH1	~SPCH1 - Spokane to Chicago Express	1	2025				1	1	1	1	1		2000	1200	60	DarkMagenta	Marengo Jct	Butte Jct W			4456	1750	8	115	Normal	AC4400CW;AC4400CW;AC4400CW;AC4400CW
~CHTC1	~CHTC1 - Chicago to Tacoma IMEX	1	2025			1	1	1	1	1	1	1	430	270	60	Green	Butte Jct W	Tacoma Port Jct			4456	1750	4	115	Normal	AC4400CW;AC4400CW
~TCCH1	~TCCH3 - Tacoma to Chicago IMEX	1	2025			1	1	1	1	1	1	1	1400	840	60	Green	Tacoma Port Jct	Butte Jct W			4456	1750	4	115	Normal	AC4400CW;AC4400CW
~SESP1	~SESP1 - Seattle to Spokane IMEX	1	2025			1	1	1	1	1	1	1	500	300	60	Green	Seattle Intermodal North End	Marengo Jct			4456	1750	4	115	Normal	AC4400CW;AC4400CW
~SPSE1	~SPSE1 - Spokane to Seattle IMEX	1	2025			1	1	1	1	1	1	1	1100	660	60	Green	Marengo Jct	Seattle Intermodal North End			4456	1750	4	115	Normal	AC4400CW;AC4400CW
~TCSP1	~TCSP1 - Tacoma to Spokane IMEX	1	2025			1	1	1	1	1	1	1	100	60	60	Green	Tacoma Port Jct	Marengo Jct			4456	1750	4	115	Normal	AC4400CW;AC4400CW
~TCSP2	~TCSP2 - Tacoma to Spokane IMEX	1	2027			1	1	1	1	1	1	1	1000	600	60	Green	Tacoma Port Jct	Marengo Jct			4456	1750	4	115	Normal	AC4400CW;AC4400CW
~SPTC1	~SPTC1 - Spokane to Tacoma IMEX	1	2025			1	1	1	1	1	1	1	310	190	60	Green	Marengo Jct	Tacoma Port Jct			4456	1750	4	115	Normal	AC4400CW;AC4400CW
~SPTC2	~SPTC2 - Spokane to Tacoma IMEX	1	2027			1	1	1	1	1	1	1	2145	1305	60	Green	Marengo Jct	Tacoma Port Jct			4456	1750	4	115	Normal	AC4400CW;AC4400CW
~TCVC1	~TCVC2 - Tacoma to Vancouver IMEX	1	2025			1	1	1	1	1	1	1	1650	1010	60	Green	Tacoma Port Jct	Cedar Falls Jct W			4456	1750	4	115	Normal	SD70ACe;SD70ACe;SD70ACe
~VCTC1	~VCTC1 - Vancouver to Tacoma IMEX	1	2025			1	1	1	1	1	1	1	345	225	60	Green	Cedar Falls Jct W	Tacoma Port Jct			4456	1750	4	115	Normal	SD70ACe;SD70ACe;SD70ACe
~NDG1E	~NDG1E - North Dakota grain	1	2025				1	1	1	1	1	1	400	240	180	Gold	Tacoma Port Jct	Butte Jct W			819	1750	2	80	Normal	AC4400CW;AC4400CW
~NDG1W	~NDG2W - North Dakota grain	1	2025			1		1	1	1	1	1	1445	885	180	Gold	Butte Jct W	Tacoma Port Jct			4456	1750	2	80	Normal	AC4400CW;AC4400CW
~CBX- E	~CBX- E - Columbian	1	2025			1	1	1	1	1	1	1	1400	840	0	Red	Seattle Union Station Platform 1	Butte Jct W			480	300	Absolute	130	Passenger	AC4400CW;AC4400CW
~CBX-W	~CBX-W - Columbian	1	2025			1	1	1	1	1	1	1	500	300	0	Red	Butte Jct W	Seattle Union Station Platform 1			480	300	Absolute	130	Passenger	AC4400CW;AC4400CW
~SHX-E	~SHX-E - Spokane Hiawatha	1	2025			1	1	1	1	1	1	1	900	540	0	Red	Seattle Union Station Platform 2	Marengo Jct			480	300	Absolute	130	Passenger	AC4400CW;AC4400CW
~SHX-W	~SHX-W - Spokane Hiawatha	1	2025			1	1	1	1	1	1	1	900	540	0	Red	Marengo Jct	Seattle Union Station Platform 2			480	300	Absolute	130	Passenger	AC4400CW;AC4400CW
~SF01	~SF01 - Sound Transit Commuter	1	2025				1	1	1	1	1		500	300	0	Red	Seattle Union Station Platform 2	Frederickson Turnback	Seattle - Frederickson commuter		1	1	Absolute	130	Passenger	DMU;DMU;DMU;DMU
~FS01	~FS01 - Sound Transit Commuter	1	2025				1	1	1	1	1		500	300	0	Red	Frederickson Turnback	Seattle Union Station Platform 2	Frederickson - Seattle commuter		1	1	Absolute	131	Passenger	DMU;DMU;DMU;DMU
~SF02	~SF02 - Sound Transit Commuter	1	2025				1	1	1	1	1		530	330	0	Red	Seattle Union Station Platform 2	Frederickson Turnback	Seattle - Frederickson commuter		1	1	Absolute	130	Passenger	DMU;DMU;DMU;DMU
~FS02	~FS02 - Sound Transit Commuter	1	2025				1	1	1	1	1		530	330	0	Red	Frederickson Turnback	Seattle Union Station Platform 2	Frederickson - Seattle commuter		1	1	Absolute	130	Passenger	DMU;DMU;DMU;DMU
~SF03	~SF03 - Sound Transit Commuter	1	2025			1	1	1	1	1	1	1	600	360	0	Red	Seattle Union Station Platform 2	Frederickson Turnback	Seattle - Frederickson commuter		1	1	Absolute	130	Passenger	DMU;DMU;DMU;DMU
~FS03	~FS03 - Sound Transit Commuter	1	2025			1	1	1	1	1	1	1	600	360	0	Red	Frederickson Turnback	Seattle Union Station Platform 2	Frederickson - Seattle commuter		1	1	Absolute	130	Passenger	DMU;DMU;DMU;DMU

The pre-header rows allow a train category to be assigned to a colour. This train category as defined by its colour can be used as a filter in the results outputs, as well as determining the colour of the train in the train graph. The available colours are shown in the colour chart in Appendix 8.

Note that column headers are for convenience only. The file input reads by column number, not header name. Hence, column headers can be modified without consequence.

Name is the train number. The train number must be unique (other than where the multiple day of operation arrangement is used).

Description is a field to describe the pertinent features of the train. This is provided for convenience only and is not an input to the simulation.

Start quarter, start year, end quarter and *end year* provide a mechanism to use a single input timetable to manage a changing train pattern over time. A run date is specified in the settings for the simulation, and trains that are not applicable to the run date are filtered out of the train scenario to be simulated. Note that a start quarter and start year must be specified for a train to be recognized, but it is not essential to specify an end quarter or end year.

The *day of the week* columns determine what day or days the train is scheduled to enter the network. Where a train with the same characteristics runs on multiple days of the week, simply enter a 1 into the column for each of the relevant days. Note though that it isn't possible to have differentiating train numbers when using this approach. The day specified in the header row is for convenience only. Traxim will originate trains starting from column G as the first day of the timetable. This can represent any day of the week. Note that the 'linked train' feature requires a unique train number to reference. Hence linked trains can only be set up where a train is entering on a single day.

Hour is the departure time in hours and minutes specified using the 24 hour clock. The syntax for entering the time is hhmm. This column is used in the xlsx version of the trains file to simplify generation of the time in the Starttime column. It is not an input to the simulation in itself and can be blank or zero in the csv file.

Starttime (mins) is the departure time of the train in minutes from midnight. This start time is automatically generated in the xlsx version of the train input file using the time entered in the hour column, but can also be manually entered.

Max Entry Variation specifies the time range for randomization of the departure of a given train. Actual train departure will be a time generated randomly between the start time, and the start time plus the Max Entry Variation. Traxim works on a principle of randomizing each timetable based on a seed. Max Entry Variation determines the allowable time band that it would be acceptable for a given train to depart within. Trains that would be typically expected to depart with a precise departure time, such as passengers, can have a range of zero specified. Note though that to the extent that trains have fixed departure times, it will reduce the uniqueness of the timetable result.

Colour specifies the colour to use for the train on the train graph pdf. It is effectively a way to group trains for simplicity of reporting, and is also used in the results outputs as the key dimension by which the performance results for groups of trains are summarised. It must be a colour as specified in the colour matrix at Appendix 8.

Start is the starting location for the train and must be identical to a node name. *End* is the terminating location for the train and must also be identical to a node name.

Rules references the timetable rules input file to determine any special rules that the train must comply with on its journey. Multiple rules can be specified for each train, in which case they should be separated by a semi-colon (";").

Linked train (forms) references the name of another train that must only depart after the subject train arrives. This function can serve three distinct purposes:

- It can be used where a single train set is being used, and one train must therefore arrive before another can depart.
- It can be used where a train's consist changes through its journey, for instance to add banking engines.
- It can be used where wagons or passengers need to meet a connecting service.

Multiple other trains can be linked, in which case they should be separated by a semi-colon (";"). This would apply, for instance, where two passenger trains can both only depart after the subject train arrives.

Weight is the trailing tonnes (ie not including locomotives) of the train. Note that for a self-propelled train, like a DMU or light engine, weight should be set to "1". Setting the weight to blank or zero will cause an error.

Length is the trailing length (ie not including locomotives) of the train. Note that for a self-propelled train, like a DMU or light engine, length should be set to "1". Setting the length to blank or zero will cause an error.

Priority is the priority to be awarded to the train when choosing between crossing options to resolve a conflict and can be any number greater than zero. Estimated delay for a crossing option is multiplied by the train priority to create a utility value. This utility value is then used in the option selection process. Hence a crossing option that causes a 20-minute delay for a train of priority 5 will have the same utility value as a crossing option that causes a delay of 10 minutes for a train of priority 10.

'Absolute' is also an option that can be entered as the train priority. In that case the train will have absolute priority over any other train that isn't also of absolute priority. When there is a conflict between two absolute priority trains, the crossing solution that generates the smallest delay will be selected. Hence, any sequence of trains of absolute priority will always resolve to the same pattern of crosses. As such, this feature can be used to generate a repeating timetable for, for instance, a commuter operation.

Max Km/Hh is the maximum permissible speed of the train. Traxim will limit the train speed to the lesser of the maximum permissible speed of the train, the maximum permissible speed of the locomotives on the train, and the applicable speedboard.

Speed class determines which set of speedboards will be applied to the train, as specified in the Speedboards input file.

Locos determines the locomotives to be used on the train, referencing the relevant Tractive Effort / Braking Curve input files for the locomotive type. The syntax for inputting the locomotives is each locomotive type separated by a semicolon (";").

The xlsx file also has a couple of additional columns (AE – AG) which are for simplifying the creation of train names and descriptions. Note that with the csv input files, any row starting with a # is ignored. If you want to temporarily change a timetable input file to exclude certain trains, these can simply be 'hashed-out' by putting a hash in front of the train name in column A.

Appendix 3 – The Rules Input File

The Rules input file format is as follows:

# Rule Name	Type: dwellat, exclude location, [waituntil, include location]	Node	Dwell / Depart
Not via Spokane	exclude location	Spokane Yard E	
Seattle - Portland pass	dwellat	Tacoma Station Platform Road W	3
Portland - Seattle pass	dwellat	Tacoma Station Platform Road E	3
Crew Change location	dwellat	Marengo Loop E	10
Via Spokane	exclude location	Rosalia Loop E	
Seattle - Frederickson commuter	dwellat	Tacoma Station Platform Road W	2
Seattle - Frederickson commuter	dwellat	Frederickson Station	2
Frederickson - Seattle commuter	dwellat	Tacoma Station Platform Road E	2
Frederickson - Seattle commuter	dwellat	Frederickson Station	2
Olympian Hiawatha	dwellat	Ellensburg Station	2
Olympian Hiawatha	exclude location	Spokane Yard E	
Olympian Hiawatha	dwellat	St Maries Station	2
Olympian Hiawatha	dwellat	Missoula Station	2
Centralia empty coal	dwellat	Centralia Power Station	150
Centralia empty coal	exclude location	Spokane Yard E	
Centralia loaded coal	exclude location	Spokane Yard E	

Rule Name is the name of the rule and is cross-referenced to the Train Plan file. A single rule name can have an unlimited number of individual rule instances. In the above example, the Olympian Hiawatha rule specifies three locations where a train subject to the rule dwells at stations, plus a rule to ensure it uses the direct mainline rather than diverting via Spokane. Note that the Train Plan file can accept multiple rules for an individual train.

There are four *types* of rule. These rules are:

- include location: Specifies a node that the train must pass through.
- exclude location: Specifies a node that the train cannot pass through.
- dwellat: A predetermined amount of time in minutes a train must dwell at a node.
- waituntil: A predetermined time before which a train cannot depart a node.

Note that the dwellat function does not apply at the destination node of a train. If the aim is to include a dwell at a location for a linked train, the dwell needs to be applied to the commencement of the journey of the second train, not the arrival of the first train.

Node is the name of the node that the rule applies to. It must be a valid node name listed in the Infrastructure file.

Dwell/Depart is the amount of time, in minutes, that the train will dwell for, or the time at which the train can depart expressed in minutes from 00:01 on Sunday. For the “include location” and “exclude location” rules this cell can be left blank.

Appendix 4 – The Speedboards Input File

The speedboards input file format is as follows:

#Speedboards, KM direction header for trains travelling within region. E.g. for passenger train going from low km to high km values they would see the passenger increasing column.										
KMDirectionSection	Normal	increasing								
KMDirectionSection	Normal	decreasing								
KMDirectionSection	Passenger	increasing								
KMDirectionSection	Passenger	decreasing								
KMDirectionSection	HSR	increasing								
KMDirectionSection	HSR	decreasing								
#TrainClasses										
#Section	km	Normal increasing	Normal decreasing	Passenger increasing	Passenger decreasing	Spare	Spare	Location		
Longview WA - Seattle WA	0.075	115								
Longview WA - Seattle WA	0.9	75								
Longview WA - Seattle WA	0.925	60								
Longview WA - Seattle WA	0.95	50								
Longview WA - Seattle WA	1.025	45								
Longview WA - Seattle WA	1.3248	55								
Longview WA - Seattle WA	1.3748	70								
Longview WA - Seattle WA	1.4248	100								
Longview WA - Seattle WA	2.1998	115								
Longview WA - Seattle WA	7.2747	105								
Longview WA - Seattle WA	7.2997	95								
Longview WA - Seattle WA	7.3247	90								
Longview WA - Seattle WA	7.3747	105								
Longview WA - Seattle WA	8.0996	115								
Longview WA - Seattle WA	9.1996	110								
Longview WA - Seattle WA	9.2246	105								
Longview WA - Seattle WA	9.2746	100								
Longview WA - Seattle WA	9.5246	105								
Longview WA - Seattle WA	9.5746	115								

Section is the region to which the speedboard applies. This cross-references the regions as per the track geometry input files to geoposition the speedboard.

km is the kilometrage of the speedboard. This location references the kilometrage in the track geometry file. The kilometrage used in the speedboard file does not need to exactly coincide with a kilometrage in the track geometry file. Traxim will interpolate the location of the speedboard if it falls between two kilometrages.

The Speedboard input file can accommodate an unlimited number of speedboard categories such that different permissible speeds apply to different train types. The type cross-references the Speed Class column in the Train Plan input file. In the Milwaukee Road example scenario, two categories of speedboard are used – 'Normal' and 'Passenger'. It is also necessary to specify whether the speedboard category applies to the increasing or decreasing km direction. Traxim draws this information from the rows above the header row.

Note that enabling the "Auto Curve Speed" setting in the Traxim Controller will cause Traxim to automatically calculate a maximum permissible curve speed based on inferred curve radius from the data in the track geometry file, and nominal superelevation and cant deficiency settings in the Regions file. In this case, maximum speed at any location will be the lesser of the calculated curve speed, the speedboarded speed, the maximum train speed, and the maximum locomotive speed.

Appendix 5 – The Track Geometry Input File

The track geometry input file format is as follows:

#region name	latitude	longitude	elevation	kilometera	comments
Portland OR - Longview WA	45.52808	-122.675	10	0	
Portland OR - Longview WA	45.52824	-122.675	10	0.025	
Portland OR - Longview WA	45.52842	-122.675	9.9	0.05	
Portland OR - Longview WA	45.52859	-122.675	9.9	0.075	
Portland OR - Longview WA	45.52876	-122.676	9.9	0.1	
Portland OR - Longview WA	45.52894	-122.676	9.8	0.125	
Portland OR - Longview WA	45.52911	-122.676	9.8	0.15	
Portland OR - Longview WA	45.52928	-122.676	9.9	0.175	
Portland OR - Longview WA	45.52946	-122.676	9.9	0.2	
Portland OR - Longview WA	45.52963	-122.677	9.9	0.225	
Portland OR - Longview WA	45.5298	-122.677	9.9	0.25	
Portland OR - Longview WA	45.52998	-122.677	9.9	0.275	
Portland OR - Longview WA	45.53015	-122.677	10.2	0.3	
Portland OR - Longview WA	45.53032	-122.677	10	0.325	
Portland OR - Longview WA	45.5305	-122.678	10	0.35	
Portland OR - Longview WA	45.53067	-122.678	10.1	0.375	
Portland OR - Longview WA	45.53084	-122.678	10.1	0.4	
Portland OR - Longview WA	45.53101	-122.678	10.1	0.425	
Portland OR - Longview WA	45.53119	-122.678	10.1	0.45	
Portland OR - Longview WA	45.53136	-122.679	9.9	0.475	
Portland OR - Longview WA	45.53153	-122.679	9.9	0.5	
Portland OR - Longview WA	45.5317	-122.679	9.9	0.525	
Portland OR - Longview WA	45.53187	-122.679	10	0.55	

Region name is the region to which the geometry datapoints for the relevant row belong. Traxim uses this cell rather than the file name when interpreting the input data. Hence, datapoints for multiple regions can be included within a single csv file. However, it is cleanest and clearest to create separate CSV files for each region and to name the file to reflect the region it covers.

Latitude (column B) and *longitude* (column C) are the positions of the datapoint using the decimal degrees system of representation. Note that Traxim uses the actual geographic distance between the datapoints as defined by their latitude and longitude to determine the distance that the train will travel for the purposes of dynamic simulation of the train performance.

Elevation (column D) is nominally the top of the railhead measured in metres above sea level. Traxim uses the difference in elevation between the front and rear of the train to determine gradient for the purposes of dynamic simulation of train performance.

Kilometrage is a reference value for the datapoint. The primary purpose of this value is to simplify cross-referencing of turnouts and speedboards to the track geometry. It is not used to calculate the distance travelled by the train. As such, it does not need to accurately match the physical distance between geographic points. Usually, it will be the kilometrage as assigned by the railway administration. This can include “long” and “short” kilometres.

Comments is an optional column provided for convenience. Any data in this column is ignored by Traxim.

The Traxim download includes the Traxim Centreline tool to assist with creation of track geometry files where track geometry data is not otherwise available in a clean and consistent format.

Appendix 6 – The Tractive Effort Curve Input File

Traxim uses a tractive effort curve to dynamically simulate the performance of the train. The input file format is as follows:

#Tractive Effort Input	
Name	SD40-2
Mass	166.9
Length	20
Vmax	104.6
endheader	
#km/h	kN Tractive effort
0	409
1	409
2	409
3	409
4	409
5	409
6	409
7	409
8	409
9	409
10	409
11	409
12	409
13	409
14	409
15	409
16	404
17	380
18	359
19	340
20	323
21	308
22	294
23	281
24	269
25	259
26	249
27	239
28	231
29	223
30	216
31	209
32	202

Name is a reference to locomotive type. This cross-references the Locos column in the Train Plan input file.

Mass is the gross weight of the locomotive in tonnes, and is used for calculating total train weight.

Length is the length of the locomotive in metres, and is used for calculating total train length.

Vmax is the maximum permissible speed of the locomotive in Km/H and will limit the maximum permissible speed of the train if it is less than the permitted train speed or the applicable speedboard.

Km/h (column A) is the reference speed for a tractive effort value.

kN Tractive Effort (column B) is the tractive effort in kilonewtons at the applicable speed.

Note that the speed / tractive effort data pair can be at any interval. Traxim will interpolate performance between the data points. In fact, it is only essential to have two data points, the starting tractive effort and any speed / tractive effort combination at a tractive effort below the starting tractive effort. Provided transmission efficiency holds constant, the complete curve can be inferred from those two data points.

A tractive effort file generator is available in the free Traxim support tools. This will assist in generating a tractive effort curve for any locomotive type you may want to incorporate into your simulation. A number of common North American and Australian locomotive types are prebuilt in the tool. Full details on how to use the generator tool are included in the 'How to" sheet of the tool itself.

Appendix 7 – The Regions Input File

The regions input file format is as follows:

# First row of input data should always be blank for region name, "White" for colour and "1" for Train Graph Order.								
# An unlimited number of train speed classes for nominal cant deficiency can be added from column H rightwards, but an empty column must be left between the last class and any column with non-input data.								
#Region Name	Colour	Train Graph Order	Opposing delay	Following Delay	Reverse Section	Nominal Superelev ation	Nominal cant deficiency	
	White	1	0	0	FALSE		Normal	Passenger
Portland OR - Longview WA	Magenta	2	90	90	FALSE	120	75	110
Longview WA - Seattle WA	Aqua	3	90	90	FALSE	120	75	110
Renton WA Triangle	SlateGray	4	90	90	FALSE	120	75	110
Black River Jct WA - Butte MT	LawnGreer	5	90	90	FALSE	120	75	110
Marengo WA - Spokane WA	Cyan	6	90	90	FALSE	120	75	110
Spokane WA - Plummer ID	Orange	7	90	90	FALSE	120	75	110
Cedar Falls WA - Everett WA	Peru	8	90	90	FALSE	120	75	110
Tacoma Jct WA - Tacoma Port WA	Tan	9	90	90	FALSE	120	75	110
Frederickson WA - Morton WA	PaleGreen	10	180	180	FALSE	120	75	110
Dishman WA - Metaline Falls WA	Magenta	11	180	180	FALSE	120	75	110
Centralia WA Power Station	SlateGray	12	90	90	FALSE	120	75	110
High Rock WA Quarry Balloon Loop	SlateGray	13	90	90	FALSE	120	75	110

Region Name is an unconstrained string field. It is suggested that you simply adopt a region naming convention that is meaningful for your circumstances.

Colour is a colour selected from the palate in Appendix 8.

Train Graph Order determines the order that the region will be displayed in the train graph output pdf. Regions do not need to be ordered by their train graph position within the regions file, but adopting this practice helps avoid omission and repetition errors.

Opposing Delay & *Following Delay* are standard time allowances added when a train needs to come to a stand due to a conflict. The key consideration in setting the delay time is the safeworking system. All safeworking systems have some amount of delay associated with determining that the section is clear, changing the turnouts and clearing the signals. Manual systems such as train orders or electric staff may also require an allowance for the train crew to undertake the safeworking activities. The default settings are 90 seconds for both opposing and following conflicts.

Reverse section reverses the direction in which the Region is displayed in the train graph outputs. By default, the region will be displayed in ascending kilometrage. Where trains pass between Regions and the kilometrage order changes, the display direction can be reversed to assist with visual continuity of the train. Direction can be toggled live in the Train Graph Viewer tool, so this setting is mainly useful where you are relying on the Train Graph pdf.

Nominal Superelevation and *Nominal Cant Deficiency* are cross-referenced with the track geometry file to auto-calculate a maximum implied speed at each data point. The can substitute for, or complement, the Speedboards input. Multiple train types can be specified for maximum cant deficiency. The order of the columns for nominal cant deficiency needs to follow the same ordering as the Speedboards file.

Note that the first region should always be left as blank, colour white, and train graph order 1. The blank region is used in the Infrastructure input file where a node exists in less than three Regions. The names of the train types for the maximum cant deficiency columns also need to go in this row.

Appendix 8 – Colour Palette

The following palette shows the colours available for use in Traxim, and their associated names: