PTV EPICS
the mind of movement

PTV EPICS

USER MANUAL

Copyright and legal agreements

Copyright

Legal agreements

The information contained in this documentation is subject to change without notice and should not be construed as a commitment on the part of PTV AG.

Warranty restriction

The content accuracy is not warranted. We are grateful for any information on errors.

Imprint

PTV AG
Haid-und-Neu-Str. 15
76131 Karlsruhe
Germany

Tel. +49 721 9651-300
Fax +49 721 9651-562
info@vision.ptvgroup.com
www.ptvgroup.com
www.vision-traffic.ptvgroup.com

Last amended: June 2015 EN-US F

Contents

1 Introduction

This document describes methodology and data provision for PTV Epics. The document is in a preliminary form and shall not be forwarded beyond the original recipients.

1.1 Motivation

To get the continuously increasing traffic problem – in the motorized individual traffic or private transport (PrT) as well as in public transport (PuT) - under control, more and more traffic lights are controlled by traffic dependent control systems. The traditional variant of such traffic signal controls is based on the 'impulse-reaction'-principle, i.e. the controlling algorithm consists of a sometimes quite complex sequence of if-then-queries, especially when Public-Transport prioritization is included.

More modern traffic-adaptive optimization procedures use iterative approaches. They use an integrated traffic model and try to estimate the conditions on the street as realistically as possible. Based on this input, the so called effect model calculates the characteristics considered in the optimization process for various control alternatives, e.g. the waiting times of each vehicle. In this case, the solution with the lowest total waiting time is accepted and handed over to the traffic-light system.

Among the traffic adaptive procedures, there are two different types:

- Network-control methods, i.e. systems which consider more than one traffic light, e.g. PTV Balance (Braun and Kemper 2008).
- Local techniques, restricted to only one intersection.

An example for the latter variant is the adaptive local signal control PTV Epics.

1.2 History and General Features

In the second half of the 1990s, Dr Joachim Mertz developed PTV Epics' traffic-model based adaptive traffic-light control method (Mertz 2001) within his PhD at the Technical University of Munich. The essential features of this program are:

- Local treatment, i.e. each intersection is controlled separately
- Mainly identical traffic modelling of individual and public traffic flows
  ⇨ consistent interface to the effect model
  ⇨ integrated optimization of the different types of traffic
- Possibility of Public-Transport prioritization by higher weighting of the Public Transport compared to the Individual Traffic; the weighting factor is configurable[1]
- Low effort due to self-calibrating algorithms
- Integration into a superior controlling system (e.g. PTV Balance - see also PTV AG 2015c) possible

[1] A typical ratio between the weight of a PuT to an PrT vehicle is 100:1.

- Optimization in real time, which means a control precision of one second
- Low detection effort: only one sensor per lane in 0-100m distance to stop line necessary
- Ready for the future: Inclusion of Car2X-information possible

2 Functionality

2.1 Type of Control

The quality of a traffic-light control, and thereby the influence on the quality of the traffic flow, directly depends on the control method. In general, fixed-time controls and traffic dependent controls are distinguished according to RiLSA (2010). Depending on the area of operation and the traffic-related and political framework conditions, the demand of complexity and flexibility of the type of control may vary.

Fixed-time control

One of the simplest control methods is the fixed-time control.

‘Fixed-time signal programs are sufficient to the needs according to local and traffic-related requirements. […] Fixed-time signal programs should be applied if the traffic volumes are predictable over long terms, because the signal-program elements do not change’. (RiLSA 2010)

It is a very strict form of controlling. Because of the stochastic variability of traffic volumes and the prioritization of the Public Transport, the fixed-time control cannot reach a sufficient quality of the traffic flow in cities. Therefore, this type of control is not applied very often any more in modern traffic-light systems.

Rule-based and model-based traffic dependent control

Besides the fixed-time control, two types of traffic dependent controls are distinguished according to the guidelines for traffic light systems (RiLSA 2010): rule-based and model-based traffic dependent control methods.

‘Often, the rule-based control methods run through a flow diagram every second. This is based on conditions and actions and leads to decisions for the signal program generation according to the current control states and parameters (see Figure 1). Reference and marginal values like the time gap value, the saturation and the travel time or the speed, respectively, but also framework requirements like the allowed period of release time or the delayed start of release times are defined by parameterization.’ (RiLSA 2010)

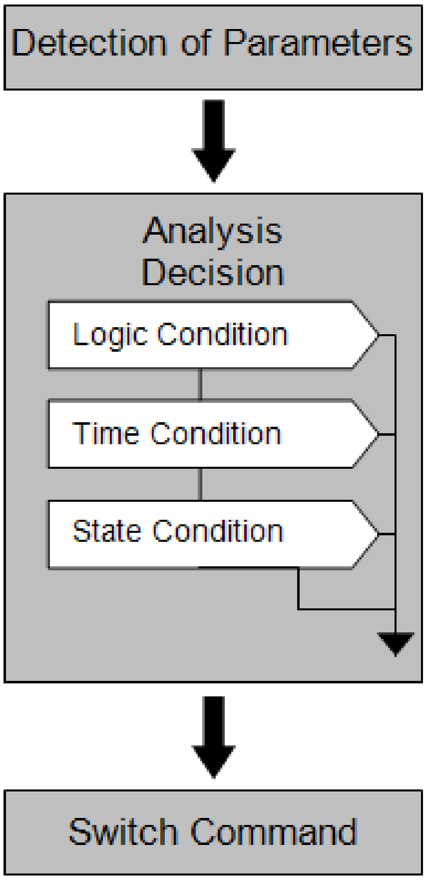


Figure 1: Rule-based control method (RiLSA 2010).

‘The model-based control method is not directly based on the measured parameters, but on values which are processed further in a model. Besides the traffic model, the model-based control method also requires a control model which describes the degree of freedom of the control. An optimization algorithm tests possible control adjustments and evaluates them on the base of the modelled effects according to the target function (see Figure 2). A model-based control method is formulated universally for intersections or traffic networks, respectively, and is adapted to the individual environments via configuration files’. (RiLSA 2010)

PTV Epics is a model-based traffic adaptive system. Therefore it is assigned to the second group of controls.

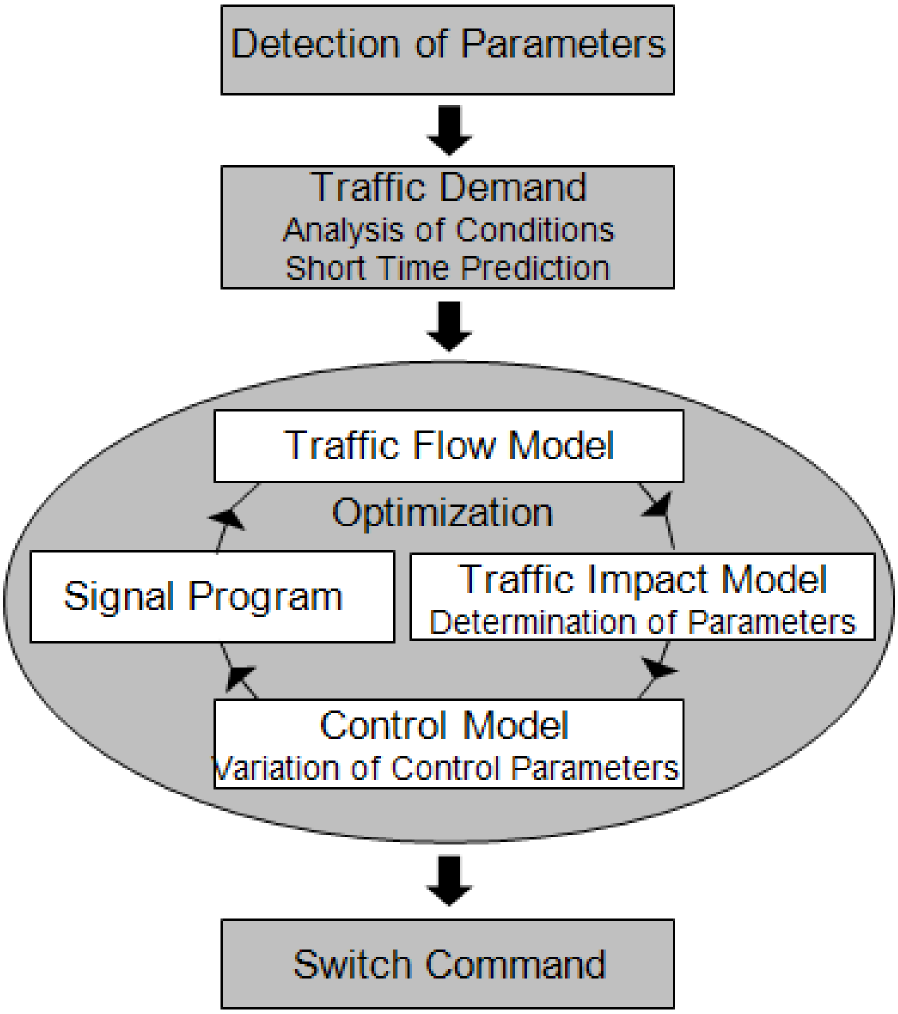


Figure 2: Model-based control method (RiLSA 2010).

2.2 Objectives of PTV Epics

In the German-speaking area, model-based control methods are mainly used for network-control methods (e.g. PTV Balance). Local intersection control systems have different conditions with respect to the performance than the network-control methods. First of all, this has an impact on the time which is available for the optimization. PTV Epics is a real-time system, which means it is able to run through its whole optimization routine in one second if integrated in a modern controller. Therefore, the following objectives can be achieved:

Adaption of the control to the current traffic situation

- In contrast to the typical rule-based controls PTV Epics achieves an improvement in quality of the optimized traffic flow for all types of traffic.
- The integrative and uniform approach of PTV Epics considers within the traffic modelling all vehicles that are relevant and can be measured automatically. This means that the traffic demands at the intersection are always considered as a whole

during the optimization. This is the only way to reach a highly effective and a flexible, controllable realization of the target function in form of a performance index.

- In combination with a superior network-control method like PTV Balance, also superior objectives of the control, like the coordination of multiple traffic lights or the network wide guidance of the traffic are possible.
- The forecast horizon of PTV Epics (100s) is sufficient to consider all relevant traffic demands.

Definition of traffic-related and political framework conditions

- Possibility to prefer particular traffic flows by the individual weighting of the target function.
- Integration into a control technical total approach of the coordination by specifying a reference signal plan. This can be generated offline or online by a superior network control.
- Besides the influence through the target function, there are other possibilities available for the specification of hard boundary conditions in order to reduce the degrees of freedom of the local control and to guarantee a minimum comfort. Examples are minimum or maximum stage durations.

Technical-organizational framework conditions

- No special controller hardware is required
- The control works completely autonomously and may run independently of external systems
- The extent of the personnel efforts, especially for experts, which are necessary for providing the whole control definition (parameters, framework conditions, etc.) is smaller than the whole effort of typical rule-based logics, especially if PuT prioritization has to be considered.

2.3 Dynamic Input Data

PTV Epics requires situation data from the control device every second in order to be informed about the traffic situation. These data have to be queried, even during an interstage when PTV Epics cannot optimize, in order to keep the program up to date. Therefore, PTV Epics needs a suitable interface to the control method on which PTV Epics is based, i.e. e.g. TRENDS or VS-PLUS or RBC, to be able to execute the query. The following data have to be available:

1. The current signal image. PTV Epics identifies the stage according to the signal image. If no stage can be identified, no optimization takes place. Furthermore, the image is required for modelling the so called ‘Vertical Queues’ (see Chapter 2.5.1), which are the queues in front of the signal groups. In order to receive the signal image, different functions for the query of the state (free, blocked, transition) of each signal group have to be available.

2. The current signal program. It is required to determine the current cycle length and also for the program dependent stage parameters.
3. The current cycle second. It is the fundamental time reference of the control device. Among other purposes it is required for the consideration of static or dynamic stage limits.
4. Detector events and edges, respectively.
5. Detector occupancies. If a detector is occupied for more than 4 seconds, PTV Epics assumes that the queue ranges beyond the detector and the tailback estimator (see Chapter 2.5.1) starts to work.
6. Detector failures. Detector failures are a critical problem for PTV Epics because the program cannot work correctly without exact information about the detector events. PTV Epics tries to create a substitute value in case of a detector failure, for example by using the values of an adjacent detector. If this is not possible, PTV Epics reports that it cannot optimize.

2.4 Definition of Detector Positions

Every modern traffic dependent control method requires stationary measuring devices to measure the traffic flow. In contrast to the rule-based traffic dependent control methods which request certain signal states basing on their inherent logics and rules that usually respect time gaps and/or detector occupancies, the model-based traffic adaptive control method PTV Epics optimizes on base of the picture of the current traffic situation that it has from status changes of the detectors and of the various signal groups (see also Chapter 2.5).

PTV Epics can map the current traffic situation in a better quality than the usual rule-based traffic dependent control systems, given the same number of detectors, due to its integrated traffic model. Therefore one does not need more devices for measuring the traffic situation than in rule-based methods.

The program PTV Epics is able to work with every possible detector position, concerning the distance to the stop line. Ideally the distance should be about 50 to 80m corresponding to about 4 to 6s in the city, so that the traffic light system can react to an approaching group of vehicles. This detector position can be compared to the typical detectors which are far from the stop line. Of course, the estimation of the queue length is also easier between the stop line and the detector than before the detector. Greater distances are not necessarily better because the driving behaviour changes with rising distance and therefore the quality of the queue-length estimator gets worse (see also Chapter 2.5.1).

Furthermore, the data base can be complemented by detectors close to the stop line. These are not necessarily required but PTV Epics uses them to correct the queue length if it is obviously overestimated in the PTV Epics picture[2]. Nevertheless, the most important

[2] If a detector, which is close to the stop line, is not occupied for more than five seconds and the queue length in the Epics picture reaches beyond this detector, the queue length is cut off at its distance from the stop line.

detectors for the quality of the PTV Epics optimization are the ones which are far from the stop line.

2.5 Traffic Model

2.5.1 Motorized Individual Traffic and Cyclists

For the Individual Traffic a deterministic model similarly to the Transyt model has been developed for PTV Epics. In this model the progression speed of the vehicles is assumed to be constant, i.e. after detection the vehicles are moved second by second by a constant range towards the stop line. There in case of Red they pile to a so called 'Vertical Queue', i.e. the tailback has no spatial extent. During Green PTV Epics calculates with an outflow of the vehicles in saturation flow (typical value: 1veh/2s).

In case of queuing beyond the detector – especially often with detectors close to the stop line – a queue-length estimator according to the method of Liu et al. (2009) is used. This technique identifies during Green the queue length beyond the detector and predicts this value for the next cycle. It is also used for online calibration by adapting the saturation flow dynamically.

Even if the distance between detector and stop line is several hundred meters, the detected inflows alone would not suffice to cover the time horizon of PTV Epics. Therefore an inflow prognosis via cyclic flow profiles is used to estimate the arrival of vehicles during the next cycle. This way a green wave is generated automatically, for PTV Epics expects that within the next cycle the bunch arrives at the same time as in the current one[3].

Traffic demands due to detected cyclists can be treated in complete analogy to the Individual-Traffic vehicles. The only difference to PrT is that one should configure a shorter time requirement for the cyclists as they usually set up in a denser way in front of the stop line.

2.5.2 Public Transport

Unlike PrT streams, Public-Transport vehicles usually appear discretely at the crossing. Therefore and due to the desired possibility of prioritization the vehicles have to be considered as single events. The differences in the arrival time of the vehicles that occur due to occasional disturbances are respected by not considering them as point like but as a trapezoidal probability distribution, see Figure 3 a). A PuT vehicle is generated in the very moment when it passes the first calling point. Just like the PrT vehicles these PuT profiles are shifted towards the stop line every second. As long as the vehicle did not deregister the probability has to be conserved, i.e. the trapeze is compressed in front of the stop line (Figure 3 b)-d)).

[3] In case the upstream traffic light is highly traffic dependent this way a green wave can only be accomplished to a limited grade as the inflows vary a lot from cycle to cycle.

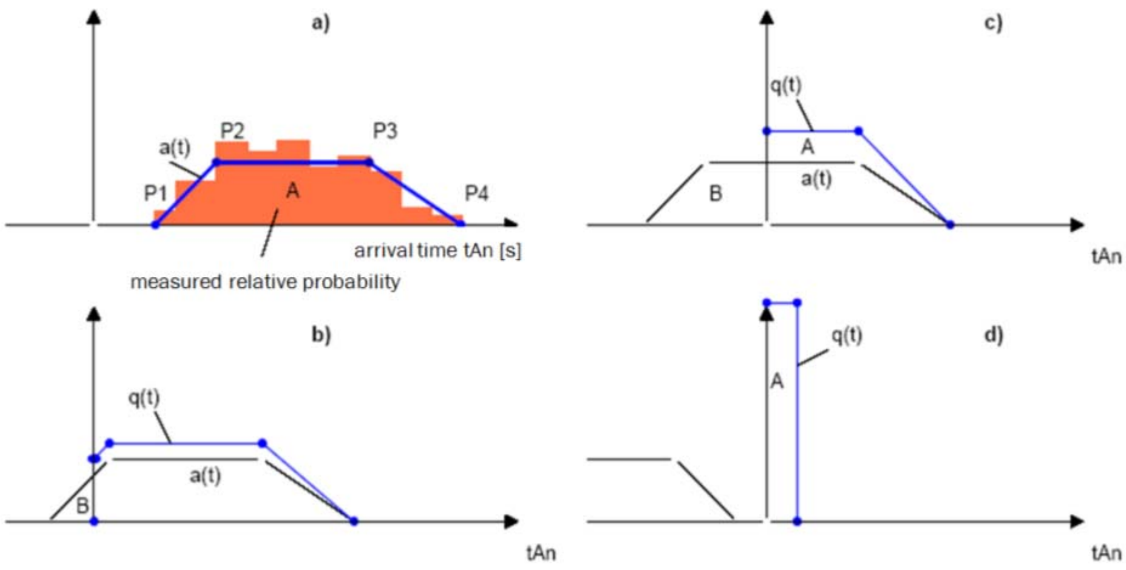


Figure 3: Display of a PuT vehicle moving towards the stop line

If a (prioritized) PuT vehicle without particular lane (e.g. a bus) approaches the stop line and if there is a tailback in front of it, the bus is promoted within the PTV Epics picture by the time distance necessary to remove the queue. PTV Epics then will make an effort to turn the relevant signal group to green even before the actual arrival of the bus, so that the vehicles in front of it flow off in time and do not hinder its crossing.

The PuT profiles can be treated within the control exactly like the flow profiles of the PrT model. Thus a consistent treatment of the different kinds of traffic, one of the fundamental goals of PTV Epics is granted.

2.5.3 Pedestrians

PTV Epics is able to consider pedestrian demands as well. Within the PTV Epics picture the pedestrians wait right in front of the stop line. In case of Green all of them ‘flow off’ simultaneously. Since a pedestrian push-button has no information about how many persons want to cross, it is advisable to weight such a demand stronger than a PrT vehicle.

2.6 Optimization Function

2.6.1 Calculation of the Performance Index

The relevant quantities entering the target function of the optimization of PTV Epics are the total delay and the number of stops of the vehicles, which are calculated by summing up all detected traffic streams each with a configurable weight. Today’s requirements for signal control systems are in addition to the objectives of transport planning often politically motivated and therefore vary strongly from one city to the other. For this reason PTV Epics provides the opportunity to directly adapt the influence of the number of stops via a weight factor which has to be configured in the traffic-demands window. In case this factor is zero (which is the default value at the moment) the optimization function will only consider the waiting time. On the other hand, if the factor is e.g. 10, the number of stops is weighted ten times higher than the waiting time. Nevertheless, the most important factor is still the waiting time, because one stop means an average of more than 10 seconds delay.

Additionally, there is the possibility to compare with a reference signal plan in order to guarantee coordination of the local control with the surrounding network as good as possible. It can be provided offline by a traffic engineer or generated online by means of a suitable network control such as PTV Balance. Together with the delays the Performance Index PI results in

$$PI(sp) = \sum_{sg \in SG} \alpha_{sg} D_{sg}(sp) + \beta\, \Delta(ref, sp)$$

with

SG	set of signal groups
sp	signal plan to be valued
ref	reference signal plan, e.g. from PTV Balance
α_{sg}	weighting of the signal group sg
D_{sg}	sum of delay at signal group sg over time horizon considered
Δ	deviation of control alternative sp from ref
β	weighting of deviation from ref

If the stops are to be respected, an analogous sum over the number of stops per signal group has to be added.

The waiting times are calculated as difference between inflow and outflow of each signal group. In doing so one assumes that during Green the vehicles flow off with the saturation flow calculated by PTV Epics. Therefore, the position and length of the green times of each signal group is decisive for the value of the PI, since the inflow is fixed from the outside. The manner of respecting the frame signal plan is sketched in Figure 4.

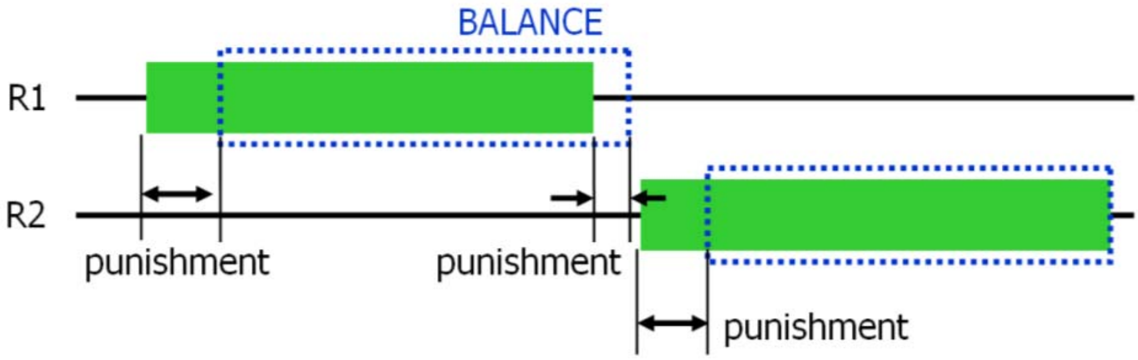


Figure 4: Influence of a PTV Balance frame on the PI

Strictly speaking there are further parameters that influence the PI. It is e.g. possible that a signal group may only be Red for a certain configurable time span. A possible violation of such a time by one of the signal plans considered during optimization is punished severely. This and other ‘hard’ constraints are explained in Chapter 3 in further detail.

2.6.2 Control Model

PTV Epics works stage- and interstage-based. That means it is the task of the control model to calculate for the given time horizon T a stage sequence that minimizes the performance index. For achieving this objective PTV Epics runs through the various ways of the phase space, spanned by the provided stages and time, Figure 5. The result of each run is a signal plan for the next T seconds, i.e. a precise description of the interstages to be actuated, which also defines the green times exactly. These again determine the possible outflow of the vehicles and thus over the expected delays and stops and the resulting PI. The signal plan with the lowest PI is accepted and transferred to the traffic-light control. The optimization runs every second, but only while no interstage is running, since during this time the signal order is fixed. Therefore at the moment it is not possible for PTV Epics to start an interstage while another one is running. A rudimentary extension to that effect has already been implemented. Furthermore, this can be accomplished via an external logic, e.g. in TRELAN.

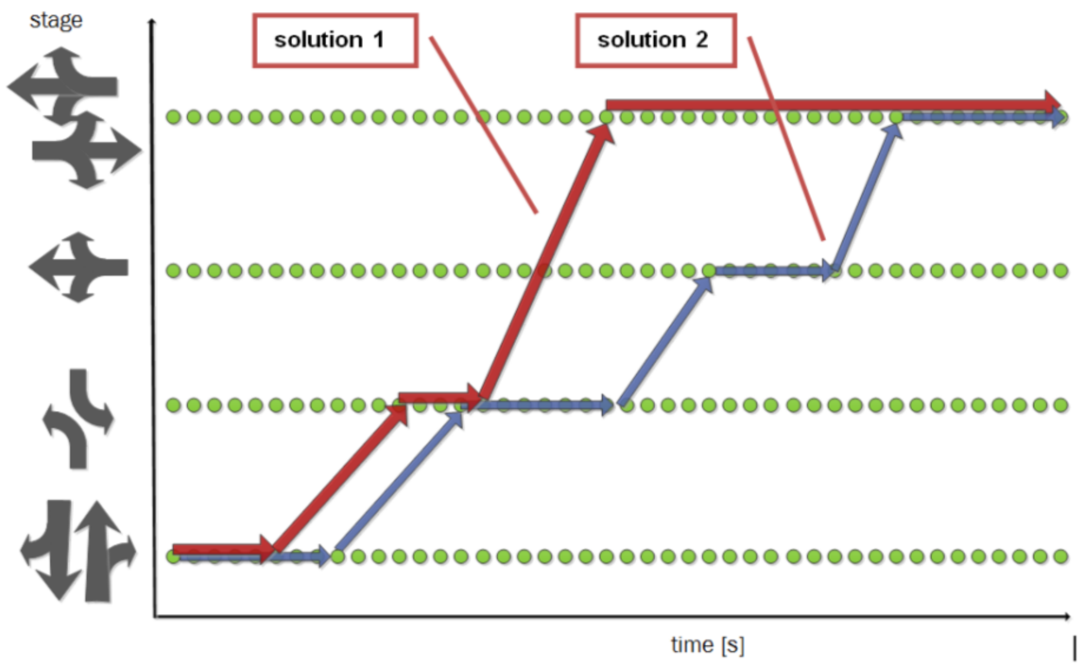


Figure 5: Searching through the phase space for the optimal way

For the sake of completeness one should mention that for performance reasons the optimization of PTV Epics works in two steps: In the first step the time horizon is split into a grid with 5 seconds width. Here the stage sequence in principle is set. In the second step the fine tuning takes place, i.e. the starting times of the interstages are optimized with one second precision. The first step uses a (slightly modified) branch-and-bound algorithm, the second one an ordinary hill-climbing algorithm. This approach reduces the originally NP-complete problem such that the required computational time scales only linearly with the number of degrees of freedom (number stages and interstages), cf. Mertz (2001).

2.7 Dynamical Output Data

After successful optimization PTV Epics has to communicate its switching recommendation to the underlying control system, e.g. TRENDS or VS-PLUS. In combination with a logic based control this can be accomplished via one single variable. As an example we describe the integration into the TRENDS system. The PTV Epics-output is stored in a variable named v42 in Figure 6. This variable is the return value of the TRENDS subprogram epics(), see Figure 7. In case it is 0 PTV Epics does not want to start an interstage at the moment. A value > 0 indicates the ID of the interstage to be started. In the example below e.g. v42 = 11 means ‘start interstage 11’. In case the optimization was not successful, e.g. due to the failure of a detector that cannot be replaced, PTV Epics returns -1. In this case, the standard reaction of the logic is to fall back to a fixed time control which has to be defined via T-times, see Figure 6. Possibly besides the time bracket one or more green times have to be queried in order to eliminate the possibility of a minimum green-time violation. These and other security queries have been omitted for clarity here.

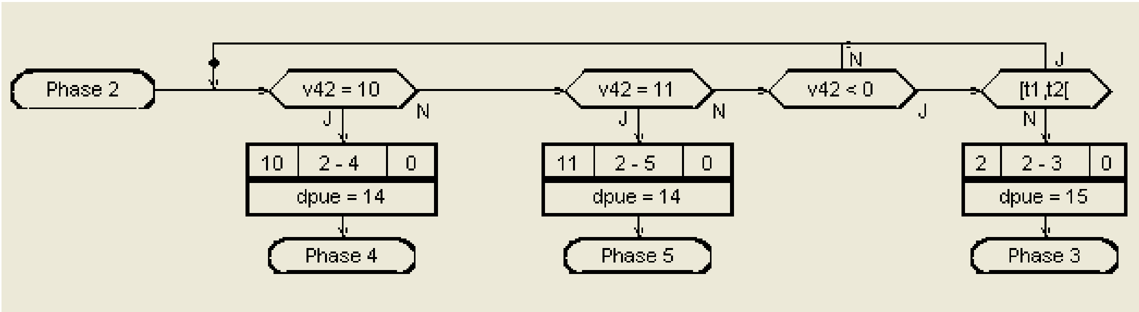


Figure 6: Query of the PTV Epics output within TRENDS logic

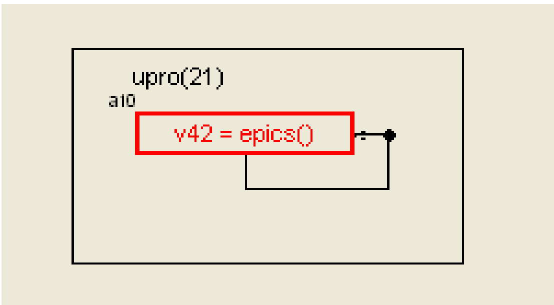


Figure 7: Structure with return value for v42

One also recognizes from Figure 6 that it is possible without any problem to add further details to the PTV Epics structures. E.g. there might be two variants of interstage 11: one with and one without advance signal, according to whether a bus did register or not. A simple if-then-request suffices to distinguish these cases, so that such decisions do not have to be made by PTV Epics. The fact that the interstage runs slightly different from what PTV Epics expected only plays a minor role. As soon as the subsequent stage is recognized (stage 5 in the example) PTV Epics starts to work again.

Although PTV Epics is stage based it can collaborate with signal-based control methods like VS-PLUS. In this case the signal plan calculated by PTV Epics is translated into VS-PLUS understandable frames. The interface needed for this conversion already exists and has been applied successfully in the cities Vienna and Salzburg.

In a comparable way PTV Epics can be integrated with RBC controllers using NTCIP. In that case PTV Epics translates its optimal stage sequence e.g. into hold and force-off commands.

2.8 Areas of Application

PTV Epics can in principle be installed at any signalized intersection. Although PTV Epics is a model-based adaptive method for single intersections, it provides many possibilities to use it in coordinated streets or complex road networks as well.

By taking into account external constraints in the optimization process of PTV Epics it is possible to coordinate several intersections and thus to set up green waves. However, when PTV Epics is to be used in complex road networks, the additional usage of a traffic adaptive network-control system like PTV Balance does make sense. Within the combination of both systems (PTV Balance and PTV Epics) PTV Balance optimizes the traffic flow with bird’s eye view of network-wide objectives by constructing a signal plan framework in a macroscopic way. With its microscopic optimization model PTV Epics is in the position to carry out an adaptive control with respect to the statistical fluctuation of traffic near the intersection and implement prioritization for Public Transport. This combination of two model-based adaptive optimization methods at the microscopic and macroscopic level is unique. Many adaptive network methods for signal control currently use model-based optimization at the macroscopic level however the microscopic level is usually controlled by rule-based signal control methods. A combination of both methods promises the exploitation of much of the currently unused potential of the road network.

PTV Epics is a real-time process that has to be able to make a complete run every second. Like any rule-based method, the model-based method PTV Epics needs modern control devices with enough computing power. Furthermore it shall be mentioned that PTV Epics reproduces the traffic flow by using a traffic model. Therefore stationary detectors like inductive loops are mandatory (see Chapter 2.4).

In general it may be summarized that the application area of PTV Epics is wider compared to conventional rule-based methods due to the fact that it has a model-based optimization of traffic demands. PTV Epics thus offers benefits that cannot be accomplished by conventional methods of signal control at single intersections.

3 Data Provision

The PTV Vision suite allows data provision for PTV Epics in a seamless way for simulation and calibration. PTV Epics data provision consists of the following parts:

- Parameters of PTV Epics and signal control data
  - Local parameters e.g. signal control data and parameters that are intersection related
    The signal controller “Epics/Balance-Local” is used to provide signal control data. This signal controller is available in PTV Visum and PTV Vissim. Vissig, PTV Balance and PTV Epics use the same sig-file format that can be shared between the different models.

For details regarding PTV Vissim, PTV Visum or PTV Balance please refer to their respective manuals (PTV AG 2015a/b/c).

Hint: PTV Balance is a network control that optimizes signal plans typically every five minutes. It has local and global parameters. It requires a local signal controller to apply the optimized signal plans. Therefore, PTV Balance is modelled with two types of signal controllers - “Epics/Balance-local” and “Balance-Central”. The local controller “Epics/Balance-local” can accomplish several tasks. It can be a pure executor of PTV Balance, it can be the local traffic signal optimization PTV Epics and it can be the combination of both.

The data provision of PTV Epics is very simple due to several self-calibrating algorithms. It is possible to provide input for PTV Epics without PTV Visum - which helps setting up large networks in PTV Vissim and is not directly required for PTV Epics - and also without PTV Vissim, though applying signal controls in the field without testing in a simulation is only advisable for experienced users and on isolated intersections.

For guidelines on using PTV Epics and PTV Balance within the PTV Vision suite, please consult the PTV Balance manual (PTV AG 2015c).

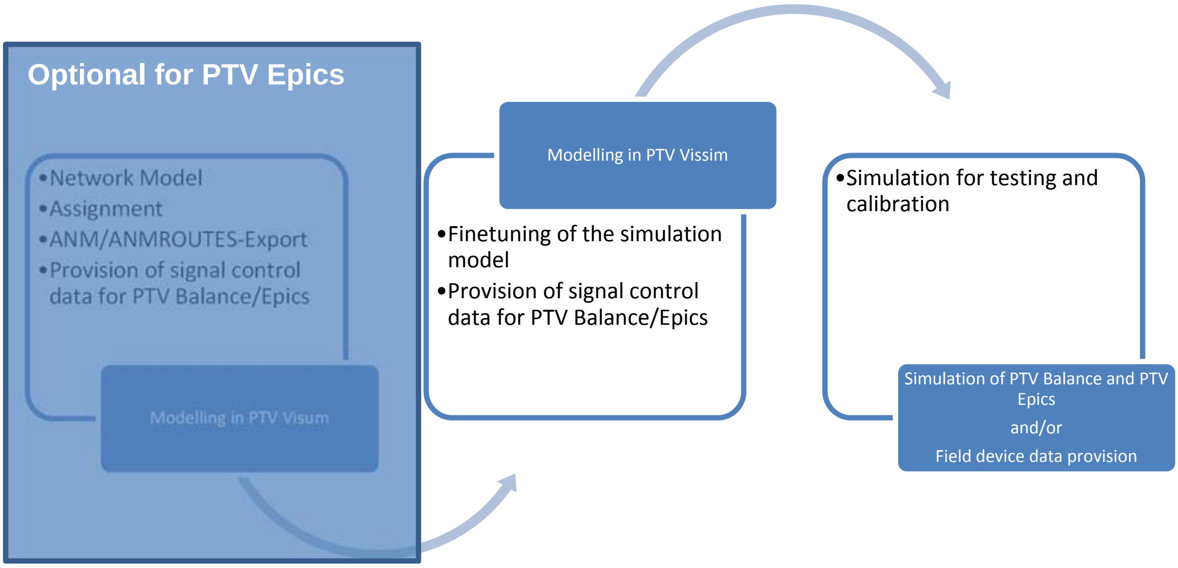


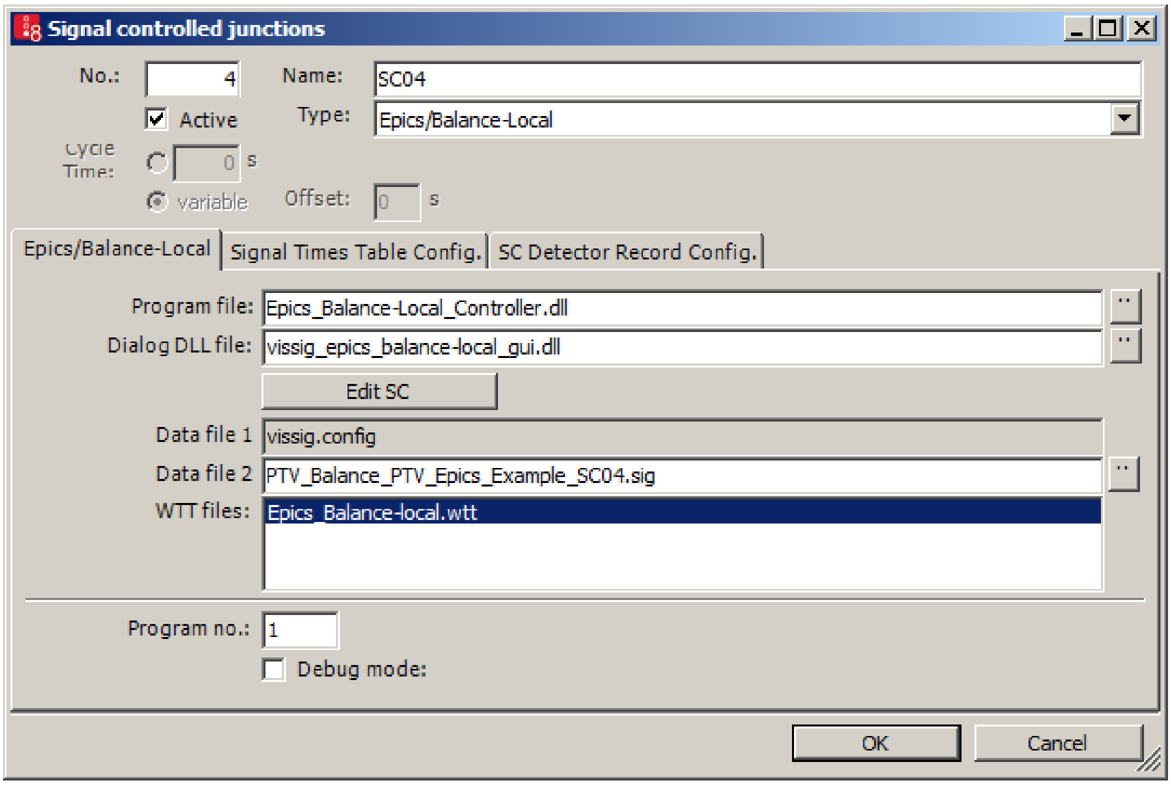
Figure 8: Workflow for data provision in the PTV Vision suite.

3.1 PTV Epics Parameters and Signal Control Data

PTV Epics related parameters are provided with the GUI of the signal controller **Epics/Balance-local** that is an extended version of Vissig. This manual will only address additional data that are relevant for PTV Epics. Sections that are not covered are not required for PTV Epics.

PTV Epics requires a stage based design.

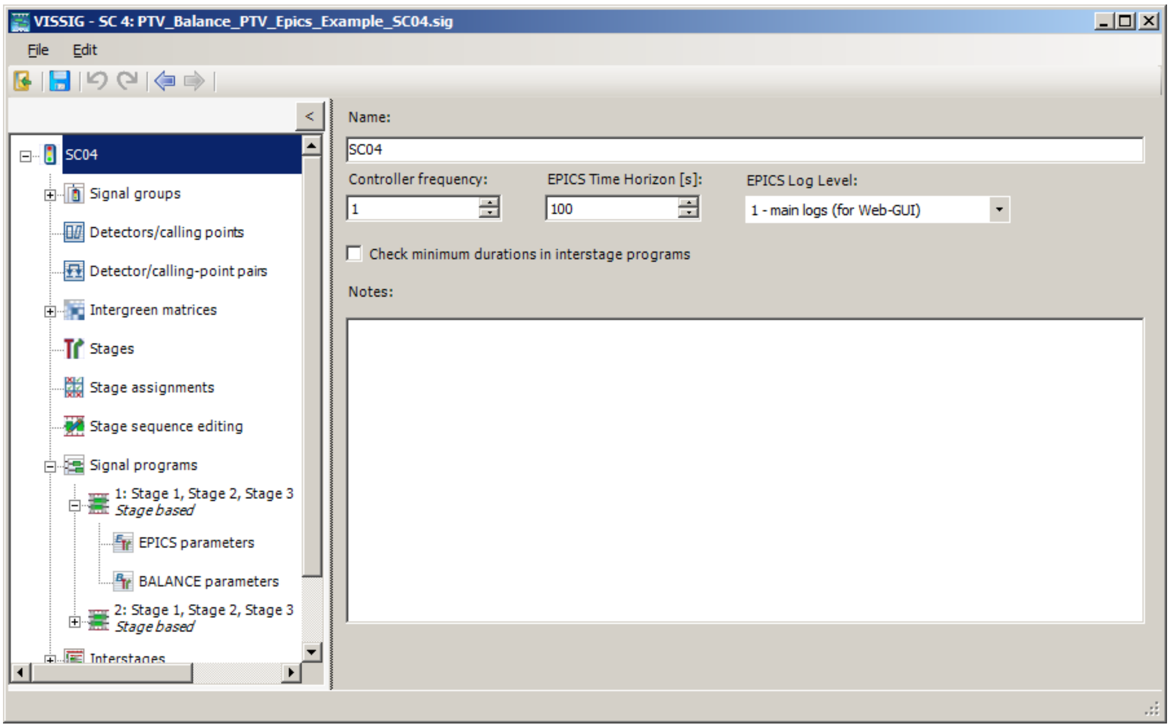
1. From the **Signal Control** menu, choose > **Signal Controllers**.
   *The **Signal Controller** list opens.*
2. Right-click the entry of your choice.
3. From the shortcut menu, choose **Edit** or **Add**.
   *The **Signal Control** window opens.*



4. In the **Type** field, select > **Epics/Balance-Local**
5. Edit the desired data:

Element	Description
Debug mode	If active, then the local controller creates log-files (see chapter 4.2). Detailed log-files are created in the subfolder Epics_Log of the directory of the inpx-file. The type of log files that are created can be set in the signal controller (see chapter 3.1.1). This significantly increases runtime.
All other elements	Please refer to help on Fixed Time control.

6. Click on **Edit Signal Control**.
   The SC Editor opens.



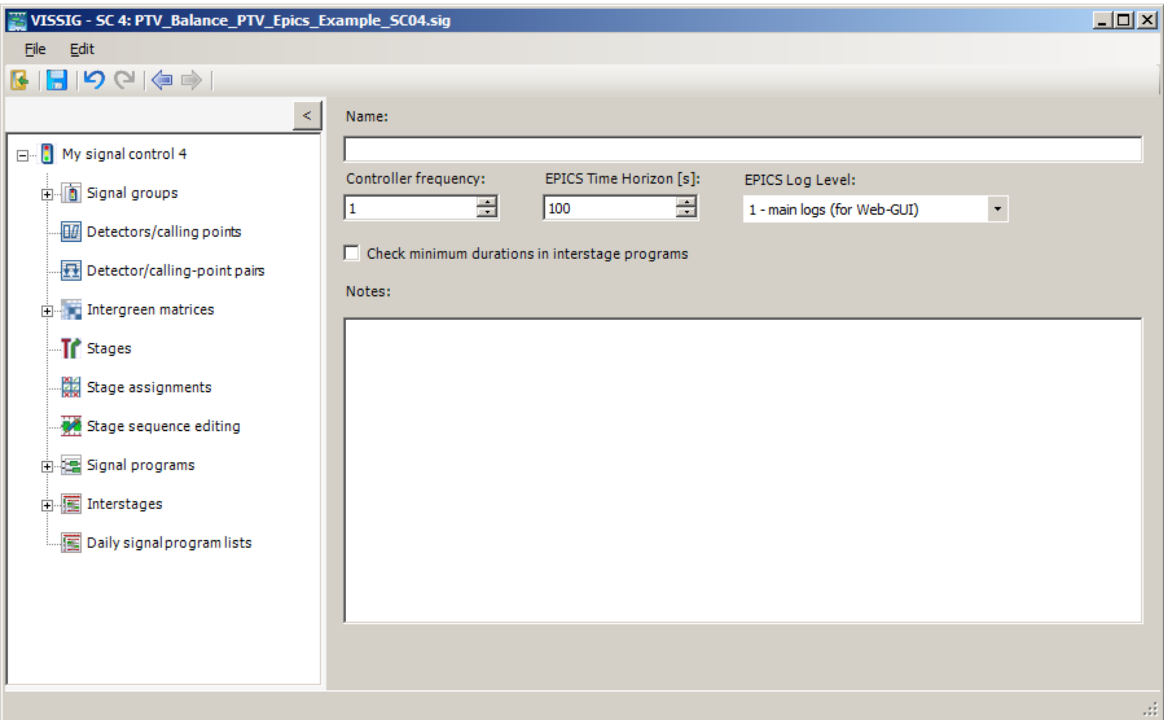
Hint: Every parameter for PTV Epics comes with a tooltip providing an explanation of its meaning. In addition, there are numerous plausibility checks with explanations.

3.1.1 Signal Control

As required for any stage based fixed time control modelled with Vissig.

Additionally PTV Epics requires the provision of the parameters described below.

1. Click on **My signal control X**.



2. Enter the desired data:

Element	Description
EPICS Time Horizon [s]:	The time horizon of the PTV Epics optimization. The time horizon should be long enough to consider all demands with respect to the given boundaries (like maximum/minimum green and red times). Under typical conditions (e.g. cycle times <=120s) the default value of 100s is sufficient.
EPICS Log Level:	This defines the type of log-files that are created by PTV Epics (see chapter 4.2) as follows: - 0 - no logs: no log files are created - 1 - main-logs (for Web-GUI) Epics_MainLog_xxx.txt, Epics_MainLog_xxx.json, Epics_Queues_xxx.json These are required for the usage with the Balance-Web-GUI - 2 - base result-logs plus Epics_Supply_xxx.txt, Epics_Queues_xxx.txt, Epics_Action_xxx.txt, Epics_StageSequence-1_xxx.txt, Epics_StageSequence-2_xxx.txt - 3 - extended result-logs plus Epics_NumberStops-1_xxx.txt, Epics_NumberStops-2_xxx.txt, Epics_PerformanceIndex-1_xxx.txt, Epics_PerformanceIndex-2_xxx.txt - 4 - all logs plus Epics_Debug_xxx.txt, Epics_Runtime_xxx.txt, Epics_APValues_xxx.txt

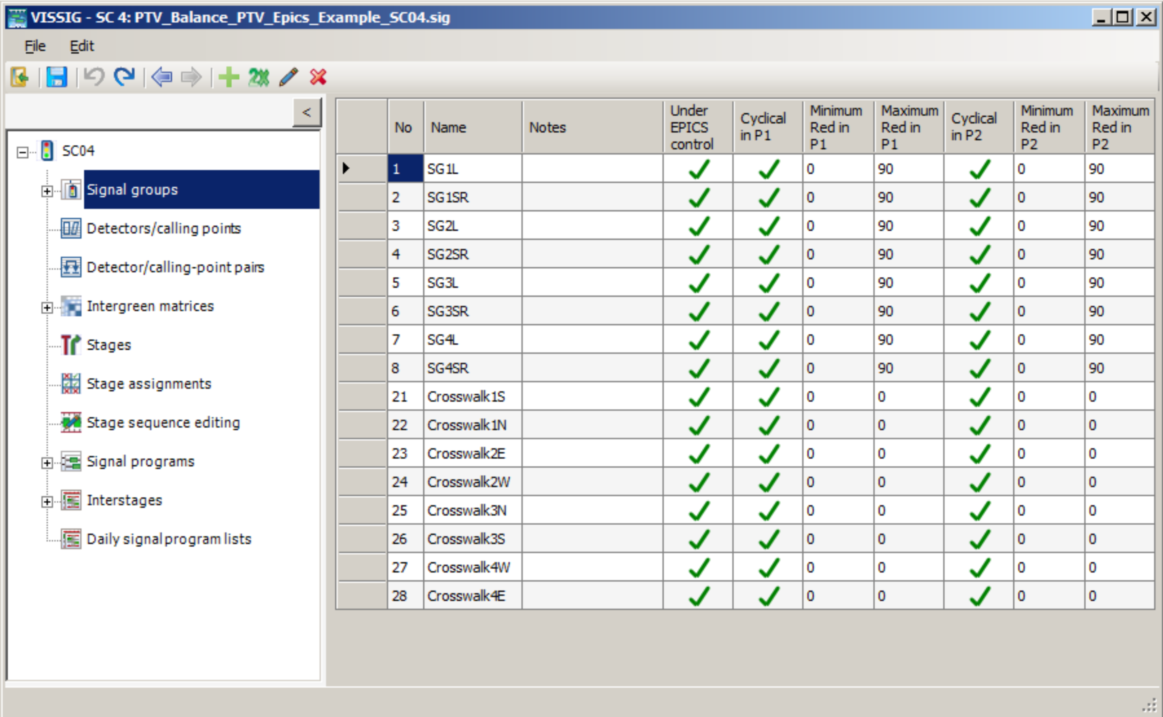
3.1.2 Signal Groups

As required for any stage based fixed time control modelled with Vissig.

Additionally PTV Epics requires the provision of signal group dependent parameters and traffic demands that connect signal groups to detectors or calling points.

3.1.2.1 Signal-group Parameters

1. Click on **Signal groups**.



No	Name	Notes	Under EPICS control	Cyclical in P1	Minimum Red in P1	Maximum Red in P1	Cyclical in P2	Minimum Red in P2	Maximum Red in P2
1	SG1L		✓	✓	0	90	✓	0	90
2	SG1SR		✓	✓	0	90	✓	0	90
3	SG2L		✓	✓	0	90	✓	0	90
4	SG2SR		✓	✓	0	90	✓	0	90
5	SG3L		✓	✓	0	90	✓	0	90
6	SG3SR		✓	✓	0	90	✓	0	90
7	SG4L		✓	✓	0	90	✓	0	90
8	SG4SR		✓	✓	0	90	✓	0	90
21	Crosswalk1S		✓	✓	0	0	✓	0	0
22	Crosswalk1N		✓	✓	0	0	✓	0	0
23	Crosswalk2E		✓	✓	0	0	✓	0	0
24	Crosswalk2W		✓	✓	0	0	✓	0	0
25	Crosswalk3N		✓	✓	0	0	✓	0	0
26	Crosswalk3S		✓	✓	0	0	✓	0	0
27	Crosswalk4W		✓	✓	0	0	✓	0	0
28	Crosswalk4E		✓	✓	0	0	✓	0	0

2. Enter the desired data:

Element	Description
Under EPICS control	Information whether the signal group shall be controlled by PTV Epics or not. This is e.g. not true for all signal groups if an intersection is divided into two partial intersections and PTV Epics only controls the main part. This check box also decides whether the signal group is taken into account in determining the current stage or not.
Cyclical in P1	This defines whether the signal group will become green every cycle or only on demand. In general, a signal group that is used for PuT prioritization is not cyclical in P1. This parameter is available for every stage based signal program. PX represents the number of the signal program.
Minimum Red in P1	Time span in seconds before which the signal group may not get green. This parameter is available for every stage based signal program. PX represents the number of the signal program.
Maximum Red in P1	Time span in seconds after which the signal group must get green, starting either with the end of last green (cyclical) or with a demand (non-cyclical). This parameter is available for every stage based signal program. PX represents the number of the signal program.

3.1.2.2 Traffic demands

Overview of traffic demands

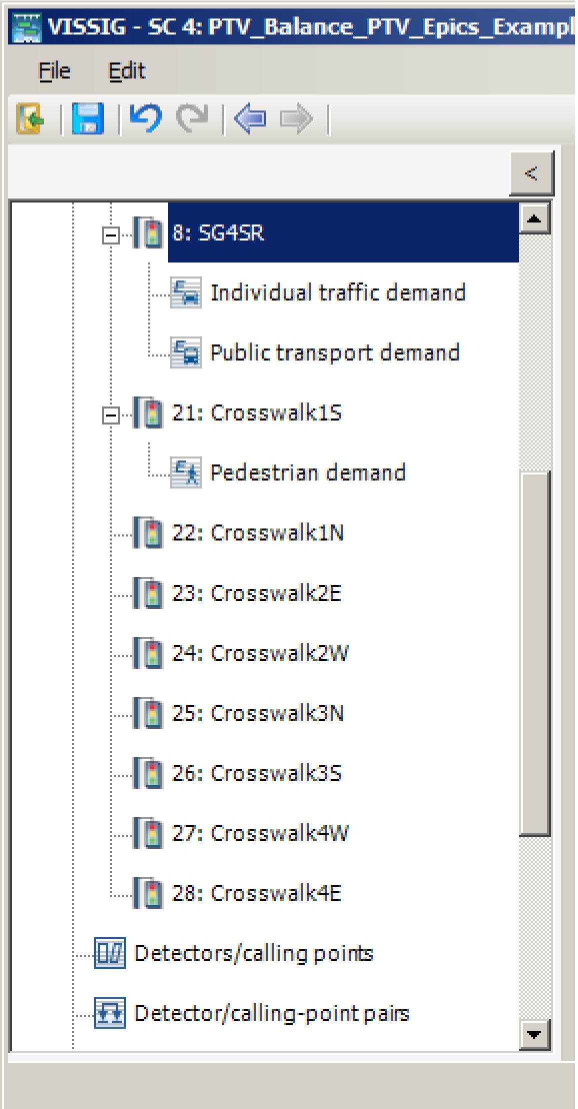
You have to assign Traffic Demands to the signal groups.

Kinds of traffic demands

The following different kinds of traffic demands are distinguished:

- individual (PrT)

- public (PuT)
- pedestrian (Ped) demands



Using weights for traffic demands

Set a Weight for every traffic demand. The standard weight of a PrT demand is 1, PuT vehicles can be prioritized via this factor. E.g. Weight 100 means that every PuT vehicle weighs 100 units, i.e. in case of a PrT weight of 1 it weighs as much as 100 PrT vehicles.

Usually the weight of a pedestrian demand is chosen larger than that of a PrT demand (e.g. five times larger), because the pedestrian demand is independent of the (unknown) number of waiting pedestrians.

Using detectors and/or calling points and detector and/or calling point pairs

Detectors and/or calling points are required for measuring the traffic. Detectors and/or calling points are contained in the table **Detectors/calling points**.

Detectors and/or calling points can be used to define detector and/or calling point pairs, which are used for PuT prioritization. Detector and/or calling point pairs are contained in the table **Detector/calling point pairs**.

If these measuring devices are not supplied yet, this has to be done now, because otherwise they cannot be assigned to the traffic demands.

> Hint: PTV Vissim and Epics/Balance-Local have a simple one-way-synchronization of detectors from PTV Vissim to Epics/Balance-Local. This means that you should provide detectors in PTV Vissim first and then upon opening the GUI of Epics/Balance-Local the detectors that are associated with the given signal controller are synced to Epics/Balance-Local. This process greatly reduces effort and the possibility to provide false channel numbers.
>
> The sync will delete detectors in Epics/Balance-Local that do not exist in PTV Vissim. If that happens accidently, do not save the sig-file after opening the GUI of Epics/Balance-Local.
>
> This does not apply when Epics/Balance-Local is used with PTV Visum.

Adding a traffic demand

Since Epics/Balance-Local does not know about the kind of traffic a signal group is responsible for, each sort of demand can be added to each group.

In some cases it is in fact necessary to have different kinds of demand on one signal group: For Individual-Traffic signal groups an additional traffic demand for Public Transport has to be created if the bus does not have a signal group of its own but shall be prioritized nevertheless.

Signal groups can also be ‘cloned’: More than one traffic demand is assigned to the same signal group. This is very helpful if the signal group controls different driving directions with different lanes.

The interface for entering the parameters is customized for the different types of traffic.

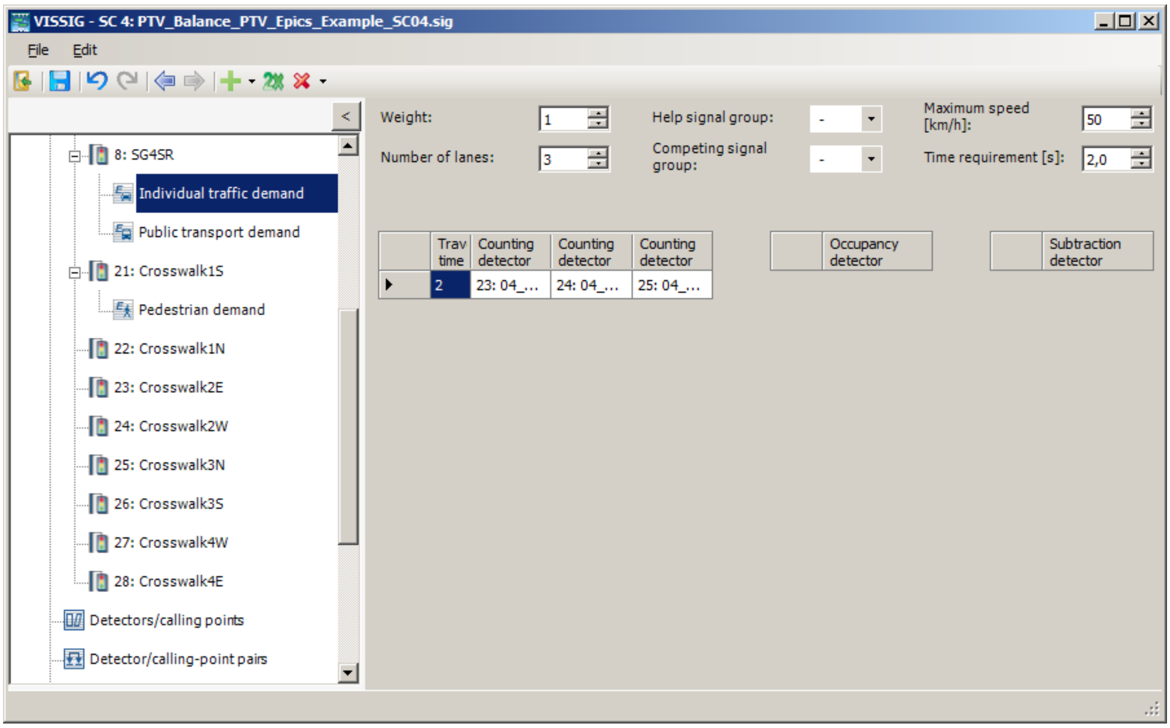
1. Click on the signal group.
2. Right-click on the signal group.
3. Select **Add individual traffic demand**, **Add public transport demand** or **Add pedestrian demand**.
   The corresponding editor view opens.

Removing a traffic demand

1. Click on the traffic demand.
2. Right-click on the traffic demand.
3. From the context menu, choose the entry **Delete traffic demand.**

Setting a PrT demand

1. In **Signal groups**, click the desired **Individual traffic demand**.



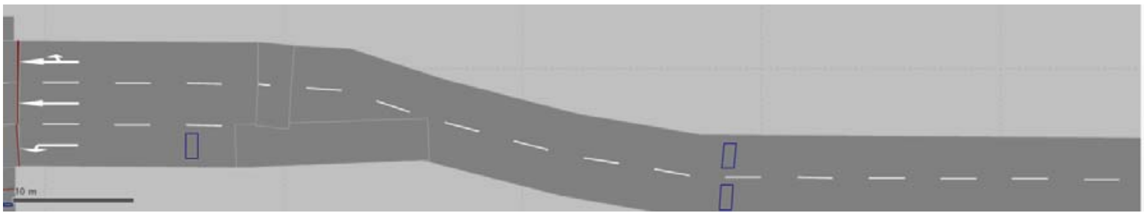
2. Enter the desired data:

Element	Description
Weight	The weighting of the traffic stream towards the signal group; in general it is recommended to assign weight 1 to an Individual Traffic demand.
Number of lanes	The number of lanes the vehicles can use for flowing off. This quantity enters the capacity linearly.
Help signal group	An example is a green right-turn arrow for straight traffic. The demand has green if either the signal group or the assigned auxiliary signal group has green
Competing signal group	An example is oncoming straight traffic vs. left-turning traffic, which means a conditionally compatible flow. If the particular signal group is green at the same time, the outflow capacity of the demand is reduced to only 50% of its actual value
Maximum speed [km/h]	The allowed maximum speed. It is required to calculate the available queue space between detector and stop line.
Time requirement [s]:	Time typically required by a vehicle for flowing off. Can be adapted by Epics during run time. Standard values: - PrT-Vehicles: 2 s - Trams: 4 s

3. Specify detectors.

Element	Description
Travel time	Only for counting detectors. Travel time from detector to stop line [s]: typical travel time of a PrT vehicle from detector to stop line.
Counting detector	Parallel detectors have to be selected from the list of supplied detectors. Each row in the table corresponds to consecutive detectors on the road.

Occupancy detector	If one or more such detectors are supplied, the signal group is only switched on if at least one of these detectors is occupied.
Subtraction detector	Subtraction detectors are used in case the lane is widened by a turn-off lane and traffic is subtracted from the main flow. The detector on the turn-off lane is the subtraction detector, see figure.



Adding a counting detector

1. In the table **Counting detector,** right click a table cell.
2. From the context menu, choose either entry **New-detection point (row)** or **New-counting detector (column)**.
   An additional column represents parallel detectors at the same distance to the stop line of the signal group.
   Different rows represent detectors at different distances to the stop line of the signal group.
3. In the table **Counting detector,** click a cell and select the detector from the table of detectors and enter the desired data.

Removing a counting detector

1. In the table, right-click the desired detector, row or column.
2. From the context menu, choose the entry **Delete** or **Delete column**.
   There always has to be at least one row or column. Therefore, the last row or column cannot be deleted.

Adding an occupancy or subtraction detector

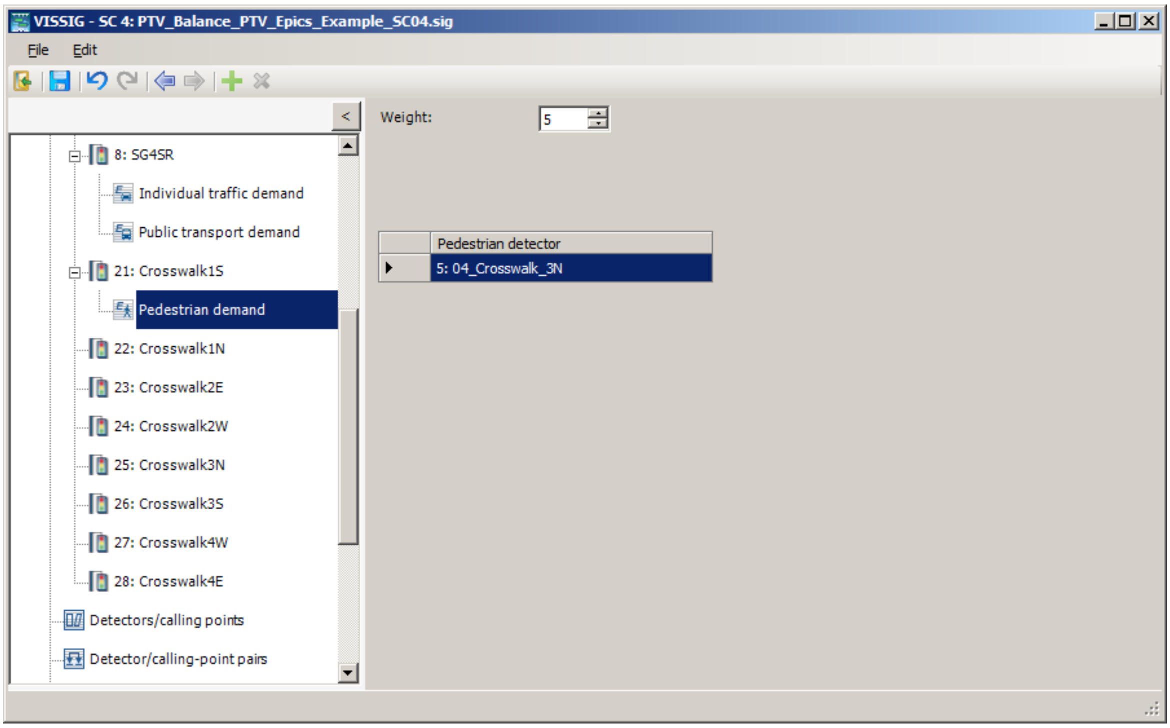
1. In the table, right-click the desired table cell **Occupancy detector** or **Subtraction detector**.
2. From the context menu, choose **New**.
   An additional row represents another detector.
3. In the table, click the table cell **Occupancy detector** or **Subtraction detector**, select the detector from the table of detectors and enter the desired data.

Removing an occupancy or subtraction detector

1. In the table, right-click the desired detector or row.
2. From the context menu, choose the entry **Delete**.

Setting a pedestrian demand

1. In **Signal groups**, click on the desired Pedestrian demand.



2. Enter the desired data:

Element	Description
Weight	Weighting of the pedestrian demand. It is recommended to set a weight larger than 1 because the number of pedestrians who want to cross the street is unknown. The default value is 5.

3. Specify detectors.

Element	Description
Pedestrian detector	Detector representing the push-button of the pedestrian crossing.

Adding a pedestrian detector

1. In the table, right click the table cell **Pedestrian detector**.
2. From the context menu, choose the entry **New**.
   An additional row represents another push-button with its own channel number.
3. In the table, click the table cell **Pedestrian detector,** select the detector from the table of detectors and enter the desired data.

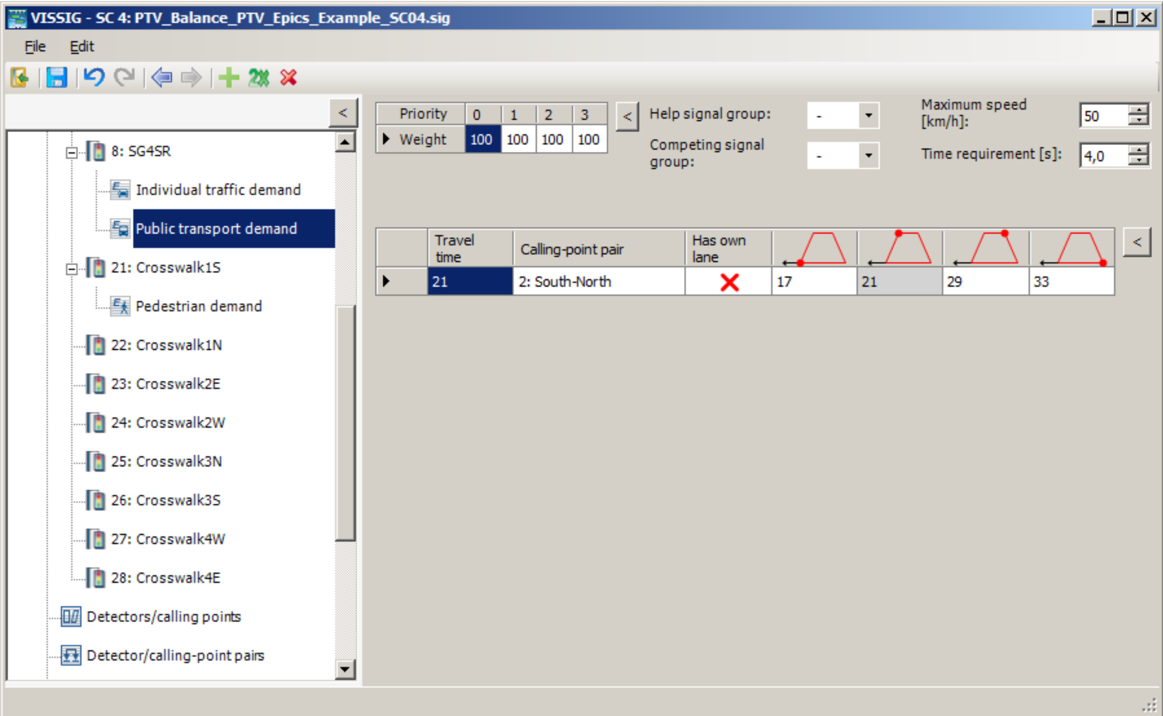
Removing a pedestrian detector

1. In the table, right-click the desired pedestrian detector.
2. From the context menu, choose the entry **Delete**.
   There always has to be at least one detector. Therefore, the last detector cannot be deleted.

Setting a PuT Demand

The supply of a PuT demand is largely identical to the supply of a PrT demand. The most important difference are the detector/calling-point pairs which have to be selected instead of detectors.

1. In **Signal groups**, click on the desired **Public transport demand**.



2. Enter the desired data:

Element	Description
Weight	see PrT demand A value of 50-100 is recommended in case of a PuT prioritization. Use the < or > button to hide or show weights for several priorities. You can supply four different weights for four priorities in order to dynamically vary the prioritization of a vehicle. When the associated Calling-point pair of the public transport demand consists of PTV Vissim standard detectors, PTV Epics will use the weight of the priority 1. Detectors of the type PT calling point are able to submit PT Telegrams with different priorities. The determination of a dynamic priority based on e.g. delay etc. is done outside of PTV Epics.
Help signal group	see PrT demand
Competing signal group	see PrT demand
Maximum speed	see PrT demand
Time Requirement [s]	see PrT demand

3. Specify calling-point pairs.

Element	Description
Travel time	Travel time between first detector or calling point of the given pair and stop line [s]: typical time which is required for a PuT vehicle to get from the calling point to the stop line.

	PTV Epics uses a trapezoid approach profile to represent fastest, typical, typical slow and slowest travel times to the stop line. PTV Epics calculates a predefined trapeze based on the travel time. In order to edit these, use the < or > button to hide or show the approach profile definition for the travel time.
Calling-point pair	Select from the table of calling-point pairs.
Has own lane	If this option is selected, the vehicle has its own lane.

Adding a calling-point pair

1. In the table, right-click the table cell **calling-point pair**.
2. From the context menu, choose the entry **New**.
   Different rows represents calling-point pairs with different travel times to the stop line.
3. Click the table cell **calling-point pair**, select the calling-point pair from the table of calling-point pairs and enter the desired data.

Removing a calling-point pair

1. In the table, right-click the desired **calling-point pair**.
2. From the context menu, choose the entry **Delete**.
   There always has to be at least one calling-point pair. Therefore, the last calling-point pair cannot be deleted.

3.1.3 Detectors/calling points

Overview of detectors/calling points

Detectors and/or calling points are used to define traffic demands. When used in conjunction with PTV Vissim the Channel Number in PTV Vissim has to correspond between PTV Epics and PTV Vissim.

For PTV Epics a detector is a standard detector in PTV Vissim and a calling point is a PT calling point in PTV Vissim.

Detectors and/or calling points can be used to define detector pairs and/or calling point pairs that are used for PuT prioritization.

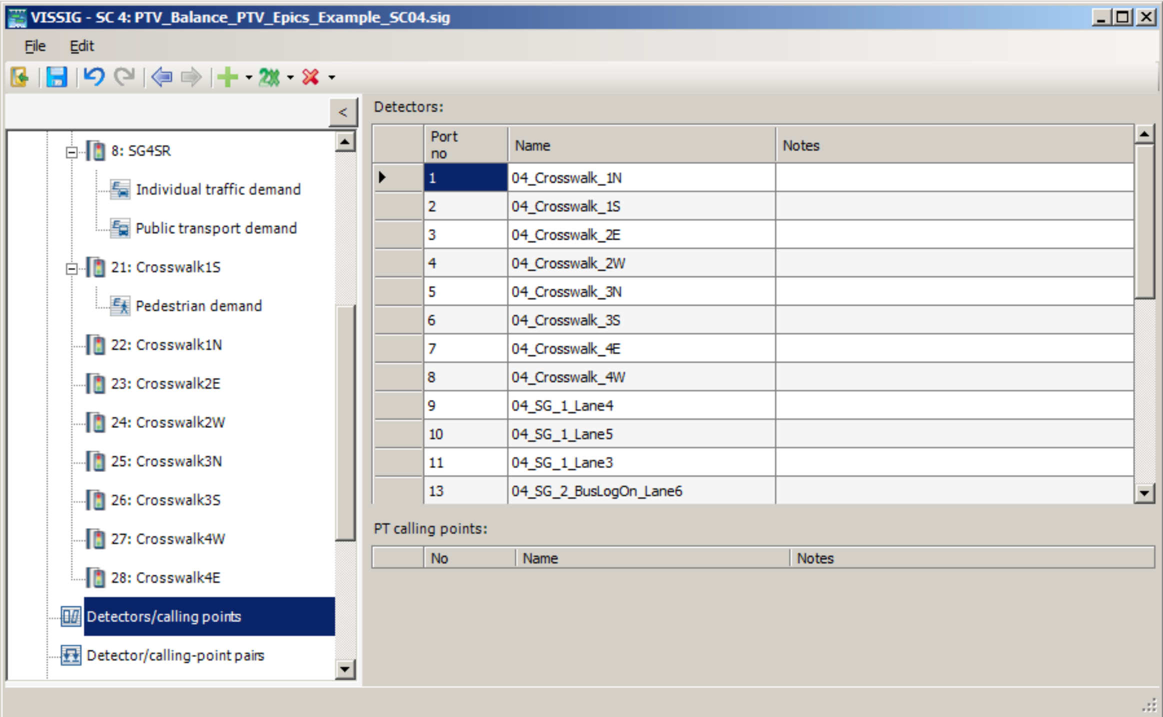
Hint: PTV Vissim and Epics/Balance-Local have a simple one-way-synchronization of detectors from PTV Vissim to Epics/Balance-Local. This means that you should provide detectors in PTV Vissim first and then upon opening the GUI of Epics/Balance-Local the detectors that are associated with the given signal controller are synced to Epics/Balance-Local. This process greatly reduces effort and the possibility to provide false channel numbers.

The sync will delete detectors in Epics/Balance-Local that do not exist in PTV Vissim. If that happens accidently, do not save the sig-file after opening the GUI of Epics/Balance-Local.

This does not apply when Epics/Balance-Local is used with PTV Visum.

Setting a detector or calling point

1. Click on **Detectors/calling points**.



2. Enter the desired data:

Element	Description
Port no/No	The corresponding channel number of the detector or calling point in PTV Vissim.
Name	Name of the detector or calling point used in selection menus of other tables.
Notes	Free text.

Adding a detector or calling point

1. In the table, right-click the table cell **Detectors** or **PT calling points**.
2. From the context menu, choose **New**.
3. Click in the table cell **Occupancy detector** or **Subtraction detector**, select the detector from the table of detectors and enter the desired data.

Removing a detector or calling point

1. In the table, right-click the desired detector or row.
2. From the context menu, choose the entry **Delete**.

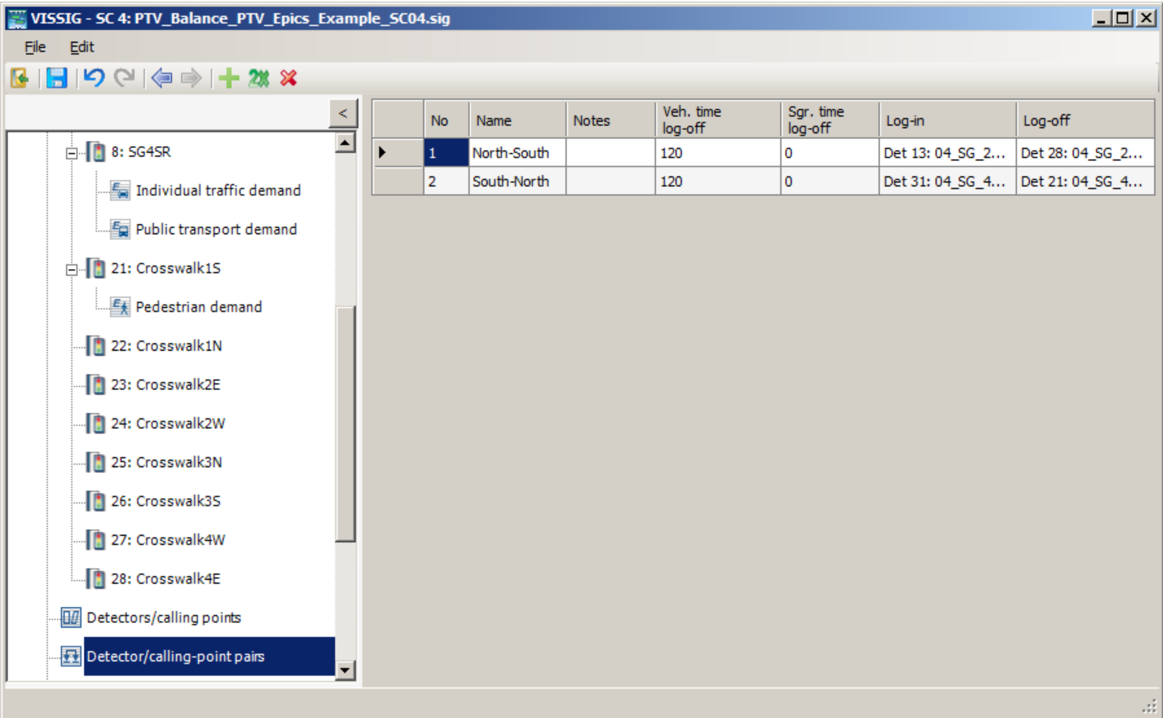
3.1.4 Detector/calling point pairs

Overview of detector/calling point pairs

Detector pairs and/or calling point pairs are used to define public transport demands. Such a pair consists of a log-in and a log-off detector or calling point as defined in the detectors/calling points table. Typically, a log-in detector is located far away from the stop line (10-40s) to allow the signal control to prepare for the arrival of the PuT vehicle. The log-off detector should be placed directly after the stop line. This detector confirms that the PuT vehicle successfully passed the intersection.

Setting a detector/calling point pair

1. Click on **Detector/calling point pairs**.



2. Enter the desired data:

Element	Description
No	Internal number to identify the pair.
Name	Name of the pair used in selection menus of other tables.
Notes	Free text.
Veh. time log-off	Time span in seconds after that a PuT vehicle is logged off automatically. In simulation, it is technically impossible that the log-off detector fails to recognize a PuT vehicle. Yet this may happen in the real world. This parameter prevents that a stage is on hold forever for a PuT vehicle. Typical values are 2-3 times the typical travel time of the associated **Public transport demand**.
Sgr. time log-off	Time span in seconds that the signal group had to be green before a PuT vehicle can be logged off automatically. See also **Veh. time log-off**.
Log-in	First detector or calling point that is able to detect the approaching PuT vehicle.
Log-off	Detector after the stop line that is able to detect the PuT vehicle.

Adding a detector/calling point pair or log-in/log-off detector/calling point

1. Right-click in the table.

2. From the context menu, choose either **New-calling point pair**, **log-in column** or **log-off column**.
   Additional log-in or log-off columns are required for parallel detectors or calling points on links with multiple lanes.
3. Click in the table cell **Log-in** or **Log-off**, select the detectors or calling points from the table of detectors or calling points and enter the desired data.

Removing a detector/calling point pair or log-in/log-off detector/calling point

1. In the table, right-click the desired row or column.
2. From the context menu, choose the entry **Delete**.

3.1.5 Intergreen matrices

As required for any stage based fixed time control modelled with Vissig.

3.1.6 Stages

As required for any stage based fixed time control modelled with Vissig.

PTV Epics has parameters that are stage and signal program dependant. These are configured in the **signal programs** settings.

3.1.7 Stage assignments

As required for any stage based fixed time control modelled with Vissig.

3.1.8 Stage sequence editing

As required for any stage based fixed time control modelled with Vissig.

3.1.9 Signal programs

As required for any stage based fixed time control modelled with Vissig.

Additionally PTV Epics requires the provision of signal program and stage dependent parameters that define priorities and boundaries for the optimization.

Overview of signal program parameters

PTV Epics has parameters concerning time and duration boundaries of stages and allowed interstages. These parameters are signal program dependent and are set to default values during the automatic creation. Some of the default values are derived from the fixed-time signal program (e.g. preferred start or end of a stage).

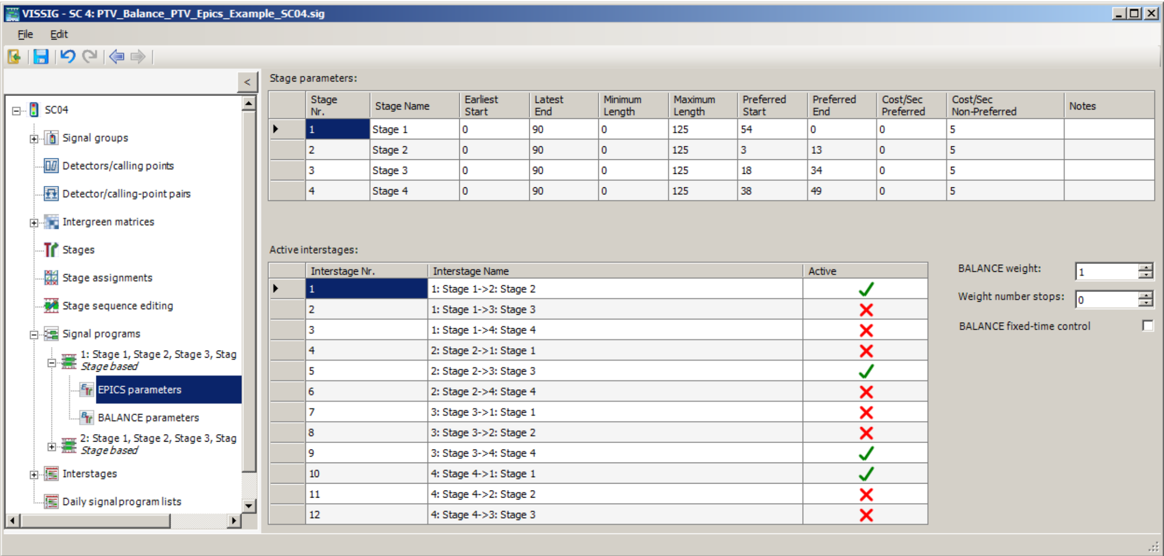
These parameters govern the optimization function of PTV Epics.

> Hint: Every parameter for PTV Epics comes with a tooltip providing an explanation of its meaning. In addition, there are numerous plausibility checks with explanations.

When you have the license for PTV Balance and PTV Epics and you want to use only PTV Balance, then activate **EPICS parameters** > **BALANCE fixed-time control** to force PTV Epics to follow the results of PTV Balance exactly.

Setting signal program parameters

1. Select a stage based signal program.
2. Expand the Navigator.
3. Select **EPICS parameters**.



4. Make the desired changes of the **Stage parameters**

Element	**Description**
Stage Nr.	Non-editable as defined in stages.
Stage Name	Non-editable as defined in stages.
Earliest Start	Cycle second [s] defining the earliest possible start of the corresponding stage.
Latest End	Cycle second [s] defining the latest possible start of the corresponding stage.
Minimum Length	The minimum length of the corresponding stage [s].
Maximum Length	The maximum length of the corresponding stage [s]. 0 means no maximum length.
Preferred Start	Preferred Start/End and Cost/Sec Preferred/Non-Preferred reflect the importance of the underlying fixed time signal plan and therefor the coordination with other intersections. The objective function of PTV Epics considers active traffic demands and potential additional costs to activate any given stage. Inside the interval defined by preferred start/end the Cost/Sec preferred is applied. This typically resembles the fixed time plan. Outside of this interval the Cost/Sec Non-Preferred is applied. Typically Cost/Sec Preferred is 0 and Cost/Sec Non-Preferred is 5-20.
Preferred End	See preferred start.
Cost/Sec Preferred	See preferred start.

Cost/Sec Non-Preferred	See preferred start.
Note	Free text.

5. Make the desired changes of the **Active interstages**

Element	**Description**
Interstage Nr.	Non-editable as defined in interstages.
Interstage Name	Non-editable as defined in interstages.
Active	If active PTV Epics is allowed to use the corresponding interstage for the optimization. Restricting the active interstages allows to e.g. reach a public transport stage from any other stage and to leave that stage only to a specific other stage. If there is only one valid path of active interstages (e.g. stage 1 -> stage 2, stage 2 -> stage 3, stage 3 -> stage 1) then PTV Epics will only optimize stage durations. If all interstages are non-active PTV Epics will not operate.

6. Make the desired changes of the other parameters

Element	**Description**
BALANCE weight	The weight of a framework signal plan given by a network signal control (e.g. PTV Balance).
Weight number stops	If 0 PTV Epics optimizes only the total delay of the vehicles. This factor is especially relevant if emissions shall be minimized. If the stops shall be considered their weight is usually set to 10-50 since one single stop contributes several seconds to the delay time.
BALANCE fixed-time control	When active and the corresponding licenses are available, PTV Epics will be forced to follow the frame signal plan from PTV Balance exactly.

Resetting stage parameters of a signal program

1. In the table **Stage parameters** right-click the desired cell or column.
2. From the context menu, choose either **Reset values of table** or **Reset values of column ...**
   This resets either the whole table or the corresponding column to the default values.

3.1.10 Interstages

As required for any stage based fixed time control modelled with Vissig.

3.1.11 Daily signal program lists

Optional but as required for any stage based fixed time control modelled with Vissig.

4 Simulation, Calibration and Operation

Testing the efficiency of an adaptive signal control like PTV Epics is nearly impossible without a modern simulation environment like PTV Vissim. It is the only way to check whether all parameters are chosen and calibrated well and if all detectors and calling points for Public Transport are correctly defined. It should also be checked whether the approximated arrival times of the Public-Transport vehicles are realistic.

On the other hand a simulation is never a 100-percent mapping of reality. Therefore the step from the simulated network to the real road network has to be observed carefully.

4.1 Simulation With PTV Vissim

4.1.1 Data Provision

There is practically no additional effort in the creation of a PTV Epics control for a PTV Vissim intersection besides the conventional build-up of a PTV Vissim simulation as PTV Epics is a defined signal control type in PTV Vissim.

For the quick and comfortable provision of large networks, the usage of PTV Visum is recommended but not required. This approach is the standard data provision for PTV Balance, hence for details we refer to the PTV Balance manual (PTV 2015c).

4.1.2 Evaluating Additional Data in the Signal Control Detector Record

You can use the SC detector record to check control logic of external control procedures. For each SC, you can show a freely configurable, precise record of the SC values and detector values as well as internal parameters of the control procedure.

For details on the standard attributes and configuring the signal control detector record please refer the Vissim user manual (PTV 2015a).

PTV Epics allows you to show additional attributes that are described below.

Result of the SC detector record evaluation, additional PTV Epics attributes

Value type	Meaning
PÜ	PTV Epics desired interstage (PÜ = Phasenübergang - German for interstage) PTV Epics displays the current desired interstage i.e. the interstage that it intends to apply: ➨ 1-99 number of the interstage ➨ * no interstage intended
PH	PTV Epics current stage (PH =Phase - German for stage) ➨ 1-98 number of the stage ➨ 99 arbitrary interstage
EP-Qu	PTV Epics queue

	EP-Qu indicates the queue state of the PTV Epics simulation model: - 0-99 vehicles in the queue

4.2 Log Files

PTV Epics provides several log files. These contain warnings, errors and results. The log files are an important tool to identify errors in the data provision and for calibration.

Log files are created in the subfolder of the directory of the inpx-file. The folder has the name Epics_Log.

Most log files provide a short explanation of its content and afterwards write one line per second. The first two columns display the PTV Epics time and the current stage. Often the third column contains the current cycle second Tx.

Files concerning the optimization algorithm do not add a line every second because PTV Epics only optimizes if the current stage is recognised. If PTV Epics does not recognise the current stage, then the signal control is currently in the state of an interstage and cannot be influenced. All other log files indicate this state with the value 99 for the current stage.

PTV Epics creates numerous log files of the following schemes and types:

- Epics_LOGTYPE_XXX.txt
  - LOGTYPE identifies the type of the log file. The most important types are described below.
  - XXX represents the number of the intersection.
- json files
  These are used internally by the visualization component.

4.2.1 Epics_MainLog_XXX.txt

The main log file. It includes error messages, warnings and information of the initialization and the optimization. All at present possible messages are summarized in chapter 6.1.

4.2.2 Epics_Supply_XXX.txt

This file is nothing more than a mirror of the PTV Epics parameters from the sig or the xml file, respectively. Nevertheless, it is often very useful, for example when the user is not sure whether he already did an intended change in the supply or not.

4.2.3 Epics_Action_XXX.txt

This file records the requested PTV Epics actions as already suggested by the name. In each line all available stages are listed. Only for the current stage an action may appear. It is either ‘h’ for ‘hold’ if PTV Epics wants to keep the current stage for another second or it reads ‘Ixy’ if PTV Epics wants to switch an interstage. E.g. I02 means ‘switch interstage 2’, I11 means ‘switch interstage 11’. It sometimes happens that the request of an interstage comes several times in a row. This usually is explained by the situation that PTV Epics still determines a stage although the interstage already started. This often happens when signal groups change to flashing green at the beginning of an interstage. PTV Epics cannot distinguish between green and flashing green. Here it shall be recommended to define interstages always as short as possible. The minimum green time of the signal groups will be added by PTV Epics and should not be defined as part of the interstage. The file 'Epics_Action_XXX.txt' can also be used to check the reproduction of a fixed-time program by PTV Epics without any detector input data. One simply checks whether the time intervals between the interstages equal the fixed-time program and whether the total cycle length is maintained.

4.2.4 Epics_Queues_XXX.txt

This file is crucial to ensure that all traffic demands are correctly provided. Every demand is represented as a column in this file. E.g. a traffic demand for Individual Transport looks like 4/8(20). The meaning of the three values is: In the PTV Epics model there are currently 4 vehicles in the vertical queue and there is space for approximately 8 vehicles between stop line and detector. 20 is the value for the required time for one vehicle passing the stop line in tenths of seconds, so the time requirement is 2 seconds in this example. This value is listed as well since PTV Epics can adapt it dynamically when it detects e.g. a disturbance in the outflow. In case of a pedestrian traffic demand the equivalent values are 5/25 (10). Here only the weight of the signal group 5 is defined by the user. For a PuT vehicle with a weight of 100 and a required time for passing the stop line of 4 seconds, the values read 16/0 (40). The value 16 arises from the fact that the vehicle has a minimum extension of 6 seconds in front of the stop line and 100/6 is about 16. The second value is 0, since the queue space is irrelevant for Public Transport. Sometimes one can observe a sudden increase or decrease of the queue length in PrT demands. The increase is the result of the implemented tailback estimator that adds the queue length upstream the detector of the last cycle to the queue length of the current cycle at the very moment the detector is permanently occupied again. A sudden decrease of the queue length is due to the renormalization algorithm of PTV Epics. In case the queue length of the PTV Epics model is larger than the storage space until the detector nearest to the stop line the length will be limited to the corresponding number of vehicles if the detector is free for a few seconds.

4.2.5 Epics_StageSequence-1_XXX.txt

In this and the next file one gets an idea of how PTV Epics forecasts the future. StageSequence-1 means the result of the first optimization step. It is represented as a series of triplets consisting of stage and cycle second and the cost up to this point. Example: (2,38,0) (2,43,640) (5,63,1280) (4,88,2560) (3,108,8960) (2,13,10060) (2,18,10240[4]) means the intersection is in stage 2 which shall be kept for another 5 seconds, afterwards stage 5 shall be switched and then as soon as possible stages 4, 3, and back to 2 again. The current cycle second is 38. The time grid of 5 seconds is used for calculation-time reasons, see section 2.6.

[4] For clarity the total performance index (10240 in the example) is also given at the beginning of the line.

4.2.6 Epics_StageSequence-2_XXX.txt

Here the signal plan of the PTV Epics time horizon (typically 100s) after the second optimization step is displayed. As an example 11520 (6,0) (5-6) (4,14) (9-12) (12,13) (11-9) (2,46) means: The total performance index is 11520, a value which itself does not give so much information. It is more relevant for comparison with the performance indices of the other optimization runs. The next values encode the predicted signal plan for the next 100 seconds: Stage 6 shall be kept another 0 seconds, i.e. PTV Epics wants to switch an interstage right now. This interstage (with the number 5) is given in the next pair. It has a duration of 6 seconds and the subsequent stage 4 takes 14 seconds. Afterwards stage 12 and finally stage 2 follow.

4.2.7 Epics_PerformanceIndex-1 _XXX.txt

This file contains two columns for the performance index (see Chapter 2.6) of the first optimization part. In general the values in both columns are equal. Only in case Epics does not find a valid way through the phase space at first go it repeats its calculation putting special emphasis on the signal groups whose maximum Red times might be hurt. In many cases it thus ends up with a different (considerably smaller) performance index. This file is usually not needed.

4.2.8 Epics_PerformanceIndex-2 _XXX.txt

It contains the total performance index for the best result of the second optimization step and the individual contributions of delays and stops. This file is usually not needed.

4.2.9 Epics_NumberStops-1_XXX.txt
Epics_NumberStops-2_XXX.txt

These two files are only of interest if the number of stops is considered in the optimization, i.e. if the FactorStops (or the weightStops, respectively) is greater than 0. They provide for optimization steps 1 and 2 the predicted number of stops per traffic demand.

4.2.10 Epics_Debug_xxx.txt
Epics_Runtime_xxx.txt
Epics_APValues_xxx.txt

These files are for debug purposes only and not described any further.

4.3 Calibration

Before starting the calibration, i.e. the optimization of the PTV Epics parameters, one should make sure that there is no fundamental mistake in the PTV Epics data provision.

First of all, it is useful to run the simulation without any vehicle input. In this case, the fixed-time control has to be reproduced, exactly like using standard Vissig. Afterwards, vehicle and pedestrian inputs are added to the simulation and the intersections controlled by PTV

Epics have to be checked step by step. To this end the file 'QueueTest_XXX.txt'[5] is particularly important. As described in Chapter 4.2, the number of vehicles assumed by PTV Epics in the queue in front of a signal group is shown in this file. This way one gets very easily an overview whether PTV Epics estimates the situation realistically or not by comparing the modelled value with the vehicles in the actual queue of the simulation. If this is not the case, there is probably a data provision error, e.g. mismatches of detectors, incorrect number of lanes, inaccurate travel times from the detector to stop line, etc. Also the value for saturation, i.e. the required time for one vehicle passing the stop line, can cause inaccuracies. In particular this sometimes happens when vehicles use the same lane for turning right and going straight (mixed lanes) and there is a green arrow for the right turn. In this case one can only estimate how many vehicles will pass the stop line during green since the value depends very much on how many vehicles want to turn right. In such situations it is sometimes useful to increase the capacity, i.e. decrease the time-requirement value, e.g. to 1,8s. If no error can be identified and PTV Epics still does not estimate the current situation correctly, please contact PTV.

Special attention should be put to Public Transport. These vehicles also appear in the 'QueueTest_XXX.txt', but are not necessarily visible there if they pass the stop line without time loss. A good criterion for a well-defined travel time between calling point and stop line is that a PuT vehicle which passes the stop line without any delay appears in this file for a few seconds and is deleted immediately afterwards. The entries will e.g. look like: 0, 7, 15, 16, 0, 0, …

If one is confident to have eliminated all major bugs one can start the actual calibration of PTV Epics. As in many other programs too, the calibration process of PTV Epics is an iterative one. Modern simulation tools like PTV Vissim offer many possibilities to evaluate traffic parameters which reflect the performance of the road network. However, one may not completely rely on these parameters. The actual simulation must always be observed as well. Sometimes the overall performance seems to be pretty good, but this results mainly from a completely jammed access on one or more intersections. Consequently less vehicles drive through the network and the total value of e.g. delays will improve. Nonetheless, such a situation is of course not acceptable.

Depending on the aim of the signal control, the evaluated parameters in PTV Vissim may differ. Due to the fact that PTV Vissim offers extensive possibilities for analysing such data, nearly all objectives of the signal control can be measured with the help of these evaluation mechanisms (for details see PTV AG 2015a).

It is recommended to configure the following PTV Epics parameters in order to calibrate the overall system:

- If needed, the first thing one should check is the coordination between the traffic lights in order to get an adequate mean between coordination and flexibility of the adaptive signal-control system PTV Epics. To this end one deactivates the Public Transport and its prioritization in the simulation, since these vehicles of course disturb the coordination. Now one varies the costs of the coordination intervals or the PTV Balance-weight, see Chapter 3.1.9. As a rule of thumb, a weak coordination corresponds to Cost2=5 (WeightBalance=1), a strong coordination to Cost2=20

[5] In future versions a direct visualisation of the most important data will be possible.

(WeightBalance=3). Once again it shall be mentioned that within the first optimization step PTV Epics only considers the Costs. In the second step, however, in case WeightBalance > 0 every second a signal group deviates from the PTV Balance plan is punished with this weight, i.e. the influence of PTV Balance does not only depend on the weight itself but also on the number of signal groups. On the other hand, in case WeightBalance = 0 also in the second step the stage related Costs are respected.

- If not only the total waiting time but also the number of stops is taken into account within the optimization routine (see Chapter 2.6.1), the additional parameter FactorStops has to be adapted as well (Chapter 3.1.9). In principle, there is an analogous factor for the waiting time, but it is recommended to leave this parameter at its default value 1. A weak impact of the number of stops is e.g. FactorStops=10[6], a large one FactorStops=50.
- In case the observed reduction of waiting time for Public Transport is less than desired, one increases the weights for these demands. At the same time, the simulation has to be monitored carefully in order to ensure that the delays do not originate from busses that are jammed together with Individual Transport. Therefore it is not advisable to neglect the Individual Transport in favour of Public Transport, at least in case the PuT vehicles do not have an own lane. If the level of service for pedestrians is not satisfying, of course one has the possibility to raise their weights e.g. to 10 instead of 5.
- As a very last possibility, one can adapt the weight factors for Individual Transport. This may be useful at a prone access to an intersection, which is known to be threatened by traffic jams. However, one should abstain from this possibility as far as possible since it hurts the principle of PTV Epics to handle all individual vehicles equally.

In order to make calibration as traceable as possible, it is useful to do minor changes. The simulation runs have to be carried out with these new changes and the achieved effects have to be checked. By the way: It is not enough to consider only one simulation run. To achieve sufficient statistical reliability of the simulated results, several simulation runs (10-20) with different random seeds have to be carried out for each scenario.

The described iterative process should be carried out until the desired objectives are fulfilled in the best possible way. The traffic engineer will develop quite fast a feeling for the individual factors since PTV Epics usually reacts in a similar way to a modification of certain parameters. Thus the objectives are quickly achieved and the calibration can be completed successfully.

Model based optimization algorithms are often ill-reputed as ‘Black-Box methods’ since their logic is elusive. The logic of rule-based methods is created by engineers, so here this prejudice does not exist. ‘Black-Box methods’ are said to be often hard to calibrate for the traffic engineer. PTV Epics opposes this prejudice. Although the advantages of PTV Epics are also founded in an optimization function instead of a direct rule-based logic the reaction of the optimization function is easy to understand for the user because of the configured weight factors etc. Of course, PTV Epics is deterministic, i.e. it behaves identically in two

[6] Reminder: every stop causes more than 10 seconds waiting time in average.

simulation runs with identical random seed and parameters. This is extremely important for the comprehension and the cognition of human beings. All decisions of PTV Epics remain comprehensible and the influences of the political or transport motivated objectives are thus directly realizable and understandable.

4.4 Operation

Concerning the operation and the effort for maintenance, a PTV Epics control behaves more or less the same as each conventional rule-based control system.

It is recommended to periodically check the input data. Due to the self-calibrating mechanisms of PTV Epics this is only necessary in larger intervals than with a rule-based control. The effort concerning the adaption of the data provision files to changing traffic demands is therefore reduced.

Regarding support and maintenance of detectors, the same rules and intervals as in other traffic dependent control systems can be applied. There is no additional effort in the operation with PTV Epics here.

For model-based control systems – as well as for all conventional rule-based controls - modern controllers are required. Since the optimization algorithm has to run every second, enough computing power has to be provided. While operating, the controller does not have to be exchanged. For PTV Epics controls that have been implemented once, the required resource demands do not increase with time. This behaviour is also quite similar to conventional control methods.

Finally it may be stated that the operation of the adaptive intersection control PTV Epics generates neither a higher effort in manpower nor in material and financial resources with respect to a conventional rule-based control.

5 Literature

Braun, R.; Kemper, C. (2008). GALOP-Online – ein Genetischer Algorithmus zur netzweiten Online-Optimierung der Lichtsignalsteuerung. Forschungsgesellschaft für Straßen- und Verkehrswesen (Hrsg.), HEUREKA '08, Optimierung in Verkehr und Transport - Tagungsband, FGSV Verlag, Köln, ISBN 978-3-939715-48-1.

Liu, H., Wu, X., Ma, W., Hu, H. (2009). Real-time queue length estimation for congested signalized intersections. Transportation Research Board C 17 (4), 412-427.

Mertz, J. (2001). Ein mikroskopisches Verfahren zur verkehrsadaptiven Knotenpunktsteuerung mit Vorrang des öffentlichen Verkehrs (Dissertation). Veröffentlichung des Fachgebiets Verkehrstechnik und Verkehrsplanung, TU München.

PTV AG (2015a). PTV Vissim manual. Karlsruhe.

PTV AG (2015b). PTV Visum manual. Karlsruhe.

PTV AG (2015c). PTV Balance manual. Karlsruhe.

RiLSA (2010). Richtlinien für Lichtsignalanlagen Köln, ISBN 978-3-939715-48-1.FGSV – Forschungsgesellschaft für Straßen- und Verkehrswesen: RiLSA –. Köln.

6 Appendix

6.1 PTV Epics Messages

In case of an error, PTV Epics writes a detailed message with a unique id to the file 'Epics_Log_XXX.txt' (see Chapter 4.2). Chapter 6.1.1 lists all possible error messages and explanations. Furthermore, chapter 6.1.2 lists warnings and (harmless) information.

As a reminder, it shall be noted once again that a PTV Epics error leads to the return value -1, which means PTV Epics is not able to optimize. However, this state might already change the very next second, e.g. if the error occurred due to a detector failure. In case of a warning PTV Epics keeps on optimizing but the origin of the message should be checked nevertheless since the efficiency of the program might be affected.

6.1.1 Errors

Id	Message	Explanation
10000	Unknown calling mode (neither 0 nor 1 nor 2).	Irrelevant for the user, can only appear in case of a programming error.
10001	Current signal program not supplied.	PTV Epics is getting called although a program without a PTV Epics data provision is running. Usually harmless since PTV Epics is also called in case of a program change to a non-PTV Epics program. It will display this message once and no longer be called in the following.
10002	TX outside of [1, cycle length].	The queried cycle second does not comply with the requirements. To be resolved outside of PTV Epics.
10003	Cycle length transferred differs from cycle length supplied.	The queried cycle length is unequal the provided one. Only possible in combination with VS-PLUS. Check both supplies.
10004	Number vehicles within calling-point pair > maximum number vehicles.	The number of PuT vehicles within a calling-point pair in PTV Epics is restricted to 6. Should be sufficient. In case the error occurs, presumably either the log-off does not work or the calling point is not properly assigned. Thinkable is e.g. that a PrT detector is connected to the channel number of the calling point.
10005	Serious failure on detector; optimization not possible.	A detector whose data cannot be replaced by some other detector reports a failure. See also the warnings 10228 – 10230.
10006	Less than two interstages active in at least one signal program.	For a running signal program obviously at least two active interstages are needed.
10007	Xml scheme of netCommand not correct.	Parsing the PTV Balance net command did not succeed since its xml scheme is not as expected. Please contact PTV.
10008	Return value of malloc is NULL.	PTV Epics was not able to allocate memory. Might be a temporary problem if it occurs after the initialization of PTV Epics. Serious if it occurs during initialization.

Id	Message	Explanation
10009	Epics could not be initialized.	Obviously an error occurred during initialization. There should be a less general message before.
10010	Signal group not known within TRENDS.	A signal group with the transferred index is not known to TRENDS. Please check carefully the data provision of the signal groups.
10011	Time since signal plan has changed > cycle length.	The frame given by PTV Epics has been constant for more than one cycle. Can only appear in combination with VS-PLUS.
10018	Value begin > Maximum value begin or < Minimum value begin.	The starting time of an interstage of the fixed-time program is outside the allowed range.
10019	No unused ID for calling point found.	There is only space for 255 detectors / calling points. In case there are more, the auto generation of the calling-point ID fails.
10020	Non-declared detector used.	A detector that is not included in the detectors list of the sig file is used in a traffic demand.
10023	ID of signal program > maximum ID (50) or < minimum ID (1).	ID of a signal program outside the allowed range. Correct if possible. In case it is not, please contact PTV. The same thing holds for messages 10024 - 10027.
10024	ID of stage > maximum ID (100) or < minimum ID (1).	ID of a stage outside the allowed range.
10025	ID of interstage > maximum ID (100) or < minimum ID (1).	ID of an interstage outside the allowed range.
10026	ID of signal group > maximum ID (255) or < minimum ID (1).	ID of a signal group outside the allowed range.
10027	ID of calling-point pair > maximum ID (98) or < minimum ID (1).	ID of a calling-point pair outside the allowed range.
10028	Number interstages in fixed-time or Balance cycle > maximum number (90).	Number interstages in fixed-time or Balance cycle outside the allowed range. Please reduce if possible. In case it is not, please contact PTV. The same thing holds for messages 10029 - 10039.
10030	Number signal displays or number default durations > maximum number signal displays (100).	Number signal displays or number default durations outside the allowed range.
10031	Number signal programs > maximum number signal programs (33) or < minimum number signal programs (1).	Number signal programs outside the allowed range.
10032	Number (real) signal groups > maximum number (real) signal groups (96) or < minimum number signal groups (1).	Number signal groups (or ‘real’ signal groups, i.e. without clones) outside the allowed range.
10034	Number calling-point pairs > maximum number calling-point pairs (30) or < minimum number calling-point pairs (0).	Number calling-point pairs outside the allowed range.
10035	Number stages > maximum number stages (30) or < minimum number stages (1).	Number stages outside the allowed range.
10036	Number interstages > maximum number interstages (90) or < minimum number interstages (1).	Number interstages outside the allowed range.

Id	Message	Explanation
10037	Number demands > maximum number demands (96) or < minimum number demands (0).	Number traffic demands outside the allowed range.
10038	Cycle length > maximum cycle length (2 * time horizon) or < minimum cycle length (1).	Cycle length of a signal program outside the allowed range.
10039	Number active interstages > maximum number (90).	Number active interstages outside the allowed range.
10040	Active interstage does not exist.	An interstage that is declared as active for at least one program has not been found in the list of the interstages. Check.
10041	Start stage for Balance not found.	In case PTV Epics is supposed to run together with PTV Balance, under the point General of the PTV Epics-xml the StartStageIDBalance has to be provided after the entry WeightBalance (happens automatically during the export from CROSSIG). It contains the ID of the stage with the LFIX. The message appears if this stage cannot be found in the list of the stages. Check stages.
10042	Balance interstage not active. Set to active.	An interstage used by PTV Balance is marked as inactive. Set to 'active' automatically. Check program-related parameters.
10043	Father signal group not found.	The father signal group of a clone has not been found in the list of the signal groups. Check signal groups.
10044	Clones must have same TimeAmber and TimeRedAmber and TimeFlashingGreen as original signal groups.	A clone has been provided with a different length of amber (or red-amber or flashing green) than its father signal group, which obviously does not make sense. Change.
10045	Field 'notStageDefining' must be equal for clones and original signal groups.	A clone has a different entry for notStageDefining than its father signal group, which obviously does not make sense. Change.
10046	Clones must be appended after all other signal groups.	The clones may only appear after the real signal groups in the list SignalGroups. Change.
10047	No valid Balance or fixed-time cycle found (interstage not found)	No closed PTV Balance or fixed-time stage sequence found; an interstage could not be identified. Check PTV Balance data provision.
10048	No closed Balance or fixed-time cycle found.	No closed PTV Balance or fixed-time stage sequence found. Check PTV Balance data provision.
10049	Attribute or value must be unsigned character (0,..,255).	Value outside the allowed range.
10050	Attribute or value must be (unsigned character)*1000 (0,..,255000).	Value outside the allowed range.
10051	No valid way through phase space found for some combination of (p t).	For the combination (stage, cycle second) no way could be found through the phase space. This message can no longer appear.
10052	Attribute 'id of detector not valid.	Attribute *id* not found for a detector.
10053	Attribute 'balIstId' or 'display' or 'begin' or value not valid.	Attribute *balIstId* or *display* or *begin* not found for an interstage of a signal program or value not valid.
10054	Value of Balance interstage not valid.	Invalid value for an interstage of the PTV Balance cycle found for one program.

Id	Message	Explanation
10055	Attribute ‘stageId’ of stage parameter not valid or stage with supplied ID not found.	Attribute *stageId* not found for a PTV Epics stage parameter or value not valid or an appropriate stage does not exist.
10056	Number stage parameters found <> number stages.	To each stage there must be a set of PTV Epics stage parameters i.e. the number of these sets must be equal to the number of stages.
10057	Number signal group parameters found <> number signal groups.	To each signal group there must be a set of PTV Epics signal-group parameters i.e. the number of these sets must be equal to the number of signal groups.
10058	Default duration does not have a valid display ID.	For the display id of a signal group an appropriate signal display could not be found.
10059	Cmd does not have a valid display ID.	For the display id of the signal group *cmd* of an interstage an appropriate signal display could not be found.
10060	Fixed state does not have a valid display ID.	For the display id of the signal group *fixedstate* of an interstage an appropriate signal display could not be found.
10061	Wrong code found for at least one stage and signal group.	The SignalStates values of a stage must be Boolean i.e. either 0 or 1.
10062	Earliest end > cycle length or < 0.	Earliest end outside the allowed range.
10063	Latest end > cycle length or < 0.	Latest end outside the allowed range.
10064	Earliest start > cycle length or < 0.	Earliest start outside the allowed range.
10065	Latest end must be at least earliest end + TSTEP-1. Hurt by at least one stage.	Latest end must have a minimum distance to earliest end of a stage. The reason is the time grid of TSTEP (at the moment 5) seconds in the first optimization step.
10066	Error in FillSignalStateArray: Either element 'activations' not found or attribute 'sg_id' not valid or signal group with supplied ID not found or Nr activations found <> Nr real signal groups.	An error while reading the *activations* of a stage. For the various error possibilities see the message.
10067	Preferred start > cycle length or < 0.	Preferred start outside the allowed range.
10068	Preferred end > cycle length or < 0.	Preferred end outside the allowed range.
10069	Element 'sgs' not found.	Element *sgs* not found for a signal program.
10070	Attribute 'actIstId' or value of active interstage not valid.	Attribute *actIstId* not found for an *activeInterstage* of a program or value not valid or an appropriate interstage does not exist.
10071	Wrong switching-on code found for at least one interstage and signal group. Code must be <= interstage length or 97, 98, 99.	The coding of the switch-on vectors of the interstages accepts only certain values (see message). Violated here.
10072	Wrong switching-off code found for at least one interstage and signal group. Code must be <= interstage length or 97, 98, 99.	The coding of the switch-off vectors of the interstages accepts only certain values (see message). Violated here.
10073	T-time ID > maximum T-time ID (100) or < minimum T-time ID (1).	TTimeID outside the allowed range, probably 0. Check table ‘ISt-definition‘ and correct if necessary.

Id	Message	Explanation
10074	Number TClosedMax > maximum number TClosedMax (255) or < minimum number TClosedMax (1).	TClosedMax outside the allowed range.
10075	Element ‘cmds’ not found.	Element *cmds* not found for a signal group of an interstage.
10076	Element ‘fixedstates’ not found.	Element *fixedstates* not found for a signal group of an interstage.
10077	Element ‘defaultDurations’ not found.	Element *defaultDurations* not found for a signal group.
10078	Attribute ‘display’ of default duration or cmd or fixed state not valid	Attribute *display* not found or value not valid; either for *defaultDuration* of a signal group or for *cmd* or *fixedstate* of an interstage.
10079	Attribute ‘duration’ of default duration or of fixed state or attribute ‘begin’ of cmd not valid	Attribute *duration* not found or value not valid; either for *defaultDuration* of a signal group or for *fixedstate* of an interstage. Or: Attribute *begin* not found or value not valid for *cmd* of an interstage.
10080	Number sub queues > maximum number sub queues (4) or < minimum number sub queues (1).	Number sub queues outside the allowed range.
10081	AdaptTR must be either 0 or 1.	AdaptTR must be Boolean i.e. either 0 or 1.
10082	Number detector IDs > maximum number (5) or < minimum number (1).	Number detectors outside the allowed range.
10084	Time distance of detector (slowest travel time) > maximum (2 * time horizon) or < minimum (1).	Time distance of a detector or slowest travel time of a PuT demand outside the allowed range.
10085	Sub queues must be ordered by TimeDistance in descending order. Hurt by at least one demand.	The order of the sub queues is chronologically predetermined. Violated here.
10088	Time horizon > maximum (255) or < minimum (1).	Time horizon outside the allowed range.
10089	Number occupancy (stealing) detector IDs > maximum number (5) or < minimum number (0).	Number occupancy or stealing detectors outside the allowed range.
10090	Help signal group not found.	Help signal group not found in the list of the signal groups.
10091	Competing signal group not found.	Competing signal group not found in the list of the signal groups.
10092	Element 'countingDetector' not found.	No counting detector found for a PrT demand.
10093	Attribute 'id' or 'name' of signal display not valid.	Attribute *id* or *name* not found or value not valid for a signal display.
10094	Neither interstage nor stay in current stage allowed. Probably supply error.	Severe. Possible reason: Very short maximum stage times or long minimum green times. Precise check of data provision necessary.
10095	Negative performance index. Probably PTV Epics error.	Severe. PTV Epics calculated a negative performance index. Must not happen. Please contact PTV.
10096	For 10 times in a row: No way through phase space found without hurting TClosedMax.	Severe. The maximum red time of a signal group TClosedMax has been violated in at least ten successive seconds. May not happen if PTV Epics works properly. Indicates a data provision error or misbehaviour of PTV Epics. The program will switch to its fall-back mechanism.

Id	Message	Explanation
10100	Name of xml file too long. Maximum number signs: 500.	Please choose shorter path.
10101	Name of supply log-file too long. Maximum number signs: 500.	Please choose shorter path.
10102	Xml supply file not found or xml scheme of file not correct.	Without TRENDS: Please check path and name of the xml file. Using TRENDS this is no possible error source since the PTV Epics data provision is contained in the vxb file. Open PTV Epics-xml with suitable program (e.g. XML Spy) and correct possible scheme errors.
10103	Root Element not found.	Messages 10103 – 10198 refer to errors that are easy to find and usually simple to resolve in the sig or the PTV Epics-xml file, respectively. Examples are missing or invalid attributes or non-optional entries or a different number of provided elements than declared before. Maybe in the future one will do without this in principle unnecessary declaration. For all these actually equal messages no explanation is given but it is referred to this general comment.
10104	Element 'General' not found.	See 10103.
10105	Element 'ControllerID' not found or value not valid or attribute 'id' of signal controller not valid.	See 10103.
10106	Element 'signaldisplays' not found.	See 10103.
10107	Element 'SystemVersion' not found or value not valid.	See 10103.
10109	Nr elements found <> Nr elements declared.	See 10103. Elements can be signal programs, signal groups, calling-point pairs, stages, interstages, detectors, sub queues or demands.
10110	Element 'SignalPrograms' or 'stageProgs' not found.	See 10103.
10111	Element 'Number' not found or value not valid.	See 10103.
10112	Element 'ProgramID' not found or value not valid or attribute 'id' of signal program not valid.	See 10103.
10113	Element 'CycleTime' not found or value not valid or attribute 'cycletime' of signal program not valid.	See 10103.
10114	Element 'ActiveStageTransitions' not found.	See 10103.
10115	Element 'SignalGroups' or ‘sgs’ not found.	See 10103.
10116	Element 'interstages' or ‘BALANCEInterstages’ not found.	See 10103.
10117	Element 'NrRealSignalGroups' not found or value not valid.	See 10103.
10118	Element 'SignalGroupID' not found or value not valid or attribute 'id' of signal group not valid.	See 10103.
10119	Attribute 'DetId' or value of 'DetectorID' not valid.	See 10103. This message can refer to counting detectors as well as to occupancy or subtraction detectors.
10120	Element 'OnDemand' not found or attribute 'underEPICSControl' of signal group not valid.	See 10103.

Id	Message	Explanation
10121	Element 'OnDemand' must be either 0 or 1.	OnDemand must be Boolean i.e. either 0 or 1.
10122	Element 'TFreeMin' not found or value not valid.	See 10103.
10127	Element 'SignalGroupName' not found.	See 10103.
10128	Value of optional parameter not valid.	Very general message. Examples for optional parameters are balanceFixedTimeControl or weightStops. The provided value does not correspond to the demanded format.
10129	Element 'RegistrationPointPairID' not found or value not valid or attribute 'id' of calling-point pair not valid.	See 10103.
10130	Element 'FirstRegistrationPointID' or 'loginPoints' or 'loginPoint' not found or value not valid or attribute 'loginId' of log-in point not valid.	See 10103.
10131	Element 'SecondRegistrationPointID' or 'logoffPoints' or 'logoffPoint' not found or value not valid or attribute 'logoffId' of log-off point not valid.	See 10103.
10133	Element 'Stages' or 'stages' not found.	See 10103.
10134	No signal group under Epics control.	Of course at least one signal group must be controlled by PTV Epics. Exception: PTV Epics only implements the frame signal plan of PTV Balance.
10135	Element 'StageID' not found or value not valid or attribute 'id' of stage not valid.	See 10103.
10136	Element 'PrgDepStageParams' or 'EPICSStageParameters' not found.	See 10103.
10137	Attribute 'ProgramID' not found or signal program with supplied ID not found.	The attribute ProgramID is missing, or an appropriate program does not exist.
10138	Element 'PrimalStart' not found or value not valid or attribute 'earliestStart' of stage parameter not valid.	See 10103.
10139	Element 'PrimalStop' not found or value not valid.	See 10103.
10140	Element 'LatestStop' not found or value not valid or attribute 'latestEnd' of stage parameter not valid.	See 10103.
10143	Element 'TClosedMax' ('TClosedMin') not found or value not valid or attribute 'epicsMaxRed' ('epicsMinRed') of signal group parameter not valid.	See 10103.
10144	Element 'Cyclical' not found or value not valid or attribute 'cyclical' of signal group parameter not valid.	See 10103.
10145	Nr program dependent signal group parameters found <> Nr signal programs.	To each PTV Epics program there must be a set of PrgDepSgrParams i.e. the number of these sets must be equal to the number of programs.
10146	Nr program dependent stage parameters found <> Nr signal programs.	To each PTV Epics program there must be a set of PrgDepStageParams i.e. the number of these sets must be equal to the number of programs.
10147	Error in FillSignalStateArray 'SignalStates': Either element not found or entry neither 0 nor 1 or attribute 'SignalGroupID' not found or signal group	An error while reading the SignalStates of a stage. For the various error possibilities see the message.

Id	Message	Explanation
	with supplied ID not found or Nr signal states found <> Nr real signal groups.	
10149	Element 'StageTransitions' or 'interstageProgs' not found.	See 10103.
10150	Attribute 'balanceFixedTimeControl' of signal program not valid.	See 10103.
10151	Element 'StageTransitionID' not found or value not valid or attribute 'id' of interstage not valid.	See 10103.
10152	Element 'StartStage' not found or value not valid or attribute 'fromStage' of interstage not valid.	See 10103.
10153	Element 'EndStage' not found or value not valid or attribute 'toStage' of interstage not valid.	See 10103.
10154	Element 'Length' not found or value not valid or attribute 'cycletime' of interstage not valid.	See 10103.
10155	Error in FillSignalStateArray 'SignalGroupsOn': Either element not found or value not valid or attribute 'SignalGroupID' not found or signal group with supplied ID not found or Nr switchings found <> Nr real signal groups.	An error while reading the switch-on vector SignalGroupsOn of an interstage. For the various error possibilities see the message.
10156	Error in FillSignalStateArray 'SignalGroupsOff': Either element not found or value not valid or attribute 'SignalGroupID' not found or signal group with supplied ID not found or Nr switchings found <> Nr real signal groups.	An error while reading the switch-off vector SignalGroupsOff of an interstage. For the various error possibilities see the message.
10158	Element 'Demands' not found.	See 10103.
10159	Supplied times for at least one PuT signal group do not define a valid trapeze.	Examples for invalid trapeze times are: All times are equal or the times are not in ascending order.
10160	Attribute 'SignalGroupID' not found or attribute 'sg_id' of signal group not valid or signal group with supplied ID not found.	The attribute SignalGroupID is missing or an appropriate signal group does not exist.
10161	Element 'Type' not found.	See 10103.
10162	Type must be either 'Private' or 'Public' or 'Ped'.	The Type of a demand must be exactly one of these key words.
10164	Attribute 'index' not found or index not valid.	See 10103.
10165	Attribute 'Priority' not found or priority value not valid.	See 10103.
10166	Element 'NrLanes' not found or value not valid or attribute 'numberLanes' of traffic demand not valid.	See 10103.
10167	Element 'MaximumSpeed' not found or value not valid or attribute 'maximumSpeed' of traffic demand not valid.	See 10103.
10168	Nr weights found <> 4.	There must be exactly four weights (corresponding to the four priority levels).
10169	Weight found < 0 or > maximum weight (65535).	Provided Weight outside the allowed range.
10170	Current stage unequal stage demanded by Epics.	The current stage is unlike the one desired by PTV Epics. Can only appear in combination with VS-PLUS.

Id	Message	Explanation
10171	Weight green wave found < 0 or > maximum weight (65535).	Provided WeightGreenWave outside the allowed range. This parameter has not been explained yet. Its usage is not advisable. Maybe it will be removed in the future.
10172	Element 'TimeRequirement' not found or value not valid or attribute 'timeRequirement' of traffic demand not valid.	See 10103.
10173	Balance order not assigned.	Pointer to PTV Balance order is NULL. Can only appear in combination with VS-PLUS.
10174	Element 'SubQueues' or 'ITDetectionPoints' or 'PTDetectionPoints' or 'pedestrianDetectors' not found.	See 10103.
10175	One of the attributes 'fastestTravelTime', 'slowTravelTime' or 'slowestTravelTime' of detection point not valid.	See 10103.
10176	Delta T > 1 and last relative time > 1.	Stage finished too early. Can only appear in combination with VS-PLUS.
10177	Train returns waiting time 0.	Implausible arrival time of a PuT vehicle received. Can only appear in combination with VS-PLUS.
10178	More Epics PuT demands than VS-PLUS traffic streams found for at least one signal group.	Data provision of PuT demands in PTV Epics and VS-PLUS obviously not identical.
10179	Element 'Detectors' or 'countingDetectors' not found.	See 10103.
10182	Element 'TimeDistanceDet' not found or value not valid or attribute 'travelTime' of detection point not valid.	See 10103.
10183	Attribute 'RegistrationPointPairID' or ‘callingPointPair’ of detection point not found or value not valid or calling-point pair with supplied ID not found.	The attribute RegistrationPointPairID or callingPointPair’ is missing or a corresponding calling-point pair does not exist.
10184	Element 'HasOwnLane' not found or attribute 'hasOwnLane' of detection point not valid.	See 10103.
10185	HasOwnLane must be either 0 or 1.	HasOwnLane must be Boolean i.e. either 0 or 1.
10187	Both 'TimeDistanceReg' and 'Trapeze' supplied.	For the approach of a PuT vehicle one has the possibility to either provide a typical travel time or to model the trapeze explicitly. Both of them may not be provided.
10188	At least one calling-point pair has more than one log-in or log-off point and one common log-in and log-off point.	In case there is more than one log-in or log-off point, log-in points must be different from log-off points.
10189	Element 'Trapeze' not found or value of 'TimeDistanceReg' not valid.	See 10103.
10190	Element 'TimeDistance' or ‘Probability’ not found.	See 10103.
10196	Element 'PedDetectorID' not found or value not valid or attribute 'pedestrianDetId' of pedestrian detector not valid.	See 10103.

6.1.2 Warnings and Information

Id	Message	Explanation
10012	Info: Epics runs in Balance fixed time.	This message means that PTV Epics does not control but only implements the requirements of PTV Balance or the preferred stage intervals.
10013	EPICS log level is neither 0 nor 1 nor 2 nor 3 nor 4. Set to 3.	The EPICS log level must be exactly one of these numbers. Else it is set to the default level (3).
10014	Info: Preferred interval of stage.	PTV Epics received a new T-Time and therefore changed the preferred interval of a stage.
10015	Info: EPICS runs according to its fallback routine.	This message means that for some reason PTV Epics had to revert to its fallback routine, which is the fixed time signal program. After a given time (at the moment 90s) it will try to optimize again.
10016	Length of fixed-time cycle of signal program > cycle length.	The distances of the interstages of the fixed-time program add to more than the actual cycle length.
10017	Two interstages of fixed-time cycle of at least one signal program too close to each other for EPICS.	PTV Epics has to keep every stage for at least one second. Probably the reason for this message is that two interstages follow each other immediately in one signal program. PTV Epics cannot exactly reproduce such sort of program.
10029	Vehicle time forced logoff = 0 for at least one calling-point pair. Forced log-off will take place after 180 s.	When PTV Epics is supposed to do forced log-off of PuT vehicles this time has to be > 0. Else the default value 180s (usually too large) is used.
10097	Neither valid way through phase space nor most critical signal group found. Probably supply error.	Uncritical during the first cycle; precise check of data provision, especially of signal-group parameters necessary when appearing later.
10098	Invalid value (< 0 or > 200) found for factorStops.	The relative weighting of the number of stops can be changed from outside during runtime. This warning appears in case the new value is outside the allowed range. Else PTV Epics writes the information 10108.
10099	factorStops is < 0 or > 200.	The provided value for the relative weighting of the number of stops is outside the allowed range. Correct data provision.
10108	Info: New value found for factorStops. Value changed.	See 10098.
10123	At least one signal group has demand in VS-PLUS and does not in EPICS. One vehicle added in EPICS.	A signal group has a demand in the VS-PLUS picture but does not in the PTV Epics picture. Demand generated. May happen if a vehicle passes the signal group during Amber. If the warning comes frequently one has to compare the PTV Epics and VS-PLUS data provision.
10124	At least one signal group has demand in EPICS and does not in VS-PLUS. All vehicles removed in EPICS.	A signal group has a demand in the PTV Epics picture but does not in the VS-PLUS picture. Demand deleted. May happen if a vehicle passes the signal group during Amber. If the warning comes frequently one has to compare the PTV Epics and VS-PLUS data provision.
10141	Less EPICS IT-demands than VS-PLUS traffic streams found for at least one signal group.	The number of PrT demands of PTV Epics should be equal to the number of PrT traffic streams of VS-PLUS. Must be corrected in the data provision.
10142	Demand on clone of at least one signal group found before demand on original signal group.	Demands on cloned signal groups must be listed after the demand on the original signal group; particularly important if the signal

Id	Message	Explanation
		group has PrT as well as PuT demands since the number of waiting PrT vehicles is relevant for the PuT demand.
10163	At least one PT signal group has Cyclicality 0 and TClosedMax 0.	In this case the PuT signal group comes only due to the weight of the vehicle. Harmless as long as this weight is chosen considerably higher than that of a PrT vehicle; severe else since the signal group might not get Green at all.
10180	Info: ID of current program.	Switched to the program mentioned.
10186	At least one signal group has no demand and Cyclicality 0. Cyclicality set to 1.	A signal group without demand can obviously not come on demand, therefore put to cyclical here. Data provision should be thought over and corrected.
10192	Info: Two neighbouring stages concerning to Epics.	PTV Epics tries to determine ‘neighbour’ stages, i.e. stages with a similar signal image. An example is one direction with and without its left-turn having Green. Interstages between such neighbours are favoured by PTV Epics, so in the example the stage for turning left should usually come before or after its straight neighbour.
10193	Calling-point pair contains more vehicles within TRENDS than within Epics.	One should carefully look into this warning because actually registration and deregistration in TRENDS and PTV Epics should take place synchronously. Maybe a calling-point pair has not been assigned correctly.
10194	Calling-point pair contains less vehicles within TRENDS than within Epics. One vehicle subtracted from Epics queue.	An interesting point that however usually does not mean a misbehaviour of TRENDS or PTV Epics. Rather the forced log-off of TRENDS in terms of the function stmp – if provided - has presumably done its job. PTV Epics adapts its number of PuT vehicles accordingly. However, in case this function is not provided the warning is in fact alarming and one should carefully look into it.
10195	Cyclicality of clone set to cyclicality of father signal group.	Of course clones must have the same cyclicality as the father signal group. If this is not the case it is done here by force. Should be corrected in the data provision.
10199	Info: New Balance cycle found.	PTV Balance changed its optimized stage sequence.
10200	No valid way through phase space found in first iteration step. Most critical signal group determined.	A signal group is endangered to hurt its maximum red time and PTV Epics will try to give it green accordingly.
10201	No way through phase space found without hurting TClosedMax.	If this message appears only from time to time and not more often than about five times in a row, it usually does not mean a misbehaviour of PTV Epics. Rather PTV Epics presumably exhausted completely the limit given by TClosedMax and still recognizes the stage afterwards although the interstage is already running since it starts with flashing green of some signal groups. This problem has already been mentioned when discussing the file ‘Epics_Action_XXX.txt’, see Chapter 4.2. The situation becomes critical when the message comes without interruption, therefore after the tenth time an error (10096) is written.
10202	Number possible interstages < 1!	There is a stage for which (due to detector occupancies) currently no interstage is allowed that leads out of it. This situation actually must not occur and it indicates either a data provision error or an error in the PTV Epics algorithm. Should maybe get promoted to an error.
10203	Number possible interstages <> 2!	This warning can only appear in combination with RBC. It means: It is expected that exactly 2 interstages are allowed, namely one interstage that switches on a certain pedestrian signal and its

Id	Message	Explanation
		pendant that fails to do so. The actually found number of interstages is different.
10204	No program found to the transferred cycle length. Program ID unchanged. PTV Balance not respected.	This warning can only appear in combination with RBC. Here the program number has to be derived from the cycle length handed over by PTV Balance. In case this is not possible, the recommendation of PTV Balance will not be respected.
10205	No valid start stage found.	This warning can only appear in combination with RBC. No valid start stage has been found for the PTV Balance cycle.
10206	More than one possible start stage found.	This warning can only appear in combination with RBC. Start stage of the PTV Balance cycle not unique.
10207	Info: Start stage found.	This information can only appear in combination with RBC. It is a pleasant one and means that a valid and unique start stage for the PTV Balance cycle has been found.
10209	Empty node name ('') received.	When a program change took place or during the initialization procedure PTV Epics did not receive a valid node name but an empty string.
10210	Cycle length transferred differs from cycle length supplied.	This warning can only appear in combination with RBC. Comparison of transferred and provided cycle length failed.
10211	Height of trapeze not constant. Height of trapeze set to first value.	If one supplies an approach trapeze by hand, the heights must fulfil the sequence (0, h, h, 0). In case the message appears the sequence is obviously (0, h1, h2, 0) with h1<>h2. PTV Epics automatically puts both heights to the first transferred value, i.e. to h1.
10212	Area of trapeze too large (> 128). Height of trapeze reduced.	An approach trapeze should have an area of about 128 corresponding to one vehicle in PTV Epics units. Only thus it can be assured that the provided weights are comparable and responsible for the prioritization. Therefore in case of a too large area the height of the trapeze gets reduced.
10213	Stage not decoded for more than one cycle.	This message is severe. It usually means that the controller got stuck in some signal image. Such thing may happen when e.g. the LFIX follows immediately an interstage. This is not allowed since else during a program change the signal groups to be switched latest are not switched at all. That again leads to the system taking on a ‘stage‘ that PTV Epics does not know since it is not a provided stage.
10214	No valid Balance cycle found (no valid start stage found).	In the following some messages are described that all have the meaning that PTV Epics cannot interpret the frame signal plan handed over by PTV Balance and therefore refuses to respect it. In case this happens from time to time please contact PTV. There is either an error in the writing of the signal plan by PTV Balance or in the interpretation by PTV Epics.
10215	Invalid latest start of interstage. Maybe not all relevant T-times used in structures.	See 10214. Maybe also the T-times or the PTV Balance parameters are not correctly provided. In combination with TRENDS there might also be the reason that not all relevant T times are used in the TRELAN structures – only those can be read from the TRENDS kernel. In this case one can solve the problem by using the T times in the structure (even without functionality).
10216	Number partial intersections > 0 but no possible PTV Balance stage sequence found. Original stage sequence used.	See 10214.

Id	Message	Explanation
10217	Info: Epics is initialized.	Initialization routine of PTV Epics completed successfully.
10218	Info: Length of interstage raised.	Due to minimum green times the length of an interstage has been raised.
10219	Info: New T-time or interstage start time received.	PTV Epics reports that PTV Balance changed a T-time. A good sign since it shows that PTV Balance is actually working.
10220	Failure on demand detector ignored; detector put on permanent demand.	A demand detector is disturbed. PTV Epics puts it to permanent demand.
10221	TFreeMin of clone set to TFreeMin of father signal group.	Of course clones must have the same minimum green time as the father signal group. If this is not the case it is done here by force. Should be corrected in the data provision.
10222	TClosedMax of clone set to TClosedMax of father signal group.	Of course clones must have the same maximum red time as the father signal group. If this is not the case it is done here by force. Should be corrected in the data provision.
10223	TFreeMin < 5 or > 10 read in for one signal group.	A hint to an unusual minimum green time of a signal group. Usually to neglect, sometimes an indication of a typo.
10224	No active interstage leaving a certain stage found in at least one program.	There is at least one stage with no interstage leading out of it. Uncritical if the stage can never be reached. Else it must be corrected.
10225	TClosedMax (TClosedMin) < 30 (20) or > 150 read in for one signal group.	A hint to an unusual maximum or minimum red time of a signal group. Usually to neglect, sometimes an indication of a typo.
10226	Signal group is not under EPICS control and has TClosedMax (TClosedMin) > 0. TClosedMax (TClosedMin) set to 0.	For a signal group that is not controlled by PTV Epics a maximum o rminimum red time does not make sense.
10227	Length of interstage reduced.	Obviously the interstage has not been defined with its minimum length. Should be done if possible. Else PTV Epics does, which means another possible source of errors.
10228	Failure on detector; values from parallel detector used.	A detector reports a failure. There is however a parallel detector whose values are used as a substitute. Of course one should repair the failure if it is permanent.
10229	Failure on detector; ignored since there is another detector further away from stop line.	A detector reports a failure. There is however an upstream detector and it is this detector that is decisive for the efficiency of PTV Epics, see Chapter 2.4. Of course one should repair the failure if it is permanent.
10230	Failure on detector; values from parallel detector (different direction!) used.	A detector reports a failure. There is however a parallel detector whose values are used as a substitute. However the replacement is questionable since it is positioned on a lane that leads to a different direction. Therefore PTV Epics is affected. Of course one should repair the failure if it is permanent.
10231	Identical TimeDistance for two succeeding subQueues found in demand.	Two Sub Queues should not have the same time distance. It simply does not make sense. The efficiency of PTV Epics might be affected. Please correct.
10233	Less detectors (or stealing detectors or occupancy detectors) with valid ID found than declared. Number reduced.	Probably detector-ID 0 provided. Not allowed.
10236	Signal group found with Cyclicality = 0 and TClosedMax = 0. TClosedMax set to 180.	Such a signal group only gets green when it has a demand and only then TClosedMax is respected. Therefore for the case of a demand, a value > 0 has to be provided. The value should be

Id	Message	Explanation
		chosen reasonably, e.g. equally the cycle length. The default value (180s) is usually too large.
10237	Same code found for SignalGroupsOn as well as for SignalGroupsOff.	Cannot be. Please control the switch-on and switch-off vectors of the concerned interstage.
10238	tB read in is < 10 or > 80.	A hint to an unusual time-requirement value of a demand. Maybe an indication of a typo. Unit: 1/10 s.
10239	Number counting detectors or number occupancy detectors unequal number lanes.	Maybe data provision of the demand not correct, especially number lanes.
10240	Balance wants to switch off without switching on.	See 10214.
10241	Balance wants to switch on without switching off.	See 10214.
10242	Parameters of netCommand invalid or netCommand no longer valid.	Balance command contains invalid data (examples are wrong intersection number or cycle length or checksum) or is out of time.
10243	Time horizon is not a multiple of the time grid used in the first optimization routine.	So far the time grid is 5 s. In case this warning appears (e.g. TH = 103 s) the log file Epics_StageSequence-1 does not cover the whole time horizon.
10244	Balance wants to switch on and off at the same time.	See 10214.
10245	TFreeMin of signal group hurt.	May not appear. In case this message comes Epics made a severe mistake and returns an error. Please report to PTV.
10246	Wrong switching order: off, on, on, off.	See 10214.
10247	Wrong switching order: off, off, on, on.	See 10214.
10248	Wrong switching order: on, off, off, on.	See 10214.
10249	Wrong switching order: on, on, off, off.	See 10214.
10250	Total length of all splits unequal cycle length.	See 10214. This warning can only appear in combination with RBC.
10251	Total length of splits 1 and 2 (3 and 4) unequal total length of splits 5 and 6 (7 and 8).	See 10250.
10252	Total length of first (second) four splits or splits until first overlap phase unequal cycle length.	See 10250.
10253	No signal group found that has green during one particular stage.	See 10250.
10254	Next signal group to be switched off has split 0.	See 10250.
10255	No valid next interstage found.	See 10250.