

RAILROAD & Co.TM

TrainControllerTM All Editions



Version 10

Change Description

January 2025

RAILROAD & Co.TM

**TrainControllerTM
All Editions**

Version 10

Change Description

January 2025

Copyright© Freiwald Software 1995 - 2025

Contact: Freiwald Software
 <http://www.freiwald.com>

All rights reserved.

The content of this manual is furnished for informational use only, it is subject to change without notice. The author assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of the author.

Table of Contents

About this Document	8
The Editions of TrainController™	8
RAILROAD & CO. TrainController™ 10 Change Description	9
The new Functions at a Glance	9
1 Introduction	14
1.4 Fundamentals of Use	14
User Interface Design	14
3 Train Control.....	15
3.3 Throttle and Brake	15
Start Delay	15
3.5 The Speed Profile	15
Adjustments to the Start Behaviour	15
3.6 Headlights, Steam and Whistle	17
Operations on Starting and Stopping	17
5 The Visual Dispatcher I	18
5.8 Arranging Indicators and Markers in a Block.....	18
The Block Wizard	18
5.10 Spontaneous Runs.....	19
5.12 Execution of Schedules	19
Waiting Time, Start Delays and Acceleration.....	19
Timing of Operations.....	20
5.13 AutoTrain - Creation and Execution of Schedules made Easy	20
Auto Train by Drag & Drop.....	21
AutoTrain Direct Mode.....	22
Auto Train and Consisting	23
5.15 Successors of a Schedule.....	23
6 Facilities	24
General.....	24
6.1 Single Track	24
General.....	24
Fully automatic Single Tracks	27
6.2 One-Way Track	27
General.....	27
Fully automatic One-Way Tracks	29
6.3 Speed Restriction Section.....	29
General.....	29
Common vs. individual Signals and Speed Limits	30
Fully automatic Speed Restriction Sections.....	31
6.4 Tunnel.....	31
General.....	31

Fully automatic Tunnels	34
6.5 Branching	34
General.....	34
Example	36
Fully automatic Branching.....	38
6.6 Exit.....	38
General.....	38
Exits and the Beginning of Schedules.....	38
Effects of Exits on Running Schedules.....	40
Train Priority	40
Example	42
Fully automatic Exits	44
6.7 Siding	44
General.....	44
Example	46
Fully automatic Sidings	47
6.8 Station.....	47
General.....	47
6.9 Booster	47
General.....	47
6.10 Flow Control.....	47
General.....	47
States of a Flow Control	49
Rules	49
Trigger	50
Fully automatic Flow Control	50
Flow Control and other Objects	50
6.11 Shunting Area	50
General.....	50
Shunting of Trains in a single Block.....	51
Stopping in Shunting Areas	53
Setup of Shunting Areas	55
Fully automatic Shunting	58
6.12 Line Up Area	58
General.....	58
Line-Up of Vehicles and Trains in a single Block	59
Stopping in Line Up Areas	62
Setup of Line Up Areas.....	62
Fully automatic Line Up	64
Manual Line Up	64
Comparison with Line Up in Version 9	64
6.13 Run-Round Loop	65
General.....	65

Run-Round.....	66
Setup of Run-Round Loops	68
Fully automatic Run-Round Loops.....	68
Run-Round Loops and other Objects.....	69
6.14 Engine Change	69
General.....	69
Engine Change.....	71
Setup of Engine Changes	72
Fully automatic Engine Changes	72
Engine Change without Exit at the Beginning	73
Engine Changes and other Objects	75
6.15 Gradient.....	75
General.....	75
Operation of a Gradient	76
Setup of Gradients	78
Fully automatic Gradients.....	78
Gradients and other Objects.....	79
6.16 Hump Yard.....	80
General.....	80
Move Up, Separation and Roll Down	81
Fully automatic Hump Yards.....	84
Hump Yards and other Objects.....	85
6.17 Hidden Yard.....	85
General.....	85
Entry	87
Exit.....	87
Optimizing the use of the Hidden Yard	88
Parking Trains in a Row	91
Line Up.....	91
Parking a Train in more than one Block	91
Holding Blocks for specific Trains	91
Origin Block	92
Engine Functions	92
Full automatic Hidden Yards	92
Hidden Yards and AutoTrain.....	93
Hidden Yards and Spontaneous Runs	94
Hidden Yards as Station.....	94
Hidden Yards and other Objects	94
6.18 Engine Shed.....	94
General.....	94
Engine Sheds and Turntables.....	96
6.19 Display Cabinet - Train Lift – Fiddle Yard.....	97
General.....	97

Basic Principle	98
Train Lifts and Paternosters	100
Display Cases and Fiddle Yards	101
Entry and Exit Tracks on different Levels	103
Example: Operation of a Traverser	104
Cabinets, Schedules and other Facilities	106
9 The Traffic Control and the Simulator	108
Manual operation of feedback indicators	108
11 Advanced Train Control.....	109
11.3 Approved Trains.....	109
Attributes of Train Descriptions	109
Conditional Train Descriptions.....	109
Using Train Descriptions to describe Similar Trains	110
14 Extended Control and Monitoring Functions.....	112
14.4 Operations	112
19 Appendix	113
19.1 Migrating Existing Data Files from TrainController™ 9	113
Open existing Files	113
Line Up and Move Up	114
Index	115

About this Document

RAILROAD & CO. is the leading product line of computer programs for digitally or conventionally controlled model railroads. It contains the following members:

- **TrainController™** is the world's leading software for model railroad computer control.
- **TrainProgrammer™** is the program, which makes programming of DCC decoders as simple as a few clicks with your mouse.
- **+SmartHand Mobile™** is the world's premium handheld railroad control system designed for computer controlled model railroads.
- **+4DSound™** is a module, that recreates realistic spatial sound effects for each model railroad layout controlled by **TrainController™** without the need to install on-board sound into each decoder.
- **+Street™** is a module for control of car systems with **TrainController™**.
- **+Cargo™** is an extension for **TrainController™** to simulate the production, consumption and transport of freight on the model railroad.
- **+Net™** is a module, that allows to control your layout with a network of several computers running **TrainController™**.

The Editions of TrainController™

TrainController™ is offered in three variants:

- **TrainController™ Bronze** provides a low-cost entry into computer controlled model railroads. It is primarily designed for users with small and medium size layouts and average requirements. Novice users, who do not know **TrainController™**, may consider doing their first steps with **TrainController™ Bronze**. The reduced functionality of this variant makes it easier to identify and to learn the basic functions of **TrainController™**.
- **TrainController™ Silver** is the successor of the established and well-known version **TrainController™ 5**. It addresses users with high demands and also users, who are not reluctant to puzzle to accomplish individual goals.
- **TrainController™ Gold** is the flagship of the **TrainController™** family and in a class of its own. **TrainController™ Gold** is primarily designed for users with supreme requirements, who want to operate their layout like the real professionals. While **TrainController™ Silver** is already able to operate even very large layouts, **TrainController™ Gold** provides much more convenience, efficiency and security for design and operation – especially for larger layouts.

RAILROAD & CO. TrainController™ 10 Change Description

This document provides an overview of the features, which are unique in **TrainController™ 10** and which distinguish **TrainController™ 10** from **TrainController™ 9**. It is mainly intended for users, who know **TrainController™ 9**, and who want to learn about the differences between **TrainController™ 10** and **9**. It is assumed, that the reader is familiar with **TrainController™ 9** and the **TrainController™ 9 Users Guide**. New users of **TrainController™ 10** should focus on the **TrainController™ 10 Users Guide** rather than this document.


The numbers of the particular chapters and sections of this document are inherited from the concerning chapters and sections of the **TrainController™ 9 Users Guide**. This allows readers, who are familiar with that Users Guide, to implicate the contents of both documents. This is also the reason for the gaps occurring in the numbering.

All text sections, that describe features of **TrainController™ 10 Gold**, which are not provided by **TrainController™ 10 Silver** or **Bronze**, are marked with a specific marking on the left side of the text in the same way as this section. Contents marked in this way do not apply to **TrainController™ 10 Silver** or **Bronze**. Users of this program version or readers only interested in **TrainController™ 10 Silver** or **Bronze** can safely ignore these contents.

All text sections, that describe characteristics of **TrainController™ Silver**, which do not apply to **TrainController™ Gold** or **Bronze**, are marked with a specific marking on the left side of the text in the same way as this section. Users of this program version or readers only interested in **TrainController™ Gold** or **Bronze** can safely ignore these contents.

Unless otherwise indicated all screen shots show the user interface of **TrainController™ Gold**. This means in particular, that user interface options may be visible, which are not available in **TrainController™ Silver** or **Bronze**.

The new Functions at a Glance

In the following all new functions are listed, that apply to **TrainController™ 10**. Outstanding improvements are marked with an .

Misc:




1. New default user interface design with improved support for Windows 10
2. More new MS Office like user interface designs (see page 14, “User Interface Design”).
3. Various improvements of program performance.

4. Improved support and better performance for multi core processors.
5. Conversion of the activation from the USB stick to activation via the Internet.



Hardware and Digital Systems:


6. The complete functionality provided by +**Hardware** Version 9 is available in the **TrainController™** core product.

Train Control:

7. The speed of a train can be directly set in the **Engines + Trains** list by editing the displayed speed value. In order to start a train or to change its speed it is not necessary anymore to open a **Train Window** with this train.
8. The speed, length or weight of trains as well as conditions can now also be specified in train descriptions, which are embedded in other objects. It is no longer necessary, to create separate train descriptions for this purpose.
9. With train descriptions, it is now also possible to check for zero speed, weight or length. In this way it is in particular possible to specify train descriptions, which apply to trains standing still.
10.  In addition to the start delays on schedule level, which apply to the beginning of a schedule or after stops, it is also possible to specify an individual delay for each engine. Such delay is applied before a train is set in motion. See page 19.
11.  Improved support for playing starting sounds when starting locomotives with sound decoders (see page 15).
12.  Vehicle operations can be executed automatically when a train starts, stops, turns its direction or decelerates to threshold speed as well as when a start delay in a schedule begins (see page 17).


Dispatcher and Automatic Operation:

13. With **TrainController™ 10** trains can be optionally set to carry out spontaneous runs by simply setting the train in motion. In this way spontaneous runs with full protection and routing can be directly executed, for example, by simply starting a train with a handheld throttle.
14. Separate commands for AutoTrain for creation of schedules in Edit Mode or execution of schedules outside Edit Mode.
15.  A new direct mode allows to perform AutoTrain by Drag & Drop in the same way as in Version 8 (see page 22)
16.  The new Block Wizard enables the initial setup of blocks with feedback indicators and markers with a few mouse clicks (see page 18).

17. In offline mode, feedback indicators in blocks can be switched on or off directly in the switchboard or dispatcher with a click of the mouse without having to open a traffic control (see page 108).
18.  The minimum acceleration time is a new schedule rule, with which light and powerful trains (e.g. locomotives without cars) can be prevented from accelerating too suddenly. A human train driver is simulated, so to speak, who does not accelerate too much and within a certain period of time when the train is light, and who turns the speed control fully up to get going, when the train is heavy. For details see page 19.
19. The successor of a schedule can now be started with a *similar* train (see page 23).
20. Start and destination operations can be performed at the beginning and end of a schedule sequence; in addition to and independently of the start and destination operations of the schedules included in the sequence.
21. In schedule sequences it is now possible to change the train between every two partial schedules and to specify various options for each change which influence the selection of the train.
22. In the settings for a block in a schedule, you can specify that the block may only be passed in one direction. In addition to the known global setting for one-direction operation, which applied to all schedules, this can now also be set individually for certain schedules, only for AutoTrain or only for spontaneous runs.
23. In the settings for a block in a schedule that is on a turntable, you can not only specify the direction in which trains leave the turntable, but also specify which trains this specification applies to.
24. If there are cars and no engines in the destination block (or blocks) of an **AutoTrain** run, it is optionally possible, to create a new train set with these cars upon arrival of the started train in the destination block (see page 23) .
25. The names of objects can now be stored or evaluated in variables.
26. The **Repeat Count** option in the **General** properties of a schedule can be specified as formula with variables instead of a fixed value.
27. The **Try Again** option in the **General** properties of a schedule can be specified as formula with variables instead of a fixed value.
28. The waiting time in the **Schedule Specific Section Settings** dialog box can be specified as formula with variables instead of a fixed time.
29. The additional start delay in the same dialog box can be specified as formula with variables instead of a fixed value.
30. Speed limits in blocks, turnouts, routes or schedules can be specified as formula with variables instead of a fixed value.
31. The value of almost all rules of schedules can be specified as formula with variables instead of a fixed value.
32. The new system state **All Blocks locked** indicates, whether all blocks are locked with the **Lock all Blocks** menu command.

33. The new system state **All Schedules locked** indicates, whether all schedules are locked with the **Lock all Schedules** menu command.
34. Object variables can be tested for zero objects, i.e. whether a reference to an object is stored in a variable or whether the variable is empty. This can be used, for example, to query via a variable whether a train is currently performing a schedule or is in a station.

Facilities:

35.  Facilities are the outstanding innovation in **TrainController™ 10 Gold**. Trains only running from A to B are out. Other programs can do the same. With facilities, however, standard tasks such as the control of single-track lines, train stations or passing loops but also complex processes in hidden yards, locomotive changes, hump yards, pushing operation at gradients, paternoster and much more can now be implemented also by less experienced users much faster and more easily than with the basic functions known from previous versions of the program. In total, **TrainController™ 10 Gold** contains more than 15 different types of facilities for different control tasks and maneuvers. A separate view is provided in the dispatcher for easy creation and management of the various facilities.
36. Single Track lines as a new type of facility allow the simple control of single-track sections, e.g. to prevent opposing trains to lock each other.
37. With one-way track facilities trains can be forced to pass certain blocks only in one direction.
38. Speed restriction sections force trains to pass certain blocks only with yellow signal or with limited speed.
39. The tunnel facility enables the storage and restoration of states of locomotive functions when entering and exiting invisible parts of the model railroad.
40. With flow control facilities the utilization of certain parts of the model railroad layout can be regulated.
41. At siding facilities, trains with lower priority stop in order to let other trains with higher priority pass or overtake.
42. At branching facilities, various criteria can be used to specify which path should be preferred.
43. At exits, various criteria can be used to specify which trains or tracks should be selected as preferred, e.g. at the start of a schedule or for exiting a station.
44. In shunting areas, trains may enter blocks where other trains or vehicles are already located in order to form a common train set with them.
45. In line up areas, several trains can be lined up one behind the other in the same block.
46. Hidden yard facilities control covered parking areas in an intelligent way and with a myriad of variants and options.
47. In engine shed facilities, the parking of locomotives is controlled with particularly attractive options when a turntable is available.

- 48. Run-round loops enable locomotives to attach to the opposite end of their train.
- 49. With engine changes you can exchange the engines in trains under control of the computer.
- 50. In gradient facilities pusher engines can assist trains to climb a steep grade.
- 51. With a hump yard facility you can establish your own hump yard on your model railroad with cars that automatically roll down the hill, as in the prototype.
- 52. Cabinets can be arranged as a specific type of facility to easily include train display cases, fiddle yards, paternosters or train lifts in the control of your model railroad.

Inspector Window:

- 53. Display of the parameters used for the adjustment of the brake compensation of engines (speed and ramp - not in **TrainController™ Bronze**).
- 54. Display of the parameters of all Facilities.
- 55. Display of the details and usage of system states and events.

1 Introduction

1.4 Fundamentals of Use

User Interface Design

In addition to the user interface styles known from previous versions **TrainController™ 10** provides the following new or changed user interface designs.

- Railroad & Co. 10
- Several Microsoft 365 (Office) styles
- A simplified custom style with the possibility to adjust the primary colors of the user interface to personal taste.

3 Train Control

3.3 Throttle and Brake

Start Delay

In addition to the start delays, that can be specified for an entire schedule or for individual blocks in a schedule it is furthermore possible in **TrainController™ 10 Gold** to specify an individual delay for each engine or car. Such delay is applied before the train is set in motion. This time span can be utilized to perform additional operations (e.g. playing motor sounds, switching on the head lights, etc.) before the train starts (see below).

This delay is always effective when a train is set in motion, i.e. not only when executing schedules, but also when driving manually.

If different delays apply to several vehicles in a train set, then the maximum of all delays is used.

For each vehicle, it is furthermore possible to specify operations, which are executed, when the start delay begins (see section 3.6, “Headlights, Steam and Whistle”).

3.5 The Speed Profile

Adjustments to the Start Behaviour

In the real world, locomotives do not start immediately, when they are moved from the stand. Rather, the diesel engine first picks up speed or a steam locomotive releases the brakes and fills the cylinders slowly. It therefore takes a certain amount of time until the locomotive actually starts moving.

Many sound decoders simulate this process by not starting immediately from a standstill with a low speed step, but first playing the corresponding noises.

If, for example, the threshold speed of a locomotive is at speed step 2, and the decoder does not start the motor when it receives speed step 1, but first plays a 3-second long starting sound, then **TrainController™ 10 Gold** can support this by initially sending

speed step 1 (to trigger the starting noise) and then after 3 seconds by setting the locomotive in motion at threshold speed.

The speed steps and their timing required for this can be configured in **TrainController™ 10 Gold** individually for each locomotive. By default, a starting locomotive is immediately set to the specified threshold speed. But it is possible to define further, lower speed steps for a fixed period of time before the threshold speed is set.

The screenshot shows the 'Advanced Fine Tuning - Passenger' window with the 'Start Behaviour' tab selected. The window has a title bar with a close button (X). Below the title bar are three tabs: 'Threshold Speed', 'Speed Profile', and 'Start Behaviour' (which is active), and a 'Decoder' tab. The main area contains a table with two columns: 'Speed Step' and 'Time'. The table has 13 rows. The first two rows are populated: Speed Step 1 with Time 2000, and Speed Step 2 with Time 1000. The remaining rows are empty. To the right of the table is a 'Time' dropdown menu set to '2000'. Below the table is a section 'Enabled by Function:' with a dropdown menu showing 'User 2'. Below that is a 'Function State:' section with a dropdown menu also showing 'User 2'. At the bottom left is a 'Library...' button. On the right side of the window, there are several buttons: 'OK', 'Cancel', 'Help', 'Delete', 'Delete All', a speed slider, a 'Speed:' section with 'Internal: 0', 'Profile: 0 km/h', and 'Decoder: 0', and a 'Simplified Profile...' button.

Speed Step	Time
1	2000
2	1000
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

Time: 2000

Enabled by Function:
User 2

Function State:
User 2

Library...

OK
Cancel
Help
Delete
Delete All

Speed:
Internal: 0
Profile: 0 km/h
Decoder: 0

Simplified Profile...

Diagram 1: Adjustments to the Start Behaviour

Diagram 1 shows the adjustments to the start behaviour of a locomotive with two entries. When this locomotive is started, first speed step 1 is set for 2 seconds (2000 milliseconds) and then speed step 2 for 1 second. Finally (after 3 seconds) the threshold speed of the locomotive is set and the locomotive starts to move.

The effect of the adjustments can be made dependent on the status of a locomotive function. In Diagram 1 it is set that the locomotive function "User 2" must be switched on in order for the speed steps to be issued. If this function is switched off when the locomotive starts up, it starts without these adjustments and the threshold speed is set immediately. This is useful for sound decoders, for example, where the noises described above

are only played when the sound is switched on, but otherwise no delays occur. With the help of the locomotive function, which can be specified here, the software can be adapted to the applicable start behaviour.

3.6 Headlights, Steam and Whistle

Operations on Starting and Stopping

In **TrainController™ Gold**, vehicle functions can also be automatically performed as operations (see section 14.4) when the associated vehicle starts, stops, toggles its direction or is about to be decelerated to threshold speed. This is also possible when the vehicle is driven manually with the train window.

The operations for the threshold speed are initiated during deceleration of a train and a few seconds before the train actually reaches threshold speed. This provides time to play braking sounds, for example, already at a speed above threshold speed. This additional lead time can be reduced by inserting a delay operation at the beginning of the operations list for the threshold speed.

It is also possible to specify operations, which are executed in that moment, when the start delay of an engine begins. Since they are executed at the beginning of the start delay, noises, for example, can be played here before the train actually starts to move.

5 The Visual Dispatcher I

5.8 Arranging Indicators and Markers in a Block

The Block Wizard

With the new Block Wizard, you can set up a new block with indicators and markers in no time at all. The Block Wizard offers a selection of typical arrangements of detectors in a block with 1 - 3 occupancy sensors or momentary track contacts as well as for two-way or one-way operation.

Block Wizard

Select Setup:

- Bidirectional - 3 Occupancy Sensors
- Bidirectional - 1 Occupancy Sensor
- Directional Right - 2 Occupancy Sensors
- Directional Left - 2 Occupancy Sensors
- Directional Right - 1 Occupancy Sensor
- Directional Left - 1 Occupancy Sensor
- Bidirectional - 3 Momentary Contacts
- Bidirectional - 2 Momentary Contacts
- Bidirectional - 1 Momentary Contact
- Directional Right - 2 Momentary Contacts
- Directional Left - 2 Momentary Contacts
- Directional Right - 1 Momentary Contact
- Directional Left - 1 Momentary Contact

OK
Cancel
Help

Distances and Ramps:

Distance between both points: 150 cm

Standard braking ramp: 100 cm

Diagram 2: Block Wizard

After choosing a setup that best suits the arrangement of the detectors in your block, a certain distance and the braking ramp are specified. These three details are sufficient for the software to initialize the block with indicators and markings.

The distance to be specified depends on the equipment of the block with sensors. For a block with 3 occupancy sections, for example, the distance given is the length of the section in the middle:



In the case of a block with one occupancy section, the distance between the stop marker and the opposite boundary of the occupancy section is specified:



For each selected setup, a corresponding symbol indicates, which distance is exactly meant for this setup. In practice, the process should be largely self-explanatory.

Don't worry if the configuration created with the wizard only roughly fits the block to be set up. The setup initially created with the wizard can be subsequently adapted as required with the block editor.

5.10 Spontaneous Runs

With **TrainController™ 10** it is also possible to optionally set trains so that spontaneous runs are started simply by setting the train in motion. With this you can carry out a spontaneous run with full protection and routing by simply starting the train with a handheld throttle, for example.

5.12 Execution of Schedules

Waiting Time, Start Delays and Acceleration

TrainController™ 10 Gold provides a new schedule rule, with which the acceleration of trains can be slowed down. The **minimum acceleration time** specifies the minimum time a train should need to accelerate to maximum speed. If the weight and power of the train require a longer time than specified in this rule to accelerate to maximum speed, the longer time is used. If weight and power allow a shorter time, the time specified in this rule is used. A corresponding proportion of the time specified in this rule is applied for accelerations with a smaller difference than the maximum speed of the train. This setting can prevent a light, powerful train (e.g. a locomotive without cars) from acceler-

ating suddenly. A human train driver is simulated, so to speak, who does not accelerate too much and within a certain period of time when the train is light, and who turns the speed control fully up to get going, when the train is heavy.

Timing of Operations

The following illustration shows at which points in time which operations are carried out when a train starts at the beginning of a schedule. This representation also applies analogously to starting after a stop during a schedule.

↓	Start of the Schedule. Reservation of blocks and routes.	Execute operations for the start of the schedule.
↓	Blocks and routes are reserved. Train must wait due to an exit lock or other obstacles.	
↓	Locks lifted. Train may start. The internally calculated block signal is set to green. Start delay specified in the rules of the schedule is running.	Execute operations specified in the section settings of the schedule.
↓	Start delay specified in the properties of the schedule expired. Start delay specified in the properties of the vehicle is running.	Execute operations specified for the start delay in the properties of the vehicles.
↓	Start delay executed. Train starts to move.	Execute operations specified for the start in the properties of the vehicles.

Table 1: Timing of Execution of Operations

5.13 AutoTrain - Creation and Execution of Schedules made Easy

AutoTrain™ is another outstanding feature of **TrainController™**. With **AutoTrain™** you can run automatic trains at any time during operation without the need to define schedules in advance.

In the edit mode, schedules can be created very quickly with the help of **AutoTrain™**.

Auto Train by Drag & Drop

The fastest way to run **AutoTrain™** is Drag & Drop with the mouse:

- Select the **Operation** tab (or the **Schedule** menu in the classic user interface) and call the **AutoTrain by Drag and Drop** command.
- In the edit mode select the **Edit** tab (or the **Schedule** menu in the classic user interface) and call the **Create Schedule by Drag and Drop** command.
- Move the mouse cursor to the start block.
- Press and hold the left mouse button on one of the following symbols:



, if you want the train to exit the start block to the left.



, if you want the train to exit the start block to the right.

- Hold the left mouse button pressed and drag the mouse to the block in the block diagram or in the switchboard, where the train will stop.
- Release the left mouse button over one of the following symbols:



, if you want the train to enter the destination block from the right to the left.



, if you want the train to enter the destination block from the left to the right.

- The train will now start and run automatically to the destination block. In the edit mode a schedule from the start block to the destination block is generated and stored.

The symbols can differ slightly depending on the state of the block under the mouse pointer.

Some additional options are available:







- Click this option, if you want to leave the **AutoTrain by Drag and Drop** or **Create Schedule by Drag and Drop** command turned on after releasing the mouse button. In this case another AutoTrain run can be initiated or created by drag and drop without the need to call the above command once more.



- By pressing and holding the mouse button on this symbol it is possible to specify all blocks in a station (see section 6.8, “Station”) as start blocks with exit of the train to the left. This station is determined by the block, where the mouse is pointing to. It is the station, where the block is located in.

This option works in the same way as if all blocks in the said station are specified as start blocks of a schedule.

-  Same as above but with exit of the train to the right.
-  By releasing the mouse button over this symbol it is possible to specify all blocks in a station as destination blocks with entry of the train from the right to the left. This station is determined by the block, where the mouse has been dragged to. It is the station, where the block is located in.
This option works in the same way as if all blocks in the said station are specified as destination blocks of a schedule.
-  Same as above but with entry of the train from the left to the right.
-  Click this option, if you want to perform a local AutoTrain run within a station.

The above options allow to perform or create an AutoTrain run from a station to a station with only one single drag and drop operation. It is also possible to mix the options for blocks and stations. In this way an AutoTrain run can be performed or created from an individual block to a station or vice versa.

It is also possible that instead of the symbol of a station, the symbol of another facility is displayed if the block is located in a suitable facility (see chapter 6, “Facilities”). Entry and exit then take place in or out of this facility according to the policies of this facility.

The above symbols are valid for horizontal blocks. Symbols for vertical blocks look accordingly.

AutoTrain Direct Mode

The direct mode of **AutoTrain** provides an alternative method for performing **AutoTrain by Drag and Drop**, in which the buttons described above do not need to be used, but which also offers fewer options. This mode is already familiar to users of **TrainController™ 8**.

The direct mode of **AutoTrain** is performed as follows:

- Select the **Operation** tab (or the **Schedule** menu in the classic user interface), ensure, that the **AutoTrain Direct Mode** menu option is checked and call the **AutoTrain by Drag and Drop** command.

- In the edit mode select the **Edit** tab (or the **Schedule** menu in the classic user interface), ensure, that the **AutoTrain Direct Mode** menu option is checked and call the **Create Schedule by Drag and Drop** command.
- Press the left mouse button near the exit of the block in the block diagram or in the switchboard, where the train or created schedule will start.
- Hold the left mouse button pressed and drag the mouse to the exit of the block in the block diagram or in the switchboard, where the train will stop or the created schedule will end.
- Release the left mouse button.
- If the edit mode is turned off, the train will now start and run automatically to the destination block. In the edit mode a new schedule will be created.
- The **AutoTrain Direct Mode** menu option remains checked until you alter its status.

Auto Train and Consisting

If there are cars and no engines in the destination block (or blocks) of an AutoTrain run, it is optionally possible, to create a new train set with these cars upon arrival of the started train in the destination block. The software then automatically sets the schedule rules required for this when it starts the train.

5.15 Successors of a Schedule

The following option is new in **TrainController™ 10 Gold**:

- It is possible to specify, that the successor schedule will be executed with a train, which is **similar** to the train, which performed the preceding schedule (see section 11.3, “Approved Trains”).

6 Facilities

General

Facilities represent a new type of object provided by **TrainController™ 10 Gold**. Two types of facilities are already known from **TrainController™ 9**, stations and boosters.

With most type of facilities the operation of schedules can be affected. Standard tasks, such as the control of single track sections or stations can be solved also by less experienced users with the corresponding facilities much more easily in **TrainController™ 10** than with the more complex basic components as they are known from **TrainController™ 9**.

All Facilities are managed in the Facility View. The Facility View is a part of the Dispatcher Window.

Facilities consist of blocks and optionally routes. Facilities are established by assigning blocks and routes to them. This is done in a similar way like assigning blocks and routes to schedules. Settings of blocks or routes, which are specific to certain facilities can be edited in a very similar way like the schedule specific section settings.

The various types of facilities are described in detail in the following chapters.

6.1 Single Track

General

We start with the description of single tracks, because these are the most simple type of facilities. A single track can be used to manage single track sections on your layout.

Single tracks are only effective for trains under control of a schedule (including AutoTrain or spontaneous runs).

With a single track

- opposing trains are prevented to lock each other in single track sections on your layout.
- trains in the same direction can be allowed or prohibited to use the single track at the same time.
- the simultaneous use of the single track can be limited to a maximum number of trains. It is for example possible to specify, that up to 2 trains may follow each other through a certain single track section at the same time.
- the subsequent use of the single track in the same direction can be limited to a maximum number of trains. It is for example possible to specify, that up to 3 trains may pass the single track section subsequently in the same direction. When 3 trains passed the section e.g. from the right to the left, then no other train can pass this section to the left, until at least one other train has passed the section from the left to the right. This makes it possible to provide more variety and also to prevent trains from jamming in front of the single-track line because other trains are constantly coming from the opposite direction without interruption.

Single tracks are established by assigning blocks to them. Routes can also be added for the sake of clarity, but have no meaning. This assignment is done in a similar way like assigning blocks and routes to schedules. Each block or route can only be assigned to at most one single track. It is not possible to assign a block or route to two single track objects or more at the same time. A route, however, can connect two blocks, which are assigned to different single tracks.

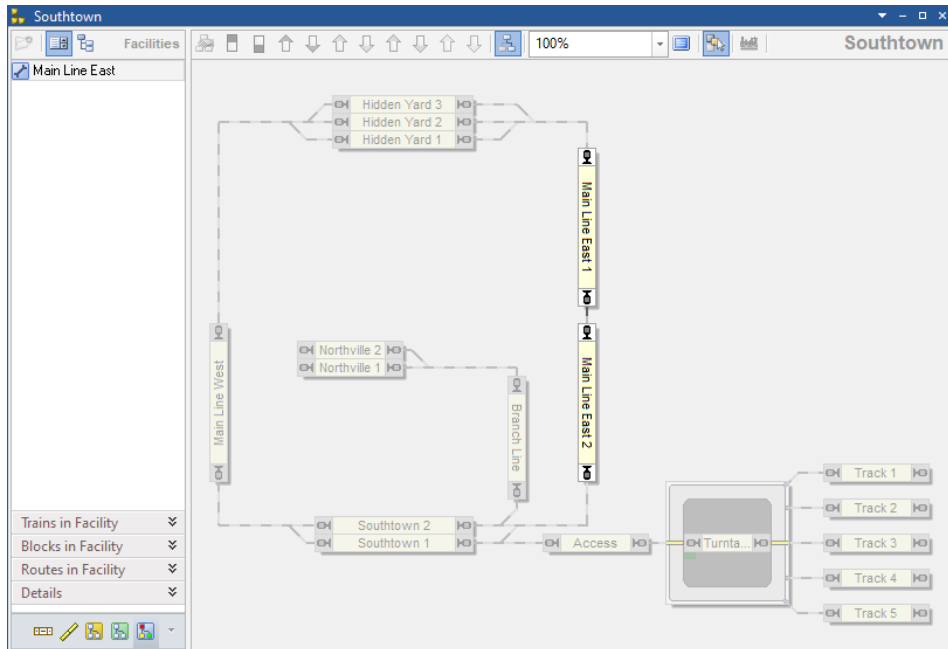


Diagram 3: Facility View with Single Track in the Dispatcher Window

Diagram 3 shows a layout with a single track facility. The single track “Main Line East” is currently selected and the block diagram of “Main Line East” is currently visible in the right part of the dispatcher window. The blocks “Main Line East 1” and “Main Line East 2” are assigned to single track “Main Line East”.

Trains can reserve a block in the single track only, if at least one block at the other end of the single track is available. Trains coming from the “Hidden Yard” blocks can reserve the single track only, if at least one block in “Southtown” is available. Trains coming from “Southtown” can reserve the single track only, if at least one of the “Hidden Yard” blocks is available. In this way trains passing the single track in different directions can be prevented from locking each other.

Note for experienced users: Single track sections work very similar to critical sections, but provide more options.

Fully automatic Single Tracks

In order to include single tracks in the automatic mode, you do not have to do anything else than

- Creation of a single track.
- Assignment of the blocks belonging to the single track.
- Check that the rules for the schedules involved take single tracks into account.

6.2 One-Way Track

General

A one-way track can be used to manage one-way operation on your layout.

One-way tracks are only effective for trains under control of a schedule (including AutoTrain or spontaneous runs).

With a one-way track

- trains can be forced to pass certain blocks only in one direction.
- the one-way operation can be limited to certain schedules or trains. That means, that some trains or trains under control of some schedules may pass the one-way track in both directions, while other trains are only allowed to pass the one-way track in the permitted direction.

One-way tracks are established by assigning blocks to them. Routes can also be added for the sake of clarity, but have no meaning. This assignment is done in a similar way like assigning blocks and routes to schedules. Each block or route can only be assigned to at most one one-way track. It is not possible to assign a block or route to two one-way track objects or more at the same time. A route, however, can connect two blocks, which are assigned to different one-way tracks.

Each block in a one-way track is marked by specifying a direction as a passing block (e.g. passing block to the right).

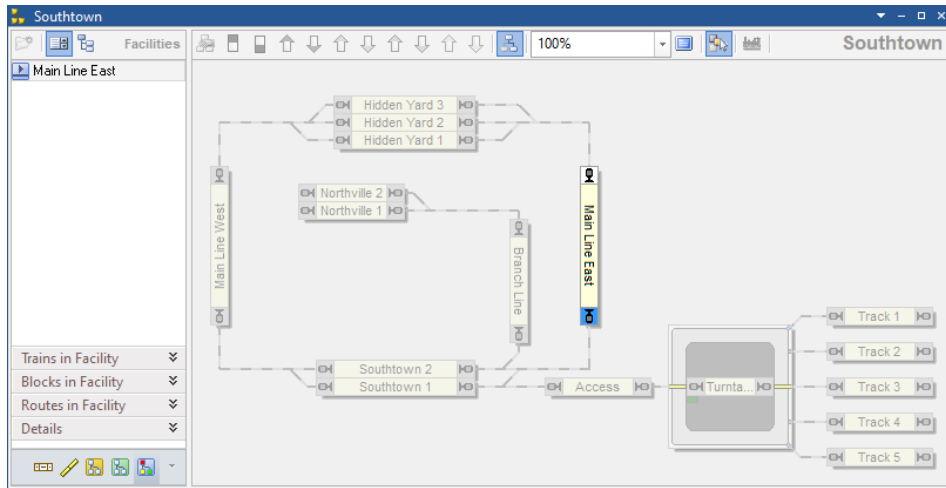


Diagram 4: Facility View with One-Way Track in the Dispatcher Window

Diagram 3 shows a layout with a one-way track facility. The one-way track “Main Line East” is currently selected and the block diagram of “Main Line East” is currently visible in the right part of the dispatcher window. The block “Main Line East” is assigned to one-way track “Main Line East” as passing block to the bottom.

Trains, which are affected by the one-way track may pass the block “Main Line East” only to the bottom.

Fully automatic One-Way Tracks

In order to include one-way tracks in the automatic mode, you do not have to do anything else than

- Creation of a one-way track.
- Assignment of the blocks belonging to the one-way track.
- Mark each block as passing block in the desired allowed direction.
- Check that the rules for the schedules involved take one-way tracks into account.

6.3 Speed Restriction Section

General

A speed restriction section can be used to enforce yellow signal aspects and speed limits on your layout.

Speed restriction sections are only effective for trains under control of a schedule (including AutoTrain or spontaneous runs).

With a speed restriction section

- trains can be forced to pass certain blocks only with yellow signal and/or limited speed.
- A speed restriction section can be limited to certain schedules or trains. That means, that some trains or trains under control of some schedules may pass the speed restriction section as fast as possible, while other trains are only allowed to pass the speed restriction section with limited speed.
- Unlike speed limits, which are entered in blocks or turnouts, for example, speed restriction sections can be decommissioned or made dependent on a condition. In this way it can be achieved that speed restriction sections are only effective temporarily or in certain situations.
- Speed restriction sections can be directional. This means that the yellow signal or speed limit can be effective only to trains running in a specific direction.
- In contrast to speed limits for blocks or turnouts, which are only lifted when the section in question is released, speed limits in speed restriction sections can optionally be lifted as soon as the train enters a subsequent block.

Speed restriction sections are established by assigning blocks and routes to them. This assignment is done in a similar way like assigning blocks and routes to schedules. Each block can only be assigned to at most one speed restriction section in the same direc-

tion. It is not possible to assign a block to two speed restriction sections or more in the same direction at the same time. A route, however, can connect two blocks, which are assigned to different speed restriction sections.

Each block in a speed restriction section is marked by specifying one or both directions as a passing block (e.g. passing block to the right).

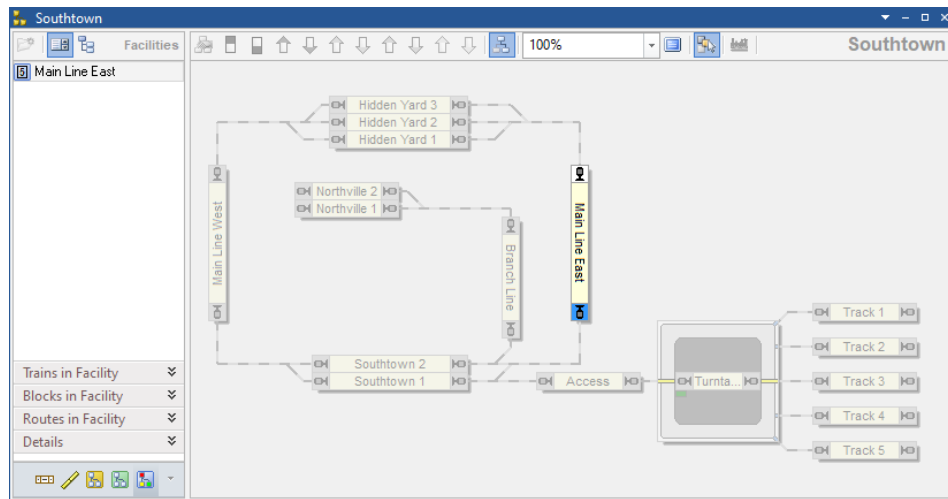


Diagram 5: Facility View with Speed Restriction Section in the Dispatcher Window

Diagram 3 shows a layout with a speed restriction section. The speed restriction section “Main Line East” is currently selected and the block diagram of “Main Line East” is currently visible in the right part of the dispatcher window. The block “Main Line East” is assigned to the speed restriction section “Main Line East” as passing block to the bottom.

Trains, which are affected by the speed restriction section may pass the block “Main Line East” to the bottom with limited speed and to the top as fast as they can.

Common vs. individual Signals and Speed Limits

The yellow signal or speed limit valid for all blocks and routes in a speed restriction section is specified in the properties of each speed restriction section. This is the common signal or speed limit of this speed restriction section.

Furthermore it is possible to specify an individual signal or speed limit in the section settings of each block or route in a speed restriction section.

The common signal or speed limit applies to all blocks and routes in the speed restriction section for which no individual signal or speed limit has been set.

Fully automatic Speed Restriction Sections

In order to include speed restriction sections in the automatic mode, you do not have to do anything else than

- Creation of a speed restriction section.
- Assignment of the blocks and routes belonging to the speed restriction section.
- Mark each block as passing block in the desired directions.
- Specify a common signal or speed limit for the entire speed restriction section or individual signals or speed limits for specific blocks or routes.
- Check that the rules for the schedules involved take speed restriction sections into account.

6.4 Tunnel

General

Tunnels are another relatively simple type of facilities. A tunnel can be used to manage engine functions in certain sections on your layout.

With a tunnel

- the current state of the functions of a train entering the tunnel can be saved and set to a certain status.
- The status of the functions of a train leaving the tunnel can be restored to their previous state.

Tunnels are established by assigning blocks to them. Routes can also be added for the sake of clarity, but have no meaning. This assignment is done in a similar way like assigning blocks and routes to schedules.

Each block at the entrance of a tunnel is marked by specifying a direction as an entry block (e.g. entry block to the right). Each block at the exit of a tunnel is marked by specifying a direction as an exit block (e.g. exit block to the left). Each entry or exit block can only be assigned in the same direction to at most one tunnel. It is not possible to assign a block as entry or exit block with the same direction to two tunnels or more at the same time.

Blocks between entry and exit blocks can also be added to a tunnel for the sake of clarity, but have no meaning.

For each tunnel it is possible to select the functions, which should be managed by the tunnel and the state of each function to be set when a train enters the tunnel.

When a train enters an entry block of a tunnel, the current state of all selected functions is saved and each function is set to the specified status (off or on). When the train enters the exit block of a tunnel, the state of all selected functions is restored to their previous state.

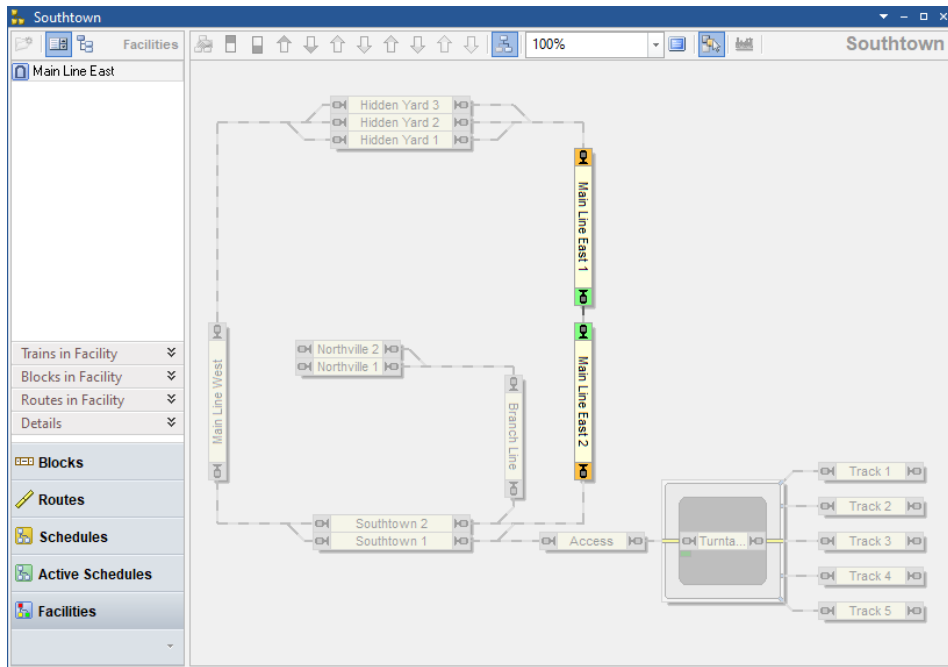


Diagram 6: Facility View with a Tunnel in the Dispatcher Window

Diagram 6 shows a layout with a tunnel facility. The tunnel “Main Line East” is currently selected and the block diagram of “Main Line East” is currently visible in the right part of the dispatcher window. The blocks “Main Line East 1” and “Main Line East 2” are assigned to the tunnel “Main Line East”.

The block “Main Line East 1”, for example, is set as an entry block to the bottom and an exit block to the top. The function states of each train entering “Main Line East 1” to the

bottom will be stored and set to the specified status. The function states of each train entering “Main Line East 1” to the top will be restored to the previously stored status.

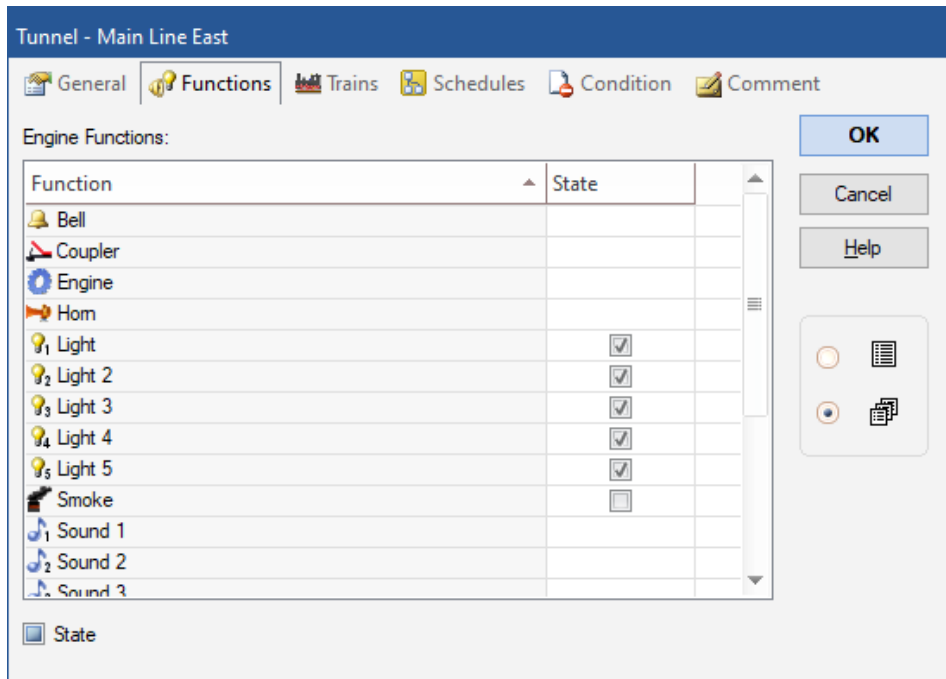


Diagram 7: Functions setting of a Tunnel

Diagram 7 shows the function settings of a tunnel. When a train enters this tunnel all light functions should be turned on, the smoke function should be turned off and all other functions should not be affected.

The effect of a tunnel can be limited to certain trains, to certain schedules and/or to manually operated trains not under control of a schedule.

Fully automatic Tunnels

In order to include tunnels in the automatic mode, you do not have to do anything else than

- Creation of a tunnel.
- Assignment and marking of the blocks belonging to the tunnel.
- Check that the rules for the schedules involved take tunnels into account.

6.5 Branching

General

A branching can be used to control the choice between different available paths in a schedule.

A Branching is established by assigning blocks to them. Routes can also be added for the sake of clarity, but have no meaning. This assignment is done in a similar way like assigning blocks and routes to schedules.

Each block in a branching is marked by specifying a direction as an entry block or passing block. (e.g. entry block to the right). Each entry block can only be assigned in the same direction to at most one branching. It is not possible to assign a block as entry block with the same direction to two branching objects or more at the same time.

A branching is only effective for trains under control of a schedule (including AutoTrain or spontaneous run).

A branching comes into play if two paths within a schedule contain the same entry block of this branching in the appropriate direction of travel. In this case, the software checks whether there is a passing block in the appropriate direction of travel behind this entry block on each of the two paths. If this is the case, the settings of these two passing blocks are compared with one another. In the end, the software chooses the path with the more favorable criteria.

The selection is controlled by the following criteria:

- **Obstacle:** The path with the minor obstacles (e.g. entry or exit locks, other trains, unfulfilled conditions, etc.) is chosen. All obstacles between the entry block and the first found passing block are evaluated. This criterion largely corresponds to the normal selection of paths in schedules without the use of a branching.
- **Priority of Block:** The path through the passing block with the highest priority is chosen. For this criterion, a number can be entered as a priority for each passing block of a branching.
It is optionally possible to use the priority as quantifier for the selection. The probability results from the ratio of the priorities of the passing blocks. If one block has priority 3000 and another has priority 1000, for example, the path via the first block is selected with a probability of 3 to 1.
- **Oldest:** The path is chosen with the passing block through which no train has been driving for the longest time.
- **Length:** The path with the shortest passing block into which the train fits is chosen.
- **Level in Cabinet:** If there are passing blocks in a cabinet (see section 6.19), a passing block on the level closest to the current position of the cabinet is preferred.

Furthermore:

- The user can specify the criteria in a list according to their importance. The first criterion is the most important. If no decision can be made based on this criterion, the second criterion is used. If this does not allow a decision, the third criterion in the list applies. And so on.
- The search ends for each path with the first passing block found.
- The search also ends with the first block behind the entry block, which is located on both paths.
- If no passing block is found on one of the two paths, the branching has no influence on the selection of the path.
- It is possible to limit the use of a branching to certain schedules or to certain trains.

Example

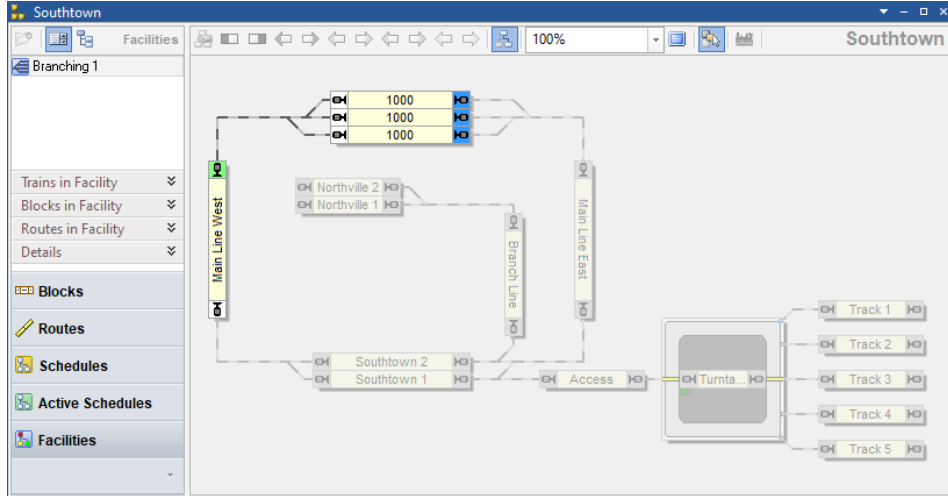


Diagram 8: Facility View with Branching in the Dispatcher Window

Diagram 8 shows a layout with a branching facility. The branching “Branching 1” is currently selected and the block diagram of “Branching 1” is currently visible in the right part of the dispatcher window. The block on the left side is marked as entry block to the top (green marking), the three blocks at the top are assigned to branching “Branching 1” as passing blocks (blue marking).

The numbers in the blocks indicate the priority entered in the respective block of the branching.

The general properties of the branching are specified as follows, for example:

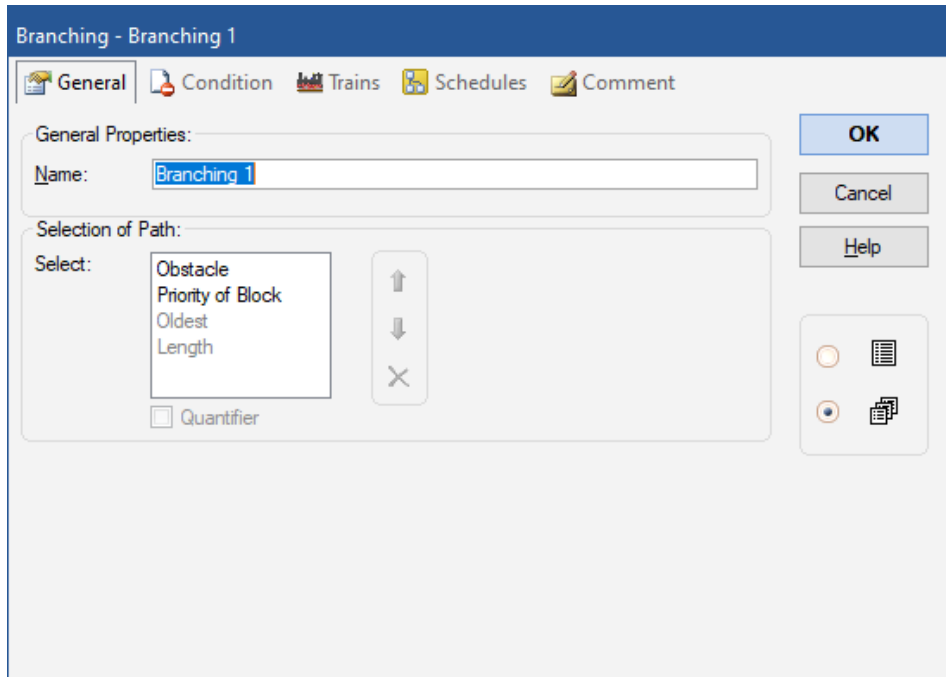


Diagram 9: General properties of a Branching

A train passing the entry block of the branching to the top will choose one of the passing blocks depending on availability. This is achieved by placing the **Obstacle** criterion at the top of the list. If more than one block is available, then the block with the highest priority (see Diagram 8) is selected, because **Priority of Block** is the second most important criterion in the list.

The **Oldest** and the **Length** criteria are not used for this branching. They could be included, however, if desired.

Fully automatic Branching

In order to include a branching in the automatic mode, you do not have to do anything else than

- Creation of a branching.
- Assignment and marking of the blocks belonging to the branching.
- Check that the rules for the schedules involved take branching objects into account.

6.6 Exit

General

Exits can be used to allow trains to exit from an area of the layout according to your own preferences. Typical examples are the exits from stations, from which passenger trains should leave before freight trains.

Exits are established by assigning blocks to them. Routes can also be added, but have no meaning. This assignment is done in a similar way like assigning blocks and routes to schedules.

Each block in an exit is marked by specifying a direction (e.g. valid to the right). Each block can only be assigned in the same direction to at most one exit. It is not possible to assign a block with the same direction to two exits or more at the same time.

Exits are only effective for trains under control of a schedule (including AutoTrain or spontaneous run). Exits work in two ways. At the beginning of a schedule with several start blocks, exits can influence the selection of the train with which the schedule is carried out. During a running schedule, the train controlled by the schedule can be temporarily held in order to allow other trains to leave.

Exits and the Beginning of Schedules

Exits can come into play at the beginning of a schedule. If two or more of the start blocks of the schedule with a train in it are located in the same exit, the software prefers the train with the more favorable criteria.

The preference is controlled by the following criteria:

- **Priority of Block:** The train in the block with the highest priority is preferred. For this criterion, a number can be entered as a priority for each block of an exit.

It is optionally possible to use the priority as **Quantifier** for the selection. The probability results from the ratio of the priorities of the blocks in the exit. If one block has priority 3000 and another has priority 1000, for example, the first block is selected with a probability of 3 to 1.

- **Priority of Train:** The train with the highest effective priority is preferred (see page 40).

It is optionally possible to use the priority as **Quantifier** for the selection. The probability results from the ratio of the priorities of the trains. If one train has priority 3000 and another has priority 1000, for example, the first train is selected with a probability of 3 to 1.

- **Oldest:** The train is preferred which has not been operated by a schedule for the longest time.
- **Length:** The longest train is preferred.
- **Count:** The train is preferred, which reserves the higher number of blocks.
- **Level in Cabinet:** When exiting a cabinet (see section 6.19), preference is given to the train that is located on a level that is closest to the current position of the cabinet.

Furthermore:

- The user can specify the criteria in a list according to their importance. The first criterion is the most important. If no decision can be made based on this criterion, the second criterion is used. If this does not allow a decision, the third criterion in the list applies. And so on.
- For the start of a schedule, an additional **Range** option causes an additional random factor. If the range is set to 2, for example, one of the two trains with the highest previously determined preference is selected first. If this train cannot be started, the attempt to start is made with one of the two trains that have the highest preference among the remaining trains, and so on. If the range is 3, the selection is always made randomly among 3 trains, etc. If range equals 1, there is no additional random factor.
- The preference is only determined for the blocks that are located in the same exit with the appropriate direction.
- It is possible to limit the use of an exit to certain schedules.
- Aspects outside an exit can also influence the selection of the train. So even if an exit determines a higher preference for train 1 than for train 2, train 2 can still move first if other aspects enforce this.

Effects of Exits on Running Schedules

Exits can also be used to cause trains in a running schedule to give preference to other trains before they exit themselves. During the execution of a schedule, an exit thus can affect several schedules. A train in a schedule is stopped to allow a train in another schedule to continue first.

The preference is controlled by the same criteria as described in the previous section. The **Quantifier** and the **Range** option described above, however, are only available for the selection of the train at the beginning of a schedule.

The effect of exits can be determined during the execution of a schedule for one or more of the following cases:

- When the train starts right after the beginning of the schedule.
- After every scheduled stop.
- After every operational, unscheduled stop.
- For all moving trains that want to leave the exit. With this definition, moving trains are stopped in the exit to allow other trains with higher priority, which wait already there, to exit first.

Train Priority

For each exit or other facilities, which work with train priorities it is possible to assign a number to each vehicle, train description or vehicle group. These numbers are used to calculate the effective priority of trains that the facility can then work with.

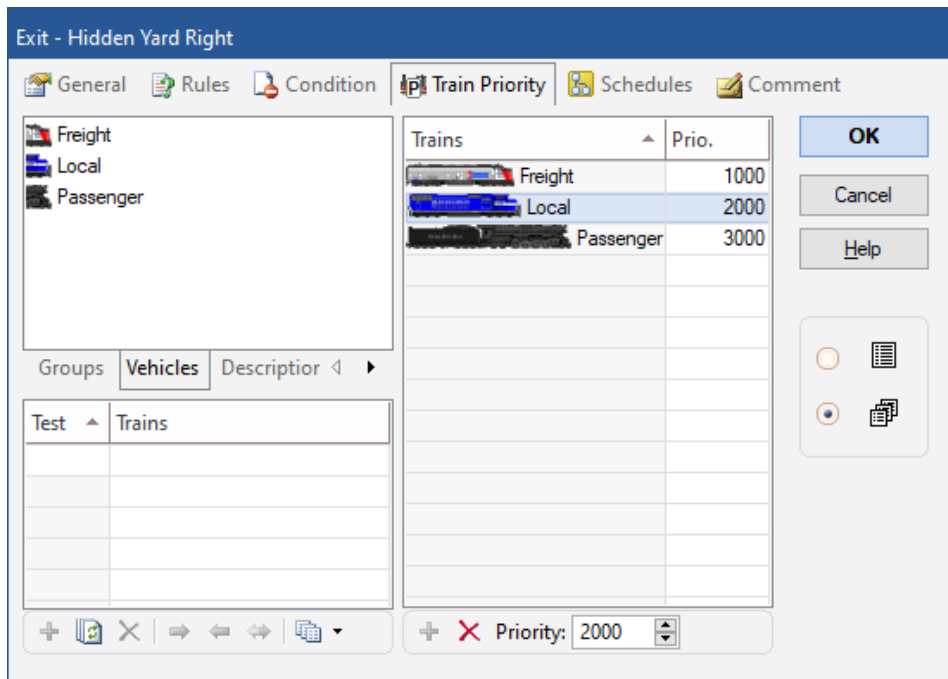


Diagram 10: Priority of Trains of an Exit

- The priority entered for a vehicle, train description or vehicle group always applies only to the associated facility. Different values can be entered for the same vehicle, description or group in different facilities.
- The effective priority of an individual vehicle results from the maximum of the value assigned to it and the values of all vehicle groups in which the vehicle is contained.
- The effective priority of a train results from the maximum of the effective priorities of its vehicles and the values of all train descriptions to which the train fits.
- If no priority is entered for a vehicle, train description or vehicle group in a facility, then 1000 is the default value. This means that there are a sufficient number of gradations available, which can be used to define low-priority (value below 1000) or high-priority (value above 1000) trains, based on the standard value.

Example

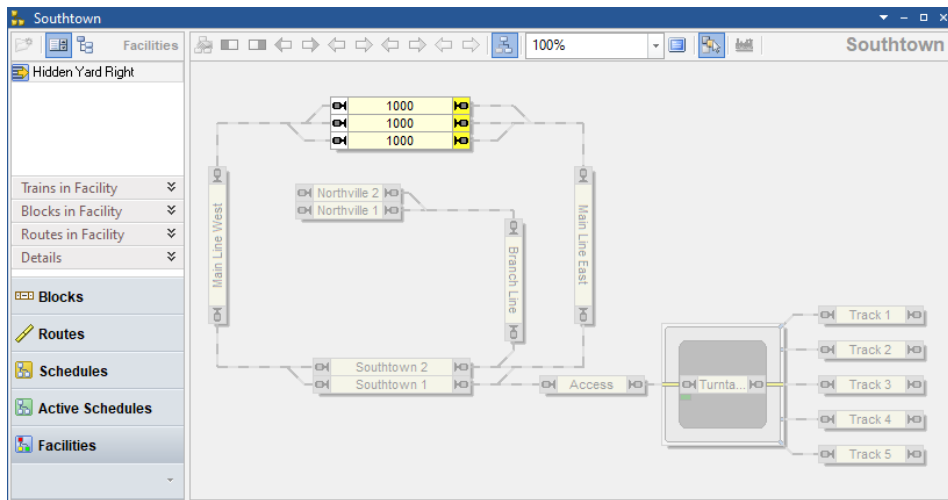


Diagram 11: Facility View with Exit in the Dispatcher Window

Diagram 11 shows a layout with an exit facility. The exit “HiddenYard Right” is currently selected and the block diagram of “Hidden Yard Right” is currently visible in the right part of the dispatcher window. The blocks “Hidden Yard 1”, “Hidden Yard 2” and “Hidden Yard 3” are assigned to exit “Hidden Yard Right” with effect to the right (light yellow marking).

The numbers in the blocks indicate the priority entered in the respective block of the exit.

The general properties of the exit are specified as follows, for example:

The screenshot shows a software window titled "Exit - Hidden Yard Right". It has a tabbed interface with the following tabs: "General" (selected), "Rules", "Condition", "Train Priority", "Schedules", and "Comment".

Under the "General" tab, there is a section "General Properties:" containing a "Name:" label and a text input field with the value "Hidden Yard Right".

Below this is a section "Exit Selection:" containing an "Exit:" label and a list box. The list box contains the following items: "Priority of Block", "Priority of Train", "Oldest", "Length", and "Count". To the right of the list box are three buttons: an up arrow, a down arrow, and a close button (X).

Below the list box is a "Range:" label and a spin box with the value "1". To the right of the spin box is a checkbox labeled "Quantifier", which is currently unchecked.

On the right side of the dialog, there are three buttons: "OK", "Cancel", and "Help". Below these buttons is a group box containing two radio buttons and two icons (a document and a folder).

Diagram 12: General properties of an Exit

A schedule with start blocks “HiddenYard 1”, “Hidden Yard 2” and “Hidden Yard 3” to the right will prefer the block with the highest priority (see Diagram 11), because **Priority of Block** is the most important criterion in the list. If two blocks have the same priority, then the train with the highest priority (see Diagram 10) will be selected.

Fully automatic Exits

In order to include exits in the automatic mode, you do not have to do anything else than

- Creation of an exit.
- Assignment and marking of the blocks belonging to the exit.
- Check that the rules for the schedules involved take exits into account.

6.7 Siding

General

Sidings can be used to automatically control train crossings or overtaking maneuvers.

Sidings are established by assigning blocks to them. Routes can also be added for the sake of clarity, but have no meaning. This assignment is done in a similar way like assigning blocks and routes to schedules.

Sidings contain

- **Holding blocks:** These are the blocks, where trains wait, until prior-ranking trains passed.
- **Passing blocks:** These are the blocks through which prior-ranking trains pass waiting trains. Optionally, it can also be specified that these blocks are preferably used for the passage, even if there is no other train to be waited for.
- **Entry blocks:** These are the blocks in which incoming trains are checked to see if they need to be directed to a holding block to let a prior-ranking train pass.
- **Control blocks:** These are the blocks that are checked to see if there is a prior-ranking train in them.

Each entry, passing or holding block can also be specified as control block. But it is also possible to specify additional blocks as control blocks.

Specifying holding blocks as control blocks, for example, is useful if you want entering trains to wait in a siding, until prior-ranking trains already waiting there (in a holding block) have left the siding.

Each block in a siding is marked by specifying a direction as a entry block (e.g. entry block to the right), control block, passing block or holding block. Each entry or holding block can only be assigned in the same direction to at most one siding. It is not possible to assign a block as entry or holding block with the same direction to two sidings or more at the same time.

Sidings affect the path only for trains under control of a schedule (including AutoTrain or spontaneous run).

A siding comes into play if a path within a schedule contains an entry block of this siding in the appropriate direction of travel. In this case, the software checks whether there is another prior-ranking train in a control block of the same siding in the appropriate direction of travel. If this is the case, the train controlled by the schedule is directed to an available holding block in the siding with the appropriate direction.

The process is controlled by the following options (without claim to completeness):

- Other trains are considered prior-ranking if they have a higher priority (see page 40) and are located in a control block or, as an option, reserve a control block of the same siding.
- You can specify, whether other trains are only taken into account as prior-ranking, if they are moving or if they are controlled by a schedule.
- You can specify, whether other trains in control blocks are taken into account, if the control block is only reserved, but not the current block of that train. This also takes into account more distant approaching trains and not only trains that are already in a control block.
- You can specify a maximum holding time. After this time a train waiting in this siding will continue its current schedule regardless of other trains in this siding. This prevents trains from being locked in a siding indefinitely.

Example

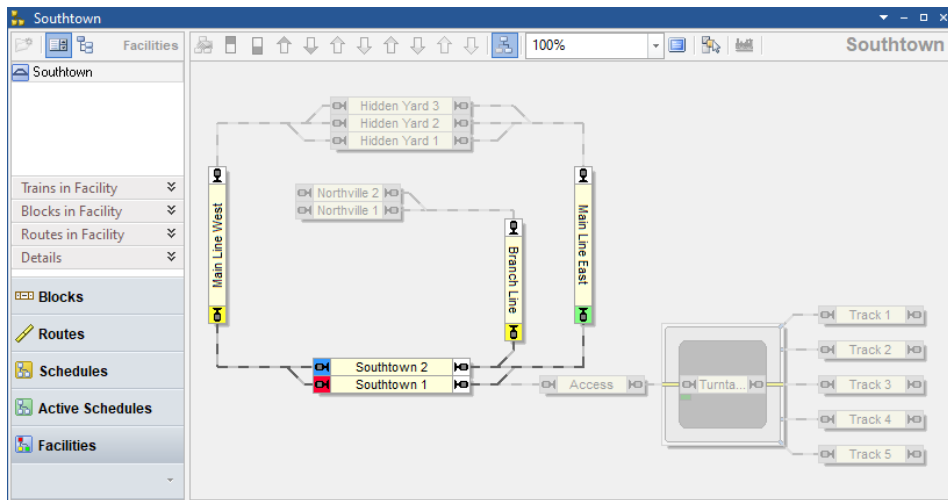


Diagram 13: Facility View with Siding in the Dispatcher Window

Diagram 13 shows a layout with a siding facility. The siding “Southtown” is currently selected and the block diagram of siding “Southtown” is currently visible in the right part of the dispatcher window. The block “Main Line East” is marked as entry block to the bottom (green marking), the blocks “Main Line West” and “Branch Line” are assigned to siding “Southtown” as control blocks (light yellow marking), “Southtown 1” is a holding block (red marking) and “Southtown 2” is a passing block (blue marking).

A train passing the entry block “Main Line East” of the siding towards “Southtown” will be directed to the holding block “Southtown 1”, if there is another prior-ranking

train located in one of the control blocks “Main Line West” or “Branch Line” and heading to the bottom, i.e. towards “Southtown”. The train will wait there, until the prior-ranking train has passed.

Fully automatic Sidings

In order to include sidings in the automatic mode, you do not have to do anything else than

- Creation of a siding.
- Assignment and marking of the blocks belonging to the siding.
- Check that the rules for the schedules involved take sidings into account.

6.8 Station

General

Stations are already known from **TrainController™ 9**. They are only mentioned here because from **TrainController™ 10** stations are listed under the generic term facility and are also described in this chapter in the Users Guide.

6.9 Booster

General

Boosters are already known from **TrainController™ 9**. They are only mentioned here because from **TrainController™ 10** boosters are listed under the generic term facility and are also described in this chapter in the Users Guide.

6.10 Flow Control

General

Flow control objects function very similar to boosters. They provide similar options found by boosters, too, with the exception of all booster options, which are related to physical boosters. Flow control objects can be used like boosters to limit the load generated by simultaneously active trains in certain areas of the layout. Unlike boosters flow

control objects do not depend on physical booster hardware and can therefore be used on every layout without specific hardware requirements.

Flow control objects are established by assigning blocks and routes to them. This is done in a similar way like assigning blocks and routes to schedules. Each block or route can only be assigned to at most one flow control. It is not possible to assign a block or route to two flow control objects or more at the same time. A route, however, can connect two blocks, which are assigned to different flow control objects.

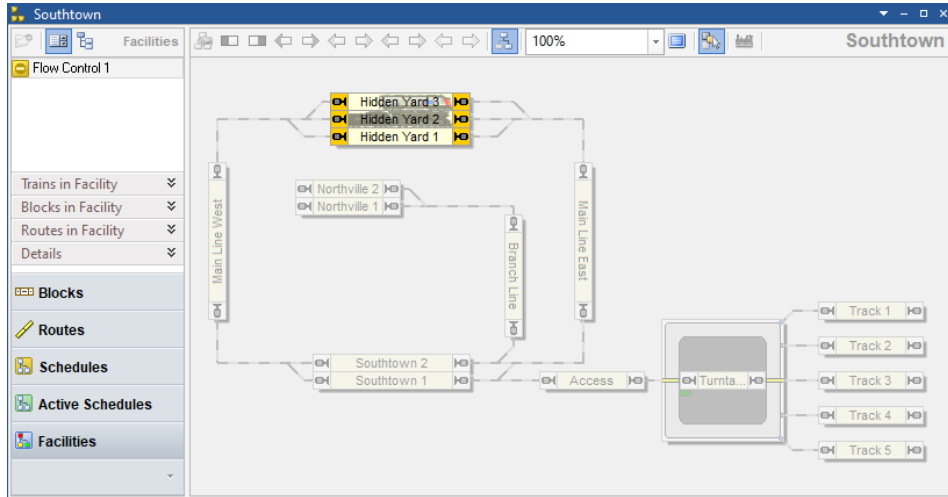


Diagram 14: Facility View with Flow Control in the Dispatcher Window

Diagram 14 shows a layout with a flow control facility. The flow control “Flow Control 1” is currently selected and the block diagram of “Flow Control 1” is currently visible in the right part of the dispatcher window. The blocks “Hidden Yard 1”, “Hidden Yard 2” and “Hidden Yard 3” are assigned to flow control “Flow Control 1”.

Flow control objects can be used to control the traffic on the model railroad as a function of the number of trains currently busy in certain areas of the layout.

States of a Flow Control

Each flow control provides the following states:




Symbol	State	Meaning
	Normal	The flow control is in normal state. Schedules may be executed in the area of this object without limitations.
	Warning	The flow control is in the warning state. Certain limitations may apply; for example trains may be prohibited to enter the area of the flow control.
	Alarm	The flow control is in the alarm state. Limitations apply; e.g. further schedules must not be executed in the area of the flow control.

Table 2: States of a Flow Control

The colors associated with each state are also displayed in the exit boxes of the blocks in the block diagram of this flow control (see Diagram 14).

Rules

In **TrainController™ Gold**, the effect of each flow control can be customized to individual needs with a variety of rules.

Depending on the state of the flow control it is possible to lock the exits or entries of blocks in the area of the flow control in order to prevent trains from leaving their current location or from entering the area from the outside. Furthermore it is possible to prevent further schedules from being executed.

For the warning or alarm state, threshold values can be specified for the number of trains, which are located or running in the area of the flow control. If the relevant number of trains reaches such threshold value, then the flow control changes to the associated state.

The option to change the flow control to warning or alarm state, when the number of active trains in the area of the flow control reaches or exceeds a certain threshold value,

and to stop the traffic in this area subsequently, provides a way to control the traffic on your model railroad layout depending on the activity of trains.

Trigger

For each state of a flow control (normal, alarm and warning) it is possible to specify an individual trigger. This causes the flow control to change to the according state each time, when the trigger specified for this state becomes true.

In this way it is possible to automatically turn a flow control to alarm state and to stop the traffic in a certain area of the layout, for example, when a certain logical condition in **TrainController™** becomes true. This feature is useful, when the states of the flow control depend on more complex conditions than plain threshold values.

Fully automatic Flow Control

In order to include flow control in the automatic mode, you do not have to do anything else than

- Creation of a flow control object.
- Assignment and marking of the blocks belonging to the flow control.
- Check that the rules for the schedules involved take flow control into account.

Flow Control and other Objects

Each state of a flow control can be evaluated in the trigger or condition of other objects. Furthermore flow control objects can perform operations, when they change their state. These features provide virtually unlimited possibilities to integrate flow control objects into the automatic control of your model railroad layout and to accomplish a smart control of the traffic on your layout.

6.11 Shunting Area

General

In shunting areas several vehicles can be arranged in one block by shunting trains in order to form a common train set. Normally it is forbidden in **TrainController™** that trains enter blocks in which another train or vehicles are already located. This ban is lifted in shunting areas. Trains are allowed to enter blocks here in order to form a train set with a train or vehicles that are already waiting there.

Shunting areas are established by assigning blocks to them. Routes can also be added for the sake of clarity, but have no meaning. This assignment is done in a similar way like assigning blocks and routes to schedules.

Each block in a shunting area is configured by defining a marker for shunting in one or both directions. Each block can only be assigned to at most one shunting area. It is not possible to assign a block to two shunting areas or more at the same time.

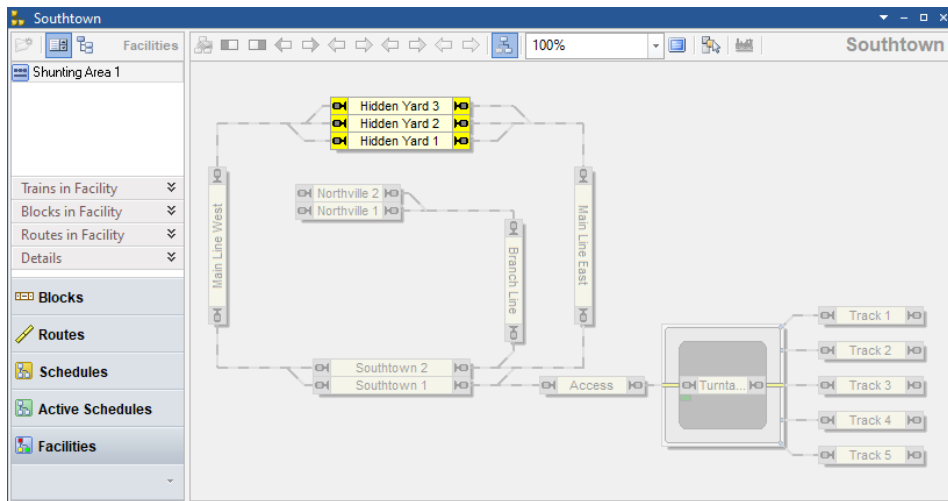


Diagram 15: Facility View with a Shunting Area in the Dispatcher Window

Diagram 15 shows a layout with a shunting area. The shunting area “Shunting Area 1” is currently selected and the block diagram of “Shunting Area 1” is currently visible in the right part of the dispatcher window. The blocks “Hidden Yard 1”, “Hidden Yard 2” and “Hidden Yard 3” are marked as blocks for shunting to both directions (light yellow marking).

Shunting of Trains in a single Block

In a shunting area, trains may enter blocks, that are already reserved by other trains or vehicles, in order to form common train sets.



Diagram 16: Shunting – Step 1

Assume, for example, a block in a shunting area with a total length of 200 cm. 20cm are kept free on both sides of the block. The remaining space for parking of trains is 160 cm. The first train to enter the block is brought to a stop using the usual methods so that it stops at a distance of 180 cm (20 cm plus 160 cm).



Diagram 17: Shunting – Step 2

Assume, that the length of the first entering train is 100 cm. In this case the software assumes, that the waiting train is located at the right end of the block and the remaining space in the left part of the block is 60 cm (160 cm minus 100cm).



Diagram 18: Shunting – Step 3

Now a second train enters the block from the left to the right. The incoming train will be stopped in a distance of 80 cm right of the left block boundary.



Diagram 19: Shunting – Step 4

Both trains now form a common train set.



Diagram 20: Shunting – Step 5

The locomotive, which pushed the new cars may leave. The situation is now similar as after step 2 and further cars may be added to the train.

Stopping in Shunting Areas

For the following consideration, we will remember how stopping in a block works normally. We are assuming the dimensions from the example in the previous section.

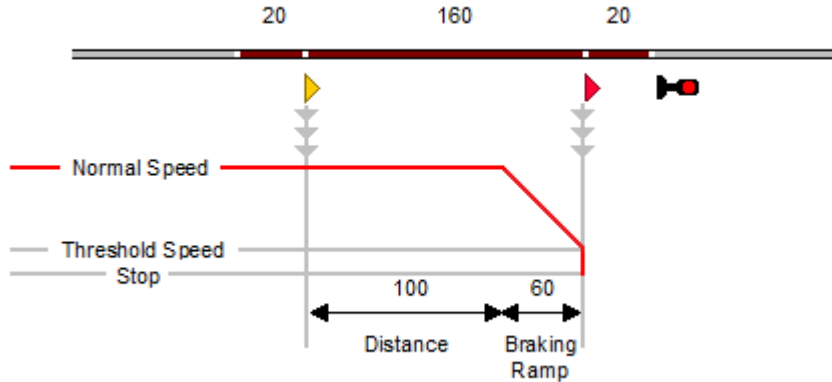


Diagram 21: Normal Stopping with Brake and Stop Markers

It is assumed that the block consists of three occupancy sections. The two outer sections are each 20cm, the middle section 160cm long. The two outer sections are each associated with a stop marker without offset. The section in the middle has brake markers for both directions with distances of 100 cm and 60 cm braking ramp.

The red line shows the speed of the first train to enter the block.

In the block editor, this block looks as follows:

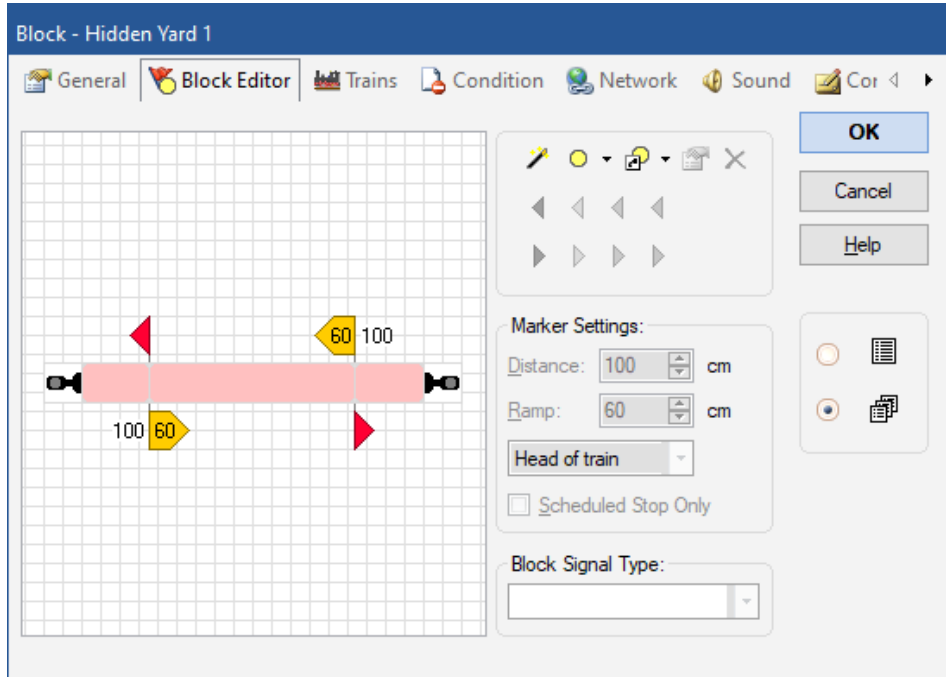


Diagram 22: Block Editor

Up to this point, this is not new. It is a conventional configuration of a block. Not much is added for shunting.



An important prerequisite for a block in a shunting area is that there is always a separate indicator on the side of the block where additional trains enter the block for coupling with waiting vehicles. This indicator must not be kept occupied by the trains already in the block. This detector is required to report the arrival of further trains. In many cases this is the case with blocks with three occupancy sections and short sections at the ends. That is why we have chosen this case for this description as well.

So that the process from the previous section works with this block, the indicator on the left side must always be switched off when another train from the left approaches trains that are already waiting. Normally, stopping in a block is controlled by the indicators that are linked to the brake and stop marker. For shunting, however, stopping is controlled by an indicator at the entry of the block. The stopping point is calculated using the position and length of the vehicles already in the block.

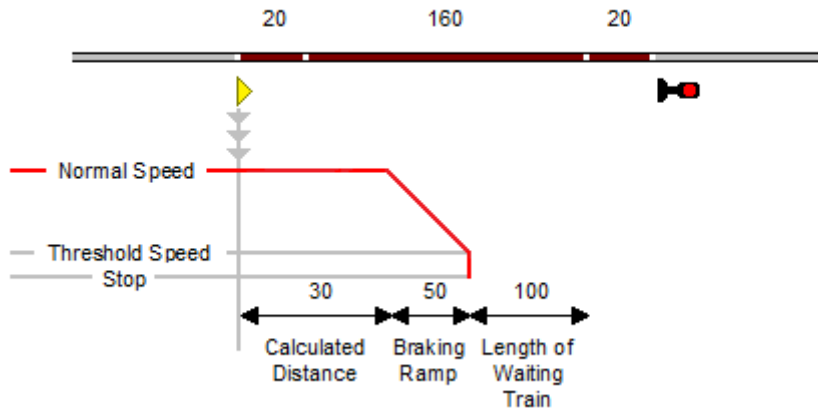


Diagram 23: Stopping for Coupling

Setup of Shunting Areas

The configuration of shunting areas is relatively simple. The software does most of the otherwise rather complex work.

To set up a block for shunting, first create a shunting area and add this block to it. This is similar to adding a block to a schedule. It is possible to add more than one block to the same shunting area.

To control the stopping in a shunting area according to Diagram 23 there is now a new type of marker for shunting. These markers are arranged in the **Special Markers** tab of the section settings for a block in a shunting area:

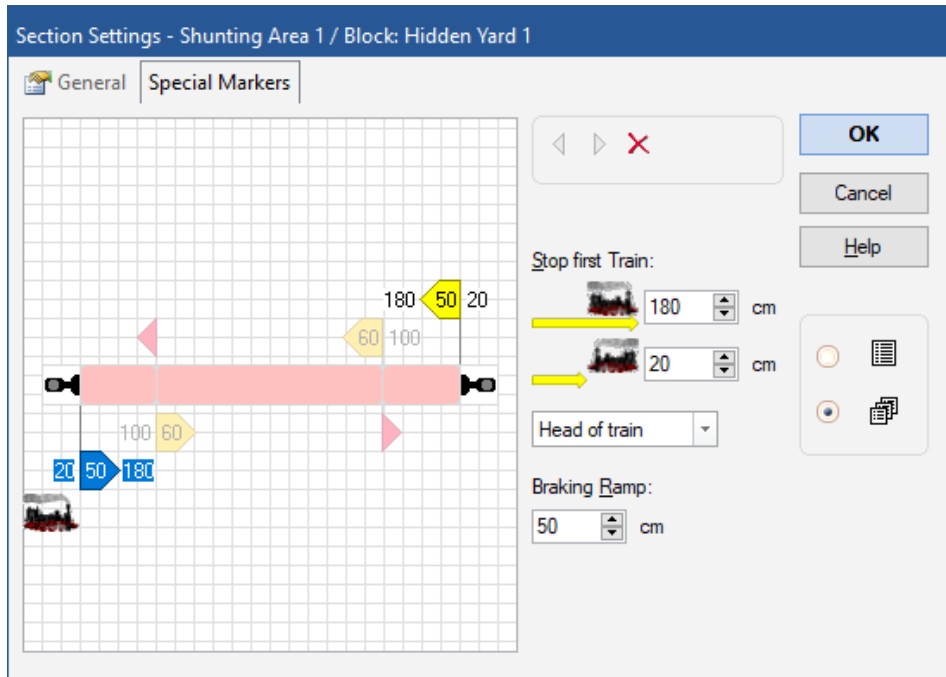


Diagram 24: Setting up a Block in a Shunting Area

Diagram 24 shows the sections settings of a block in a shunting area with the light yellow special markers for shunting in both directions.

In the **Special Markers** tab there is a window that is very reminiscent of the well-known block editor. The information from the block editor of the block is displayed in pale color for information.



Diagram 25: Shunting – Special Marker

Two distances can be specified for each marker, the first for adding vehicles to the end of the train that is waiting (e.g. for appending cars according to Diagram 16 to Diagram 20). The setting is used to describe where in the block the first arriving train will stop. This location is determined with the distance from the point at which the reference indicator of the special marker is switched on. In the above diagram this distance is 180 cm, as the head of the train comes to a stop 180 cm (= 20cm plus 160 cm) after passing the left boundary of the left section.

The second distance is used for adding vehicles at the head of a waiting train (e.g. for changing locomotives) according to the following diagrams:



Diagram 26: Adding Vehicles at the Head – Step 1

A train has entered a block and is ready for a locomotive change.



Diagram 27: Adding Vehicles at the Head – Step 2

The locomotive has been uncoupled and left the block. It is assumed that the locomotive was about 30cm long.



Diagram 28: Adding Vehicles at the Head – Step 3

Now another locomotive train enters the block from the left to the right. The incoming locomotive will be stopped in a distance of 50 cm (20cm plus 30cm) right of the left block boundary.



Diagram 29: Adding Vehicles at the Head – Step 4

Finally a new train set has been formed with another locomotive.

If you compare Diagram 19 with Diagram 29 with each other, you will see that the distances that the coupling locomotive has to cover in the block are different. The current distance therefore depends on whether the locomotive is coupled at the end (Diagram 19) or at the head (Diagram 29) of the waiting vehicles. For this reason two distances can be specified for each marker used for coupling.

In Diagram 24, the second distance is set to 20 cm, as the head of a train entering from the right comes to a stop 20 cm in front of the left boundary of the left section.



Diagram 30: Shunting – Special Marker

Only the two distances visible in Diagram 25 and Diagram 30 are important for setting up a special marker for coupling. All other distances shown in the flow charts above are automatically calculated by the software. You don't have to worry about them. They are shown here only to understand the principle.

Fully automatic Shunting

In order to include shunting in the automatic mode, you do not have to do anything else than

- Creation of a shunting area.
- Assignment and marking of the blocks belonging to the shunting area.
- Establishing the special markers.
- Check that the rules for the schedules involved take shunting areas into account.

6.12 Line Up Area

General

In line up areas several trains can be parked in one block.

Line up areas are established by assigning blocks to them. Routes can also be added for the sake of clarity, but have no meaning. This assignment is done in a similar way like assigning blocks and routes to schedules.

Each block in a line up area is configured by defining a marker for line up in one or both directions. Each block can only be assigned to at most one line up area. It is not possible to assign a block to two line up areas or more at the same time.

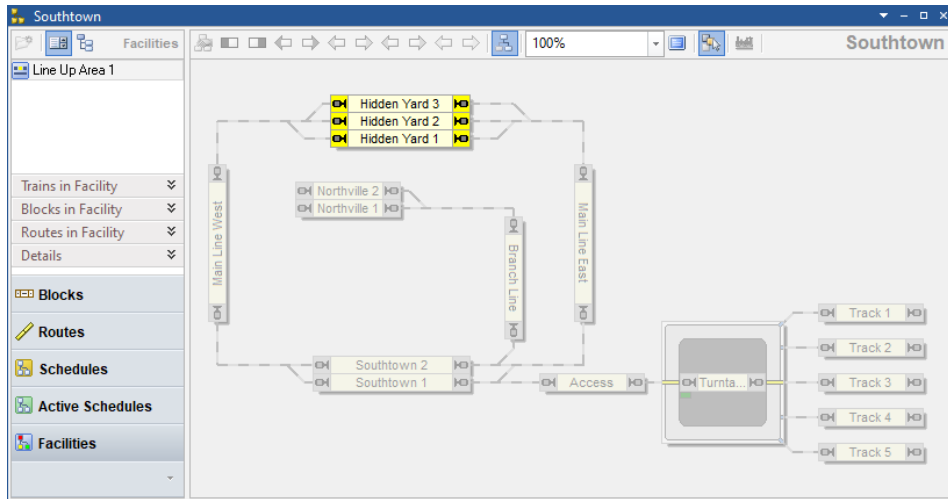


Diagram 31: Facility View with a Line Up Area in the Dispatcher Window

Diagram 31 shows a layout with a line up area. The line up area “Line Up Area 1” is currently selected and the block diagram of “Line Up Area 1” is currently visible in the right part of the dispatcher window. The blocks “Hidden Yard 1”, “Hidden Yard 2” and “Hidden Yard 3” are marked as blocks for line up to both directions (light yellow marking).

Line-Up of Vehicles and Trains in a single Block

In a line up area it is possible to line up more than one vehicle or train in a single block without connecting them to a continuous train set. This is for example useful, if you want to park several locomotives in a single track of a hidden yard.

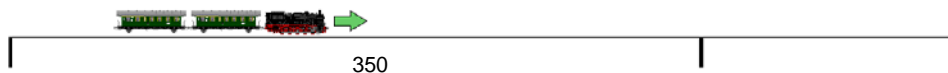


Diagram 32: Line-Up – Step 1

Assume, for example, a block in a line up area with a total space of 350 cm for the line up of trains. The first train to enter the block is brought to a stop using the usual methods so that it stops at a distance of 350 cm.

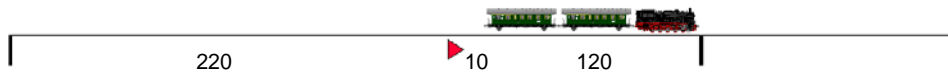


Diagram 33: Line-Up – Step 2

Assume, that the length of the first entering train is 120 cm. The gap specified for line up in the line up area is set to 10 cm. In this case the software assumes, that the waiting train is located at the right end of the block and the remaining space in the left part of the block is 220cm (350 cm minus 120 cm minus 10cm).

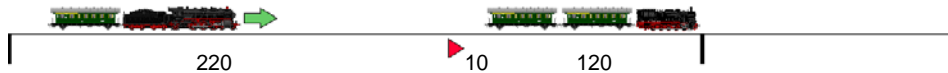


Diagram 34: Line-Up – Step 3

Now a second train enters the block from the left to the right. The incoming train will be stopped in a distance of 10 cm to the waiting train.

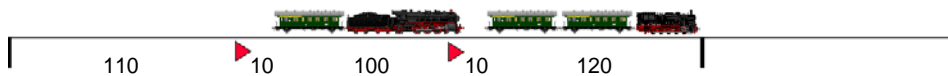


Diagram 35: Line-Up – Step 4

It is assumed, that the length of the second train is 100 cm. Now 110 cm (220 cm minus 100cm minus 10cm) are still available in the left part of the block.

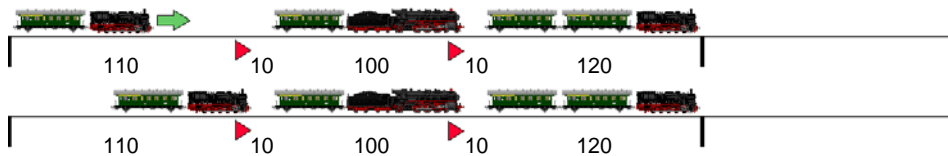


Diagram 36: Line-Up – Step 5

A third train with a length of up to 110 cm can be directed here – but also only if it enters the block from the left to the right.

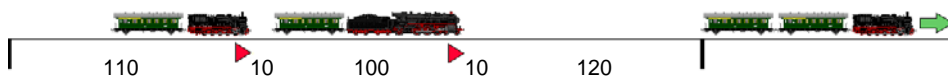


Diagram 37: First train has left the block.

When the first train (length 120 cm) leaves the block to the right later, then free space of 130 cm (120 cm plus 10 cm) becomes available in the right part of the block.



Diagram 38: Move up – Step 1

The second train in the queue will now automatically move up by 130 cm to the right.

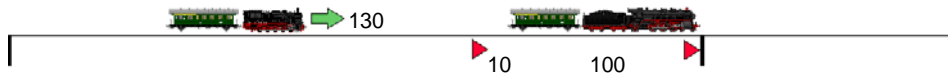


Diagram 39: Move up – Step 2

When this has been done, the third train will perform the same move.

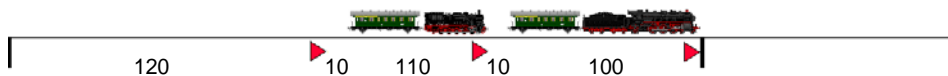


Diagram 40: Move up finished

Finally both trains are located at the right side of the block with 120cm free space in the left part of the block. Now another train with a length of up to 120 cm can be directed here - if it enters the block from the left to the right.

Line up and move up for a certain block can be performed only in one direction. This is always the direction in which the first train entered. In this example it is the direction from the left to the right.

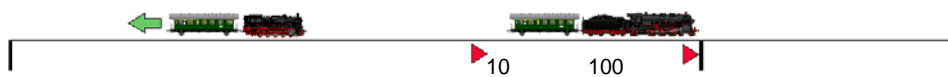


Diagram 41: Last Train leaving

However, it is also possible that the last train leaves in the direction where it came from. In this case, no move up is carried out.

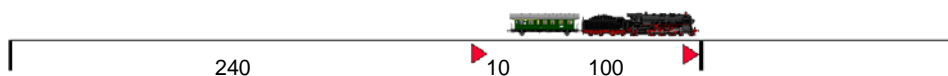


Diagram 42: Last Train has left

After this train has left, there is more space available for other trains - if they enter the block from the left to the right.

The direction of the line up can only be changed when the block has become empty again. The first train that follows can then come in from both directions.

Stopping in Line Up Areas

Stopping in line up areas works almost exactly like stopping in shunting areas. The content of the section “Stopping in Shunting Areas” (see page 53) applies to line up areas as well.

The only deviation is that there is a gap between the lined up vehicles.

A very important prerequisite for line up, which applies just as much as for shunting, should be repeated here:



An important prerequisite for a block in a line up area is that there is always a separate indicator on the side of the block where trains enter the block to be lined up. This indicator must not be kept occupied by the trains already in the block. This detector is required to report the arrival of further trains. In many cases this is the case with blocks with three occupancy sections and short sections at the ends.

Setup of Line Up Areas

The configuration of line up areas is as simple as the setup of shunting areas and almost identical to it.

The only deviation is that there is a gap between the lined up vehicles. This gap is specified in the **Rules** tab of the line up properties.

General Rules	
Gap	20
Line Up by Tracking	<input checked="" type="checkbox"/>
Speed	100
Move Up All	<input type="checkbox"/>

Gap
Specify the gap between two trains in a row.

Reset all rules to default Settings Apply to all Line Up Areas

Diagram 43: Rules of a Line Up Area



For safety reasons, the gap should be as large as possible so that trains in a row can be parked at a safe distance from each other. Don't be tempted to specify gaps that are too short in order to squeeze as many trains as possible into a block. Note that the line up and move up of trains is based on travel-time calculations that rely on the rolling stock running accurately. Certain inaccuracies must therefore be taken into account. Better to take a train less, but at a safe distance from each other.

Quite similar to shunting, stopping for line up is controlled by special markers for line up. These are set up practically in the same way as markers for shunting.

In contrast to shunting, only a single distance is specified for markers for line up, because the vehicles that are entering a block in the line up area are always lined up at the end of the waiting vehicles, i.e. in the same direction of travel in which the waiting vehicles previously entered.

Fully automatic Line Up

In order to include line up in the automatic mode, you do not have to do anything else than

- Creation of a line up area.
- Assignment and marking of the blocks belonging to the line up area.
- Establishing the special markers.
- Check that the rules for the schedules involved take line up areas into account.

Manual Line Up

If desired it is also possible to line up trains by hand or by driving a train manually into a block in a line up area with other trains already there.

Whenever a train set is composed by hand there is the choice to line up parts of the train set.

The rules of a line up area provide the option to specify, whether manually driven trains should be lined up with waiting trains in the line up area or not.

Comparison with Line Up in Version 9

In **TrainController™ Gold 9** it was possible to line up trains in the destination block of a schedule and move them up later with the help of a schedule rule.

However, this function had some weaknesses. Setting up the line up in version 9 was much more complicated than in version 10 and required, for example, the input of complicated formulas for markers. The subsequent move up could not be interrupted in version 9 and restarted again. In version 10, on the other hand, it is even possible to end the session while moving up or to interrupt it by switching to edit mode. At a later start, the move up can then be continued again. In version 9, the train that entered first had to be the first to leave again (FIFO principle); and thus necessarily always in the same direction as it entered before. In version 10, the train that entered last can also be the first to leave; then of course in the direction from which it came (LIFO principle). In contrast to version 9, this means that in version 10 trains can also be lined up in dead end tracks. In version 9, move up was only carried out at threshold speed, while in version 10 the move up can be accelerated by (moderate!) specification of higher speeds. Likewise, in version 9, only one connected part of the vehicles in a block was moved up at the same time. In version 10, if desired, all vehicles in the block can move up at the same time and thus shorten the time required for this.

The line up of version 9 is technically not compatible with the line up of version 10. Therefore, the line up method offered in version 9 is no longer supported in version 10. The corresponding schedule rule from version 9 does not exist anymore. Users who have used the line up in version 9 should switch to the new method in the course of switching to version 10 so that this works again.

6.13 Run-Round Loop

General

On real railroads, a run-round loop enables a locomotive to attach to the opposite end of the train.

With run-round loop facilities, **TrainController™ Gold 10** supports you to establish run-round loops on your model railroad layout.

A large number of options are available for the individual control of run-round loops.

Run-round loops are established by assigning blocks and routes to them. This assignment is done in a similar way like assigning blocks and routes to schedules.

Run-round loops contain

- **Holding blocks:** These are the blocks in which an entering train hold and where the cars of the train are waiting for the locomotive to be attached to the opposite end.
- **Exit blocks:** These are the blocks into which the separated locomotive moves after separation from the cars.
- **Passing blocks:** These are the blocks through which the separated locomotive passed the waiting cars.
- **Entry blocks:** These are the blocks from which locomotives enter the holding blocks in order to attach to the waiting cars at the end of a run-round.
- All routes that are required so that locomotives can move from one of the above-mentioned blocks to the next.

Each block in a run-round loop is marked by specifying a direction as exit, holding, passing or entry block. Each block can be used in any number of run-round loops.

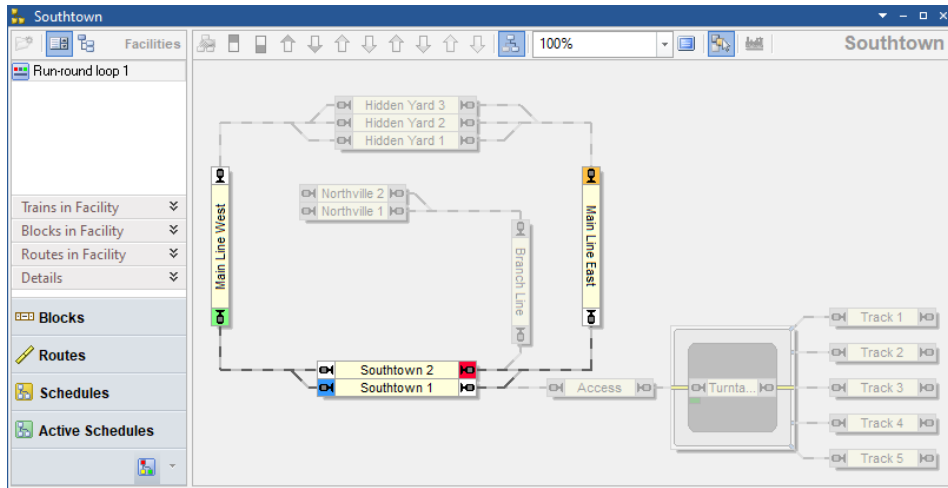


Diagram 44: Facility View with a Run-Round Loop in the Dispatcher Window

Diagram 44 shows a layout with a run-round loop. The run-round loop “Run-round loop 1” is currently selected and the block diagram of “Run-round loop 1” is currently visible in the right part of the dispatcher window. The block “Southtown 2” is marked as holding block to the right (red marking). The block “Main Line East” is marked as exit block to the top (orange marking). The block “Southtown 1” is marked as passing block to the left (blue marking). And the block “Main Line West” is marked as entry block to the bottom (green marking).

Run-Round

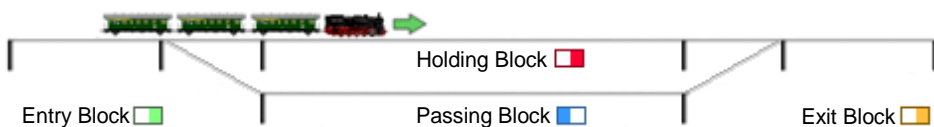


Diagram 45: Run-Round – Initial Situation 1

In the initial situation, a train with pulled cars enters the holding block.

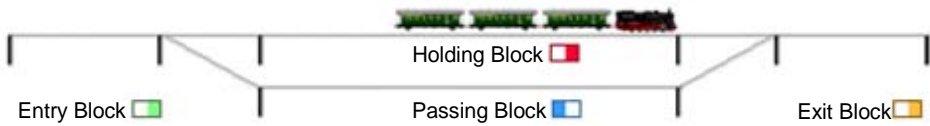


Diagram 46: Run-Round – Initial Situation 2

The train stops in the holding block.

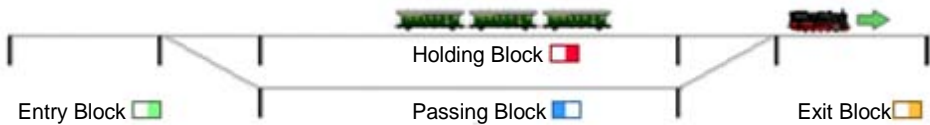


Diagram 47: Run-Round – Step 1

The locomotive is separated from the cars and moves to the exit block.

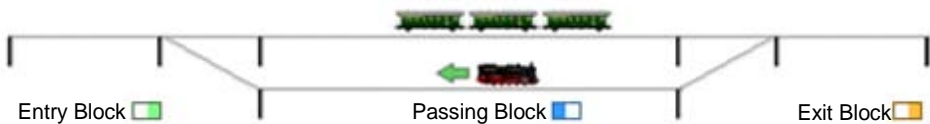


Diagram 48: Run-Round – Step 2

In the exit block, the locomotive toggles its direction of travel and moves back through the passing block...

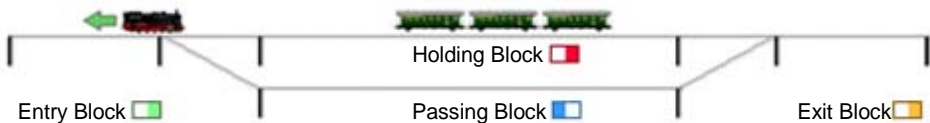


Diagram 49: Run-Round – Step 3

... to the entry block.

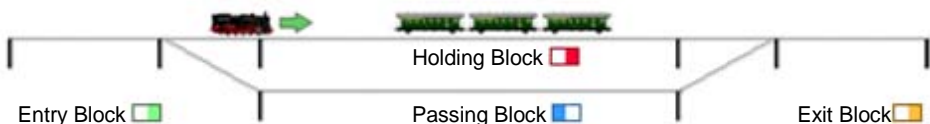


Diagram 50: Run-Round – Step 4

In the entry block it toggles its direction again and moves back to the holding block.

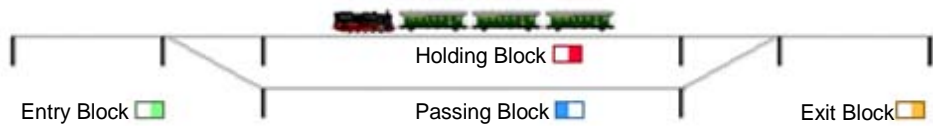


Diagram 51: Run-Round – Final Situation

Back in the holding block, the locomotive is attached to the waiting cars. The train can now leave the holding block in the opposite direction where it originally came from.

Setup of Run-Round Loops

The configuration of run-round loops is relatively simple. The software does most of the otherwise rather complex work.

The main part of the work consists in assigning the used blocks with the correct color marking in the correct direction as holding, exit, passing and entry blocks.

It is also important that the routes over which the locomotives are to travel when moving from one block to the other are also entered in the facility.

There is no need for passing blocks. If there is no passing block, the software searches for a suitable path from the exit to the entry block through the assigned blocks.

Please also note that the holding block of a run-round loop also represents a built-in shunting area. This must also be set up as described in section 6.11, “Shunting Area” and in particular as outlined on page 55 and Diagram 24.

Fully automatic Run-Round Loops

In order to integrate run-round loops into the automatic train operation, the user forms schedules which end in a holding block of the run-round loop. The run-round loop is entered as the successor to this schedule. A train that performs this schedule is automatically passed to the control of the run-round loop, when it terminates the schedule in the holding block.

On the opposite side, the user forms schedules which start in a holding block of the run-round loop. These schedules are entered as successors of the run-round loop. They are carried out after the run-round is completed according to Diagram 51.

In the example of Diagram 44, a run from the hidden yard via "Main Line West" with run round in "Southtown" and back to the hidden yard could be put together in a schedule sequence as follows:

- Schedule from the hidden yard to block "Southtown 1"
- "Run-round loop 1"
- Schedule from block " Southtown 1" to the hidden yard

The run-round maneuver would be automatically incorporated into this sequence by **TrainController™**.

Run-Round Loops and other Objects

A run-round can be performed as operation called by other objects. In this way it is for example possible to control a run-round loop with push buttons.

6.14 Engine Change

General

An engine change is a location where a locomotive is exchanged for another locomotive.

With engine change facilities, **TrainController™ Gold 10** supports you to perform the exchange of engines automatically on your model railroad layout.

A large number of options are available for the individual control of engine changes.

Engine changes are established by assigning blocks and routes to them. This assignment is done in a similar way like assigning blocks and routes to schedules.

Engine changes contain

- **Holding blocks:** These are the blocks in which an entering train hold and where the cars of the train are waiting for the new locomotive to be attached.
- **Exit blocks:** These are the blocks into which the separated locomotive moves after separation from the cars.
- **Entry blocks:** These are the blocks in which engines are provided for the change and from which these engines enter the holding blocks in order to attach to the waiting cars at the end of an engine change.
- All routes that are required so that locomotives can move from one of the above-mentioned blocks to the next.

Each block in an engine change is marked by specifying a direction as exit, holding (non directional), or entry block. Each block can be used in any number of engine changes.

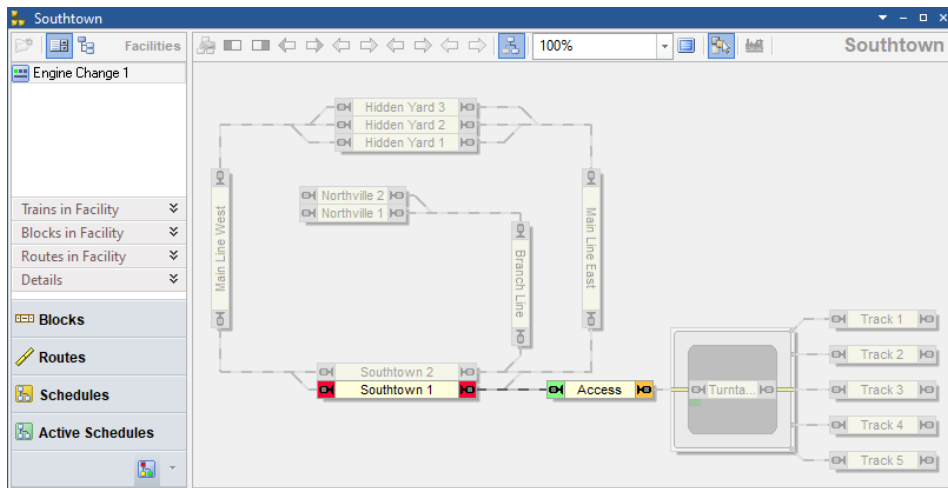


Diagram 52: Facility View with an Engine Change in the Dispatcher Window

Diagram 52 shows a layout with an engine change. The engine change “Engine Change 1” is currently selected and the block diagram of “Engine change 1” is currently visible in the right part of the dispatcher window. The block “Southtown 1” is marked as holding block. The block “Access” is marked as exit block to the right (orange marking) and as entry block to the left.

Engine Change



Diagram 53: Engine Change – Initial Situation 1

In the initial situation, a train enters the holding block. The locomotive is to be exchanged for the locomotive waiting in the entry block.



Diagram 54: Engine Change – Initial Situation 2

The train stops in the holding block.



Diagram 55: Engine Change – Exit 1

The locomotive is separated from the cars and moves to the exit block.



Diagram 56: Engine Change – Exit 2

The locomotive is stopped in the exit block. The first pass of the engine change, called “exit” is completed now.



Diagram 57: Engine Change – Entry 1

The second pass of the engine change, called “entry”, is performed now. The other locomotive, that was previously waiting in the entry block, moves to the holding block.



Diagram 58: Engine Change – Final Situation

After entering the holding block, the second locomotive is attached to the waiting cars. The train can now leave the holding block.

Setup of Engine Changes

The configuration of engine changes is relatively simple. The software does most of the otherwise rather complex work.

The main part of the work consists in assigning the used blocks with the correct color marking in the correct direction as holding, exit and entry blocks.

It is also important that the routes over which the locomotives are to travel when moving from one block to the other are also entered in the facility.

Please also note that the holding block of an engine change also represents a built-in shunting area. This must also be set up as described in section 6.11, “Shunting Area” and in particular as outlined on page 55 and Diagram 24.

Fully automatic Engine Changes

In order to integrate engine changes into the automatic train operation, the user forms schedules which end in a holding block of the engine change. The engine change is entered as the successor to this schedule. A train that performs this schedule is automatically passed to the control of the engine change, when it terminates the schedule in the holding block.

On the opposite side, the user forms schedules which start in a holding block of the engine change. These schedules are entered as successors of the engine change. They are carried out after the engine change is completed according to Diagram 58.

By default, the exit and entry of an engine change usually take place directly one after the other, as shown in the diagrams above.

However, it is also possible to call up exit and entry separately, e.g. in a schedule sequence and for example in connection with an engine shed (see section 6.18, "Engine Shed").

In the example in Diagram 52, a run from the hidden yard via "Main Line West" with an engine change in "Southtown" and back to the hidden yard could be put together in a schedule sequence as follows:

- Schedule from the hidden yard to block " Southtown 1"
- Engine Change: exit
- Engine Shed: Entry (parking the retracted locomotive)
- Engine Shed: exit (possibly with turning on a turntable)
- Engine Change: entry
- Schedule from block " Southtown 1" to the hidden yard

Since the block "Access" in Diagram 52 is used as entry and exit block, the entry can only function with a different locomotive than the exit if there are further runs between the exit and the entry, which drive the exiting locomotive out of the "Access" block and provide another locomotive in the "Access" block. This is accomplished with the help of the engine shed.

Engine Change without Exit at the Beginning

It is also possible to omit the exit at the start **of the engine change**. There is only **an** entry from an **entry** block to the holding block. This is useful, for example, if the holding block is **in a dead end** and the locomotive that has entered cannot exit before the locomotive is changed. In this case, the locomotive that has been swapped in is placed at the other end of the train right at the start. Optionally, it can be set that at the end of the entry, the locomotive that originally entered is separated from the other vehicles. At this point the **engine** change ends.

The process looks like this:

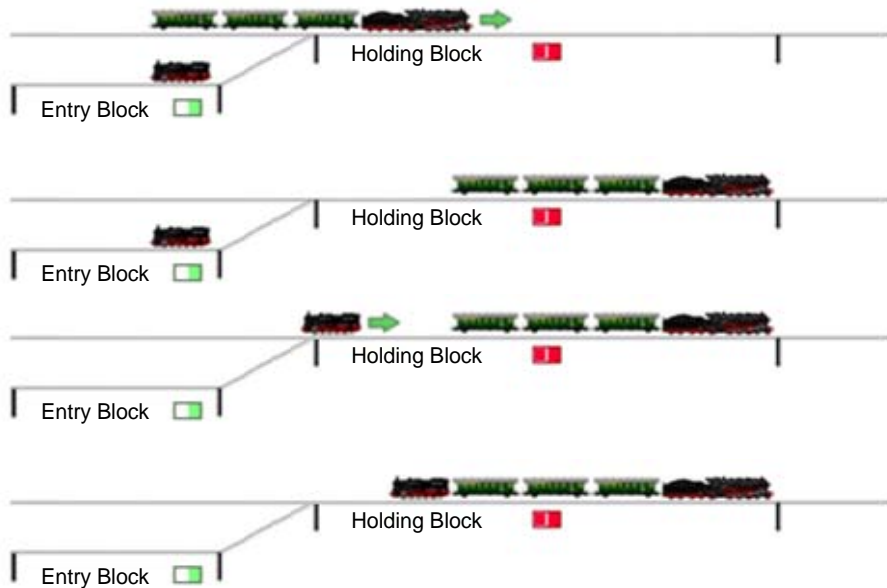


Diagram 59: Engine Change without Exit at the Beginning

The exchanged locomotive can then depart together with the other vehicles with a normal schedule in the opposite direction.

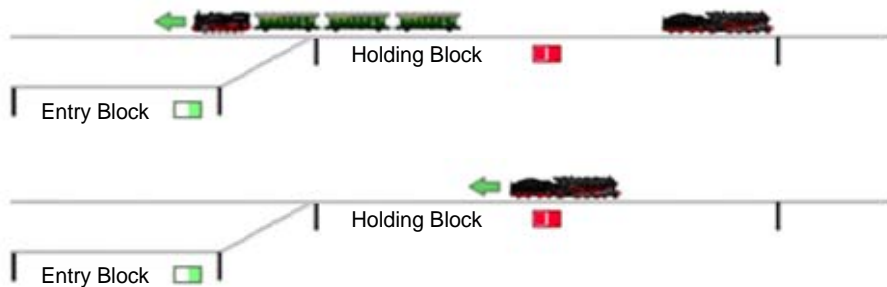


Diagram 60: After the Engine Change

The original locomotive can then finally be driven onto a suitable track with a normal schedule.

Engine Changes and other Objects

An entire engine change as well as individual entries or exits can be performed as operation called by other objects. In this way it is for example possible to control an engine change with push buttons.

6.15 Gradient

General

In gradient districts on real railroads, pusher engines are often added to trains to assist them to climb a steep grade.

With gradient facilities, **TrainController™ Gold 10** supports you to control pusher engines on your model railroad layout with the computer.

A large number of options are available for the individual control of gradients.

Gradients are established by assigning blocks and routes to them. This assignment is done in a similar way like assigning blocks and routes to schedules.

Gradients contain

- **Holding blocks:** These are the blocks at the beginning of the gradient in which entering trains hold and wait for the pushing locomotive to be added.
- **Entry blocks:** These are the blocks in which pushing engines are provided and from which these engines enter the holding blocks in order to attach to the trains to be pushed.
- **Exit blocks:** These are the blocks at the end of the gradient, where the pushing engine is separated from the pushed train and from which the pushing engine optionally returns to its origin entry block.
- All routes that are required so that trains can move from one of the above-mentioned blocks to the next.

Each block in a gradient is marked by specifying a direction as entry, holding (non directional), or exit block. Each block can be used in any number of engine changes.

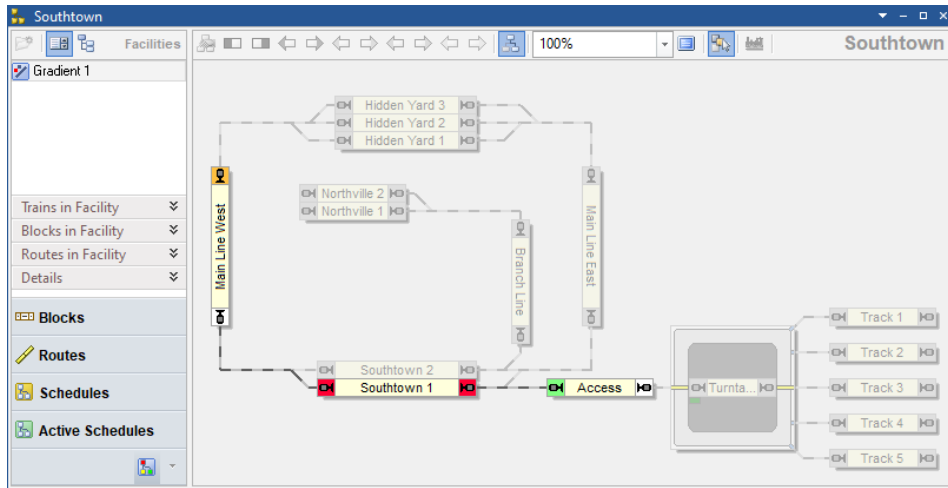


Diagram 61: Facility View with a Gradient in the Dispatcher Window

Diagram 61 shows a layout with a gradient facility. The gradient “Gradient 1” is currently selected and the block diagram of “Gradient 1” is currently visible in the right part of the dispatcher window. The block “Southtown 1” is marked as holding block. The block “Access” is marked as entry block to the left (green marking) and the block “Main Line West” is marked as exit block to the top (orange marking).

Operation of a Gradient

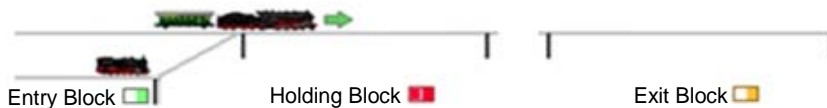


Diagram 62: Gradient – Initial Situation 1

In the initial situation, a train enters the holding block. The pushing locomotive is provided in the entry block.



Diagram 63: Gradient – Entry 1

The train stops in the holding block. The pushing locomotive leaves the entry block and approaches the train in the holding block.



Diagram 64: Gradient – Entry 2

The pushing locomotive touches the waiting train.



Diagram 65: Gradient – Exit 1

The entire train set leaves the holding block and climbs the gradient towards the exit block.

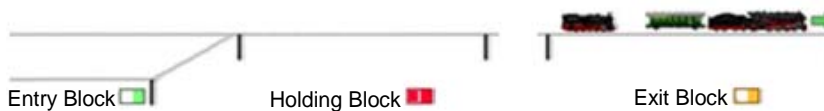


Diagram 66: Engine Change – Exit 2

After entering the exit block, the pushing locomotive is separated from the pushed train. The pushed train proceeds, the pushing locomotive is stopped.

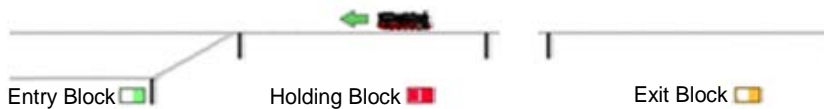


Diagram 67: Engine Change – Return

After the pushed train has left the exit block, the pushing locomotive returns to the entry block.



Diagram 68: Engine Change – Final Situation

Back in the entry block, the pushing engine is ready to assist the next train on the gradient.

Setup of Gradients

The configuration of gradients is relatively simple. The software does most of the otherwise rather complex work.

The main part of the work consists in assigning the used blocks with the correct color marking in the correct direction as holding, entry and exit blocks.

It is also important that the routes over which the locomotives are to travel when moving from one block to the other are also entered in the facility.

Please also note that the holding block of a gradient also represents a built-in shunting area. This must also be set up as described in section 6.11, “Shunting Area” and in particular as outlined on page 55 and Diagram 24.

Likewise, each exit block of a gradient needs a special yellow marker in the exit direction, if the pushing locomotive is to be separated in the exit block from the pushed train while it is in motion. This special marker indicates the point at which the pushing locomotive is stopped after separation from the pushed train. This special marker is set up in a very similar way to the special markers for shunting or line up (similar to Diagram 24). However, the specified distance does not determine the stopping point of a locomotive driving in and joining waiting vehicles, but the stopping point of a locomotive after separation from a running train.

Fully automatic Gradients

In order to integrate gradients into the automatic train operation, the user forms schedules which end in a holding block of the engine change. The gradient is entered as the successor to this schedule. A train that performs this schedule is automatically passed to the control of the gradient, when it terminates the schedule in the holding block.

On the opposite side, the user forms schedules which start in an exit block of the engine change. These schedules are entered as successors of the gradient. They are carried out after the exit of the gradient is completed according to Diagram 66.

Note, that the sequence of schedules only contains the schedules, which take the trains to the holding block, the gradient itself, which takes the train along with the pushing engine from the holding block to the exit block and the schedule, which takes over the train in the exit block.

The move of the pushing engine from the entry block to the back of the train in the holding block is automatically inserted into this sequence of schedules by the software; as well as the return of the pushing engine back to the entry block. You do not have to take care of these additional moves.

In the example of Diagram 61, a run from the hidden yard via "Main Line East" with gradient between "Southtown" and "Main Line West" and further to the hidden yard could be put together in a schedule sequence as follows:

- Schedule from the hidden yard to block "Southtown 1"
- "Gradient 1"
- Schedule from block " Main Line West" to the hidden yard

The pushing maneuver would be automatically incorporated into this sequence by **TrainController™**.

By default, the return of the pushing engine back to the entry block is performed automatically as shown in the diagrams above. If desired, this return can also be disabled. In this case the pushing engine is stopped in the exit block and remains there. From there it can be moved back or to another location by the regular means of schedules.

In the diagrams above the pushing engine is separated while the train is still running and the pushed train is not stopped in the exit block. This is not the default, but was demonstrated here to show, what can be done. By default, the train stops in the exit block and the pushing engine is separated while the train is standing still.

Gradients and other Objects

The entire operation of a gradient can be performed as operation called by other objects. In this way it is for example possible to control a gradient with push buttons.

6.16 Hump Yard

General

On real railroads, a hump yard is a railway yard found at some freight train stations, used to separate railway cars onto one of several tracks. First the cars are taken to a track on an artificially built hill called a hump. From there the cars are sent by utilizing the force of gravity through a series of turnouts onto the classification tracks.

With hump yard facilities, **TrainController™ Gold 10** supports you to establish a hump yard on your model railroad layout.

A large number of options are available for the individual control of hump yards.

Hump Yards are established by assigning blocks and routes to them. This assignment is done in a similar way like assigning blocks and routes to schedules.

Hump Yards contain

- **Entry blocks:** These are the blocks through which trains enter the hump yard and in which the cars are separated.
- **Exit blocks:** These are the blocks into which the separated cars roll down in order to form new train sets.
- All routes that are required so that the separated cars can move from one of the above-mentioned blocks to the next.

Each block in a hump yard is marked by specifying a direction as entry or exit block. Each entry block can only be assigned to at most one hump yard. It is not possible to assign a block as entry block to two hump yards or more at the same time.

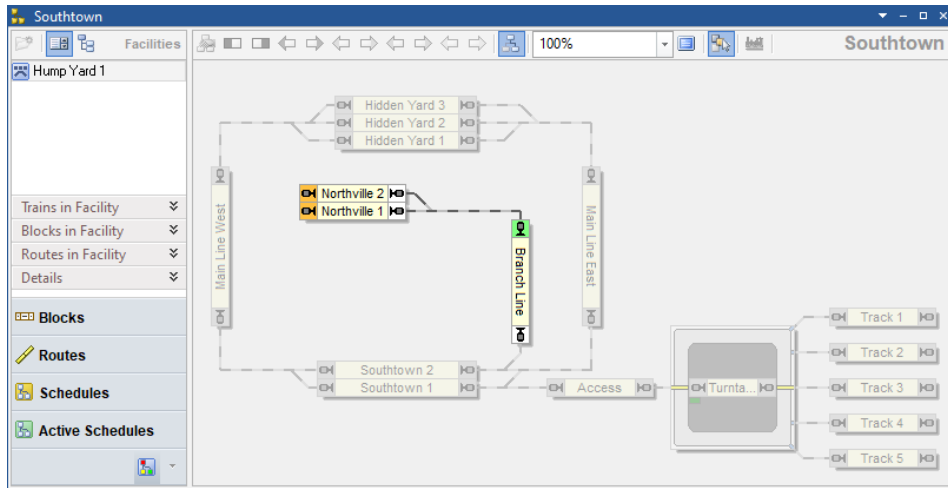


Diagram 69: Facility View with a Hump Yard in the Dispatcher Window

Diagram 76 shows a layout with a hump yard. The hump yard “Hump Yard 1” is currently selected and the block diagram of “Hump Yard 1” is currently visible in the right part of the dispatcher window. The block “Branch Line” is marked as entry block to the top (green marking) and the blocks “Northville 1” and “Northville 2” are marked as exit blocks to the left (orange marking).

Move Up, Separation and Roll Down



Diagram 70: Hump Yard – Initial Situation

In the initial situation, a train with pushed cars drove into the entry block of the hump yard. With its tip it is at the position where the cars are to be separated later (separation point). If, for example, a decoupling track is used, the separation point is determined by the position of the coupling plank. If couplings are used on the vehicles, the separation point is selected appropriately. It is important that the separation point is chosen in such a way that, after separation, the cars can roll by themselves over a suitable slope to the exit blocks.

The following process only works if the train has entered the entrance block with the wagon pushed.



Diagram 71: Hump Yard – Move Up

The process begins with the move up.

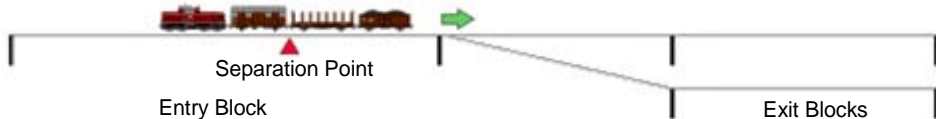


Diagram 72: Hump Yard – Move Up

The move up ends when the gap between the two vehicles to be separated has reached the separation point.

There are several options to determine, when this happens:

- In automatic mode by calculating the distance using the length of the vehicles to be separated.
- In automatic mode by a stop marker that is triggered by a suitable feedback indicator (e.g. an optical sensor at the uncoupler).
- In the case of trains driven by hand, by the user stopping the train at the appropriate point.

A suitable rule in the properties of a hump yard can be used to specify which of the two options specified for automatic mode (length calculation vs. stop marker with sensor at the separation point) the user would like to use.

It is of course also possible that the train is already standing as shown in Diagram 72 directly after entering. In this case, the first move up is not necessary. The user can set this with a program option so that the program knows how the train has stopped.

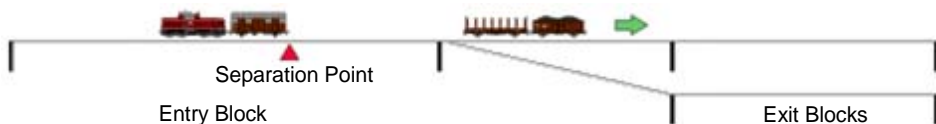


Diagram 73: Hump Yard – Roll Down

After the move up has ended, the corresponding vehicles are separated from the rest of the train. These vehicles now roll autonomously down the hump yard into an exit block.



Diagram 74: Hump Yard – Next Move Up

After the cars arrive in an exit block, the sequence begins again with the move up of the remaining train and the separation of the remaining cars.

The next move up can optionally also begin while the already separated cars are still rolling and the path to a (different) exit block is not yet available (instant move up). For this purpose, there must be enough space between the separation point and the end of the entry block so that cars can advance into this area if the further path behind the entry block has not yet been cleared by the rolling cars.



Diagram 75: Hump Yard – Final Situation

The whole process ends when all cars have been separated from the pushing locomotive and have arrived in exit blocks.

Additional options and notes:

- Operations can be initiated in the moment, when cars are separated. These operations can be used, among others, to control an uncoupling track or couplings on the vehicles.
- As already written, the train can already move up while rolling vehicles block the further path. It is even possible that the separation is carried out without the pushing train being stopped. Of course, the train can also be stopped for separation if the cars can only be separated when the train is at a standstill.
- The rolling down of the cars is carried out as a normal schedule including activation of routes between the entry and exit block. Since cars do not have their own motor and therefore their speed cannot be controlled, a speed that can be set by the user is used as a basis for display purposes.
- The arrival of the rolling cars in an exit block and thus the end of this schedule is reported either by the stop marker in the exit block or by the expiry of a period of time specified by the user if such a marker is not available.
- The user must ensure that rolling cars are physically slowed down, stopped in the exit block and properly coupled to vehicles that are already waiting there. The software has no way of influencing the rolling cars. It just lets them roll.
- The process can run completely automatically. However, it is also possible to trigger the individual steps and events (move up, separation and ending the roll down) by hand (e.g. by operations of a push button).
- The moving train can be controlled either by the computer or by the user. Reaching the right position for separation can be determined in various ways. This has already been explained above.
- The program determines the number of separated cars at random. For this purpose, however, the user can define a maximum number of cars to be separated. Alternatively, the user can determine the separation himself if he logically separates the train set with the usual means between the desired cars before moving up.
- For the selection of the exit blocks into which the cars roll, the same possibilities are available as for the selection of destination blocks in schedules. For example, cars will not roll into blocks that are locked with entry locks or conditions that are not met. It is of course also possible to combine the hump yard with a branching (see chapter 6.5, “Branching”) in order to control the selection of the destination blocks for the rolling cars.

Fully automatic Hump Yards

In order to integrate hump yards into the automatic train operation, the user forms schedules which end in an entry block of the hump yard. The hump yard is entered as the successor to this schedule. A train that performs this schedule is automatically passed to the control of the hump yard, when it terminates the schedule in the entry block.

On the opposite side, the user forms schedules which start in an exit block of the hump yard. These schedules are entered as successors of the hump yard. They are carried out when all cars have been separated from a train in the hump yard and have reached their exit block in the hump yard.

Hump Yards and other Objects

Move up, separation, ending the roll down or termination of all active runs in a hump yard can be performed as operations called by other objects. In this way it is for example possible to control all activities in a hump yard with push buttons.

6.17 Hidden Yard

General

A large number of options are available for the individual control of hidden yards.

Hidden Yards are established by assigning blocks and routes to them. This assignment is done in a similar way like assigning blocks and routes to schedules.

Hidden Yards contain

- **Holding blocks:** These are the blocks in which trains are parked in the hidden yard.
- **Passing blocks:** These are the blocks through which trains pass through the hidden yard when they are not parked there.
- **Entry blocks:** These are the blocks through which trains enter the hidden yard.
- **Exit blocks:** These are the blocks through which trains exit the hidden yard.
- **Control blocks:** These are blocks in which trains can temporarily stop to proceed to a holding block when becomes available later or can be parked in a dead end track, when the holding block in this track is already used by another train. Control blocks can be used to park trains one after the other in the hidden yard (see also section “Parking Trains in a Row” below).
- All routes that are required so that trains can move from one of the above-mentioned blocks to the next.
- A hidden yard can also contain other blocks that are not specially marked. These blocks are then passed without stopping on runs in the hidden yard.

Each block in a hidden yard is marked by specifying a direction as one of the block types listed above. Each block can only be assigned to at most one hidden yard. It is not possible to assign a block to two hidden yards or more at the same time. It is also not possible to assign the same block to a hidden yard and to a station.

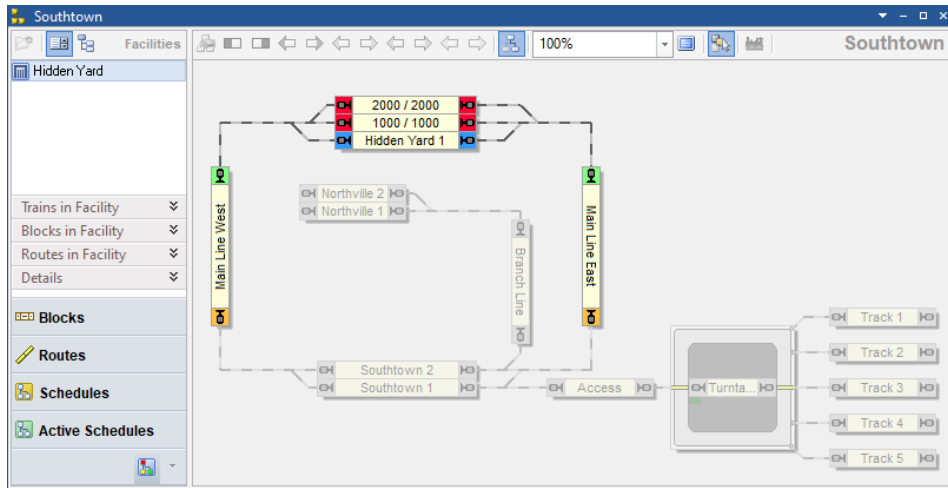


Diagram 76: Facility View with Hidden Yard in the Dispatcher Window

Diagram 76 shows a layout with a hidden yard facility. The hidden yard “HiddenYard” is currently selected and the block diagram of “Hidden Yard” is currently visible in the right part of the dispatcher window. The blocks “Main Line West” and “Main Line East” are marked as entry blocks to the top (green marking) and exit blocks to the bottom (orange marking), the blocks “Hidden Yard 2” and “Hidden Yard 3” are assigned to the hidden yard as holding blocks to both directions (red marking). The block “Hidden Yard 1” is marked as passing block to both directions (blue marking).

Very often only the blocks used for parking are called a hidden yard. But a hidden yard in the sense of **TrainController™** does not only include these blocks, but also those blocks outside the parking area that are relevant for the control of the hidden yard. This means in particular that the entry and exit blocks are often outside of what the user calls a hidden yard; these blocks can also be in the visible area on the open track.

The entry and exit blocks are less the first or last blocks in the hidden yard, but more the blocks from which the hidden yard is entered or into which the hidden yard is left.

Blocks in which trains are to be parked are always holding blocks (red) or control blocks (light yellow) in the relevant direction of travel - never entry and exit blocks (green or orange). However, this only applies to each direction of travel. One and the same block can theoretically be green (entry block) in one direction and red (holding block) in the other.

Hidden yards are actually composite objects. In a hidden yard, the functionality and most properties of a station, an exit and several schedules (for entry, exit and passing) are also included.

Entry

Entries lead from an entry block to a holding block of the hidden yard. They are largely carried out like normal schedules. However, there are a few special features and additional options.

Entries can be performed by using a menu command, as an operation by other objects (e.g. buttons or macros), as successors of schedules or spontaneous runs as well as part of AutoTrain runs.

In the event that the hidden yard is full, i.e. no holding block is available, when a train enters the hidden yard, the following can be specified as a reaction:

- **Pass:** Passing through the hidden yard without stopping.
- **Wait:** Wait in the entry block until a holding block becomes free.
- **No Entry:** Do not perform an entry. In this case, the entry fails and will have to be tried again later or replaced by another action for the train.

Another option can cause a train, that is currently occupying one of the most suitable holding blocks, should first exit when the hidden yard is full, in order to automatically create space for the incoming train.

It can also be specified that a holding block should always be kept free. With this option, no fixed passing blocks need to be defined. Passing through the hidden yard is then always carried out through the still free holding block.

Furthermore, almost all the rules that are also offered for normal schedules are available for the entry.

Exit

An exit leads from a holding block to an exit block of the hidden yard. Exits are largely carried out like normal schedules. However, there are a few special features and additional options.

Exits can be performed by using a menu command, as an operation by other objects (e.g. buttons or macros), as successors of schedules, as predecessors of spontaneous runs as well as part of AutoTrain runs.

The selection of the holding block from which the departing train is started can be influenced using the same criteria as described on page 38 for the exit facility.

Furthermore, almost all the rules that are also offered for normal schedules are available for the exit.

Optimizing the use of the Hidden Yard

The selection of the holding block as the destination of an entry is controlled by the following criteria:

Diagram 77: Settings for the Selection of Start and Destination Blocks in a Hidden Yard

- **Obstacle:** The holding with the minor obstacles (e.g. entry or exit locks, other trains, unfulfilled conditions, etc.) on its access path is chosen. All obstacles between the entry block and the holding block are evaluated. This criterion largely corresponds to the normal selection of paths in normal schedules.
- **Priority of Block:** The holding block with the highest priority is chosen. For this criterion, a number can be entered as a priority for the entry for each holding block of a hidden yard.

It is optionally possible to use the priority as quantifier for the selection. The probability results from the ratio of the priorities of the holding blocks. If one block has priority 3000 and another has priority 1000, for example, the first block is selected with a probability of 3 to 1.

- **Position:** This criterion is effective if the hidden yard has control blocks that can be used to park trains one after the other (see also section “Parking Trains in a Row” below). A suitable holding block (block in the first row) is preferably selected for this criterion. If no holding block is available, an control block is selected as an alternative, which is directly adjacent to a holding block (block in the second row). If such block is not available, an control block in the third row is chosen, etc.
- **Reservation:** A holding block, which is already reserved for the entering train is chosen.
- **Origin Block:** The holding block is chosen, where the train started from at the previous exit from the hidden yard and which has been kept free for this train. This criterion is useful to let trains return to the same block in the hidden yard, where they exited from.
- **Count:** It is possible to specify for certain blocks in the hidden yard that the end of long trains may extend into them when they are parked. With this criterion, holding blocks are preferred in which the parked train occupies the smallest possible number of blocks.
- **Length:** The shortest holding block into which the train fits is selected. If parking in more than one block is possible, the appropriate blocks are selected, which are the shortest overall. With this criterion, the available track length in the holding blocks of the hidden yard can be optimally used.
- **Max Distance:** The holding block furthest from the entry block is selected. This criterion is useful when the hidden yard has several parking areas one behind the other. This criterion can then be used to ensure that the train advances as far as possible in the hidden yard so that the exit is as short as possible.
- **Min Distance:** The holding block closest to the entry block is selected.
- **Level in Cabinet:** If there are holding blocks in a cabinet (see section 6.19), a holding block is selected in the level that is closest to the current position of the cabinet.

The user can specify the criteria in a list according to their importance. The first criterion is the most important. If no decision can be made based on this criterion, the second criterion is used. If this does not allow a decision, the third criterion in the list applies. And so on.

With some of the criteria described above, the use of the hidden yard can be optimized according to your own needs.

By using **Priority of Block**, certain blocks are always preferred over other blocks at the entry.

If long trains are allowed to occupy several blocks when parked, **Count** makes it possible to park as many trains as possible. However, this can mean that many short trains are parked and there is no more space for long trains that occupy more than one block when parked.

Length can ensure that the track space available in the holding blocks is used optimally. However, this can result in a long train occupying three shorter blocks rather than a long one if these three blocks are shorter than the long one.

If trains can be parked one after the other, **Position** means that trains are preferably parked in the first row so that they can immediately leave again unhindered. If no holding block is available in the first row, this criterion means that as few trains as possible are in front of them after they have been parked.

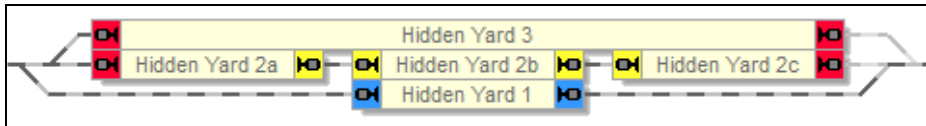


Diagram 78: Optimizing the use of a Hidden Yard

Diagram 78 shows an example. Let's assume, a short train is parked in "Hidden Yard 2c". If **Position** is the selection criterion with the highest priority, then another short train entering the hidden yard will be directed to "Hidden Yard 3". This allows both trains to exit quickly. "Hidden Yard 3", however, is no longer available for a long train.

This can be avoided by setting **Length** as the criterion with the highest priority. In this case the second short train will be directed to "Hidden Yard 2b", leaving "Hidden Yard 3" available for a long train. **Length** however, may cause an entering long train to go to "Hidden Yard 2c" and to occupy "Hidden Yard 2a" and "Hidden Yard 2b", too, if the hidden yard is empty in this moment. This will leave only one block, "Hidden Yard 3", available for further trains.

This can be avoided by setting **Count** as the criterion with the highest priority. The long train will then be directed to "Hidden Yard 3", when the hidden yard is empty. However, an even longer train, that does not fit into "Hidden Yard 2 a,b,c", but fits into "Hidden Yard 3", cannot be parked anymore.

As you can see, there is no one strategy that can optimally solve all possible cases. But by arranging the criteria in a list according to their importance (see Diagram 77), the user can define that strategy for the optimal use of the hidden yard, that fits best his personal needs.

Parking Trains in a Row

By marking blocks in a hidden yard as control blocks, it is possible to park several trains one after another in separate blocks.

A train is parked in an control block when it enters the hidden yard and cannot advance to a real holding block. As soon as a block is released in front of the train, it moves up automatically until it reaches a real holding block. The entry is then ended there.

Optionally, this also works if the session is interrupted or terminated and restarted in between.

The **Position** option available for the entry (see section “Optimizing the use of the Hidden Yard”) can be used to control whether trains prefer go to real holding blocks before they make a stopover in a second, third or further row in an control block.

Line Up

It is not only possible to park several trains one after another in separate blocks of a hidden yard by using holding and control blocks. It is also possible to park several trains one after another in a single holding or control block.

The holding and control blocks of a hidden yard always form a line up area as described in section 6.12, “Line Up”. In a certain holding block, line up can be activated by defining markers for line up in the section settings of a block in a hidden yard (see also page 62).

Parking a Train in more than one Block

If necessary, long trains can also be parked so that they occupy more than one block. For this, a corresponding option can be set in the settings of each block for a hidden yard.

The **Count** option for the entry (see section “Optimizing the use of the Hidden Yard”) can be used to control whether long trains prefer to go to sufficiently long holding blocks before they occupy several shorter blocks when they are parked.

Holding Blocks for specific Trains

The trains that may be parked there can be specified for each holding block or control block. If no trains are specified for a block, all trains can be parked there.

If a train is not allowed to be parked in a holding block, it cannot be parked in any preceding control block. Because that assumes that moving up to the following holding block is allowed. This does not apply to holding blocks in a dead end track.

If a train is not allowed to be parked in an control block, it only moves in the direction of this block, if it can either be parked in a preceding control block or in a subsequent holding block.

Origin Block

When a train enters a hidden yard, it is possible to direct this train to a holding block that has been kept free for it. At the exit, it is possible that the holding block from which the train starts is kept free for this train. With the **Origin Block** criterion for selection of the holding block (see above), it is then possible to direct the train to this block at the next entry. If this criterion is placed at the top of the selection list (see Diagram 77), trains always return to the block from which they left the hidden yard. By simply moving a train from one block to another, it is also possible to change the block, where a train returns to, at any time.

Engine Functions

6.18 By suitable combination of a hidden yard with a tunnel (see 6.4, “Tunnel”) it is possible to specify that all the engine functions that are currently switched on are automatically switched off when a train enters the hidden yard or tunnel, respectively. These functions are later turned on again, when the train leaves the hidden yard or tunnel.

Full automatic Hidden Yards

In order to integrate hidden yards into the automatic train operation, the user forms schedules which end in an entry block of the hidden yard. The entry into the hidden yard is entered as the successor to this schedule. A train that performs this schedule is automatically passed to the control of the hidden yard, when it terminates the schedule in the entry block. The control of the hidden yard then leads the train to a suitable holding block.

On the opposite side, the user forms schedules which start in an exit block of the hidden yard. These schedules are entered as successors of the hidden yard. A train that performs an exit is automatically put under control of a successor schedule, when it terminates the exit in the exit block.

Operations can be performed automatically at the beginning or end of an entry or exit. These operations work just like the operations that can be set for the start or end of normal schedules.

It can also be specified that after a train arrives in the hidden yard, another train waiting there will automatically exit. It can also be specified that the two trains must be "similar". A train description is given for this. Two trains are similar if there is at least one line in the train description that fits both trains (see also page 109). In this way e.g. the arrival of a passenger train can cause another passenger train to automatically exit the hidden yard; or an arriving freight train can cause the exit of another freight train; etc.

Hidden yards can be automatically emptied at the start of a session. For this purpose, as many exits are automatically carried out at the start of a session until there is only a predetermined number of trains in the hidden yard. This process can be called not only when starting a session, but also by menu command or operation by other objects.

Hidden Yards and AutoTrain

Hidden yards can be combined with AutoTrains runs, too.

If an AutoTrain run is defined that ends in a hidden yard, a series of schedules is created. This sequence consists of an AutoTrain run from the previously defined start block (or start blocks) to a suitable entry block of the hidden yard followed by an entry into the hidden yard.

If an AutoTrain run is defined, that starts in a hidden yard, a series of schedules is also created. This sequence consists of an exit from the hidden yard to a suitable exit block, followed by an AutoTrain run, that starts in this exit block and ends in the previously defined destination block (or blocks).

It is also possible to define an AutoTrain run between two hidden yards, or an AutoTrain run that starts and ends in the same hidden yard.

In this case, a series of schedules is also created, which combines the two cases described above. So this series consists of an exit from the start yard followed by an AutoTrain run from an exit block of the start yard to an entry block of the destination yard and finally followed by an entry to the destination yard.

It is also possible to define an AutoTrain run that begins and ends outside a hidden yard and only passes through a hidden yard. Similar to the cases above, a series of individual schedules is generated here.

Hidden Yards and Spontaneous Runs

Hidden yards can be combined with spontaneous runs, too.

For this purpose, it can be determined whether trains will automatically perform an entry to the hidden yard when they have reached the entry block of the hidden yard under the control of a spontaneous run.

For the opposite side, it can be set that trains that were under the control of a spontaneous run before entering the hidden yard are put under control of a spontaneous run again after performing an exit from the hidden yard.

Hidden Yards as Station

Every hidden yard is also a station at the same time. The same options offered for stations can also be set for hidden yards. Hidden yards also support the same functions as stations. Further information can be found in the explanations for stations.

Hidden Yards and other Objects

Entries and exits of hidden yards can be performed as operations called by other objects.

The states of a hidden yard can be evaluated in the trigger or condition of other objects. It can be evaluated separately from one another, whether entries, exits, passes or any of them are currently being carried out or not.

In this way it can be determined whether or not activities are taking place in a hidden yard.

6.18 Engine Shed

General

A large number of options are available for the individual control of engine sheds.

Since locomotives are parked in engine sheds in a similar way like entire trains in hidden yards, engine sheds and hidden yards have a lot in common.

Most of the features described in section 6.17, “Hidden Yard”, are also available for engine sheds. However, there are a few variations and additional options that are described below.

The use of engine sheds is particularly recommended where the locomotives enter the shed via a turntable. This is where the full performance of this type of facility comes into play.

Engine sheds are established by assigning blocks and routes to them. This assignment is done in a similar way like assigning blocks and routes to schedules.

Engine sheds contain holding, entry and exit blocks, but no passing or control blocks.

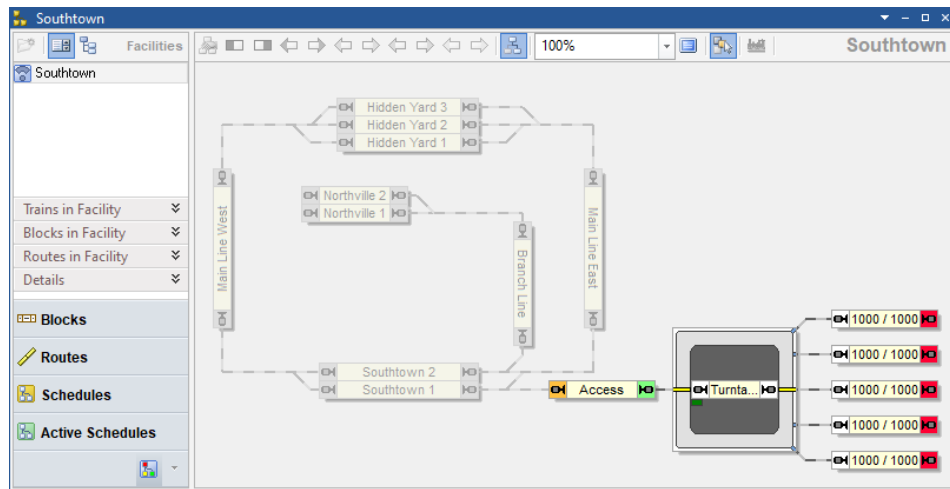


Diagram 79: Facility View with an Engine Shed in the Dispatcher Window

Diagram 79 shows a layout with an engine shed facility. The engine shed “Southtown” is currently selected and the block diagram of “Southtown” is currently visible in the right part of the dispatcher window. The block left of the turntable is marked as entry block to the right (green marking) and exit block to the left (orange marking), the blocks right of the turntable are assigned to the engine shed as holding blocks to the right (red marking).

Entries and exits of engine sheds work in a similar way like entries and exits of hidden yards. They are called by menus, operations, etc. and can be combined with schedules, AutoTrain or spontaneous runs like entries or exits of hidden yards.

The selection of a suitable holding block can be controlled with the same criteria as explained in the section “Optimizing the use of the Hidden Yard”.

Engine Sheds and Turntables

Engine sheds develop their full potential in connection with turntables. Specific features are supported, if the engine shed contains a block, which is associated with a turntable.

In this case it is possible to specify, whether engines will be parked in the engine shed with forward or backward orientation. The same is possible for exits. I.e. it is possible to specify, whether an engine exits the engine shed running forward or backward. Train descriptions can be added which describe, which engines are affected (see Diagram 80).

The screenshot shows a software window titled "Engine Shed - Southtown". It has a menu bar with "General", "Rules", "Successors", "Trains", "Train Priority", and "Condition". The "General" tab is active. Below the menu bar, there are buttons for "OK", "Cancel", and "Help". The main area is divided into sections: "General Properties:" with a "Name:" field containing "Southtown"; "Entry / Exit Selection:" with two lists. The "Entry:" list contains "Reservation", "Origin Block", "Obstacle", "Position", "Count", "Priority of Block", and "Length", with a "Quantifier" checkbox below it. The "Exit:" list contains "Priority of Block", "Priority of Train", "Oldest", "Length", and "Count", with a "Quantifier" checkbox and a "Range:" field set to "1" below it. Between the lists are up/down arrows and a close button. The "Orientation:" section has checkboxes for "Entry Forward", "Entry Backward", "Exit Forward", and "Exit Backward", each with an "Engines..." button next to it.

Diagram 80: General properties of an Engine Shed

Engine sheds support special functions that can be called via menu or as an operation from other objects:

- **Track Change:** With this function, a locomotive standing in an entry block of the engine shed is driven onto the turntable and out again via another track to an exit block of the engine shed. The engine changes its direction of travel. That means, if the engine drives forwards on the turntable, it drives backwards back. This function can be used, for example, to move an engine to an adjacent track without changing its orientation.
- **Turn:** With this function, a locomotive standing in an entry block of the engine shed is driven onto the turntable, performs a 180 ° turn and drives out again over the same track. For this, the entry block must also be marked as an exit block in the opposite direction. This function can be used to turn a locomotive around.
- **360° Turn:** With this function, a locomotive standing in a holding block of the engine shed is driven onto the turntable. Here it makes two 180° turns and then returns to the holding block. In this way a locomotive can be presented on a turntable, like during open days on real railroads.

6.19 Display Cabinet - Train Lift – Fiddle Yard

General

Cabinets can be arranged as a specific type of facility to easily include train display cases, fiddle yards, paternosters or locomotive lifts in the control of your model railroad.

Cabinets are established by assigning blocks and routes to them. This assignment is done in a similar way like assigning blocks and routes to schedules.

Cabinets contain

- **Holding blocks:** These are the blocks in which trains are parked in the cabinet.
- **Entry/Exit blocks:** These are the blocks from which trains enter the cabinet or to which trains leave the cabinet.
- All routes that are required so that trains can move from one of the above-mentioned blocks to the next.

Each block in a cabinet is marked as one of the block types listed above. Each block can only be assigned as holding block to at most one cabinet. It is not possible to assign a block as holding block to two cabinets or more at the same time.

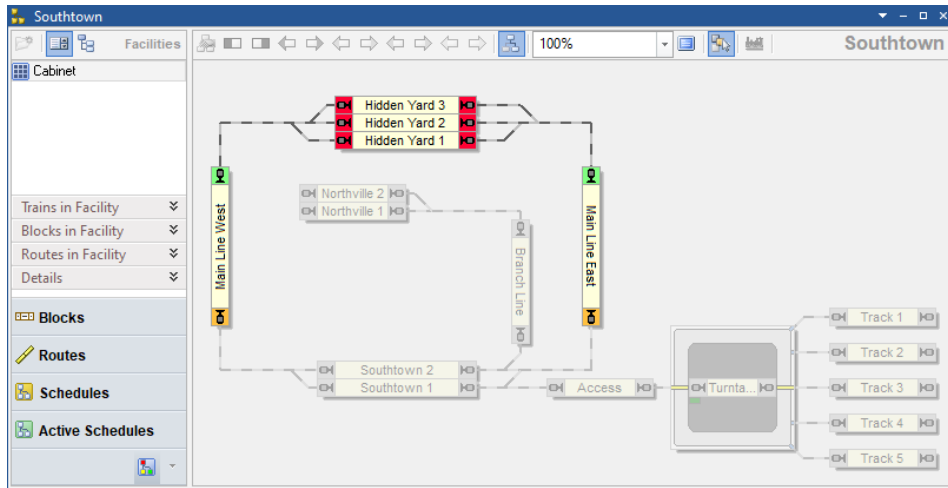


Diagram 81: Facility View with Cabinet in the Dispatcher Window

Diagram 76 shows a layout with a cabinet facility. The cabinet “Cabinet” is currently selected and the block diagram of “Cabinet” is currently visible in the right part of the dispatcher window. The blocks “Main Line West” and “Main Line East” are marked as entry blocks to the top (green marking) and exit blocks to the bottom (orange marking), the blocks “Hidden Yard 1”, “Hidden Yard 2” and “Hidden Yard 3” are assigned to the cabinet as holding blocks (red marking). It is important to note that the routes between the holding blocks and the entry/exit blocks must also be entered in the cabinet.

Very often a display case or train lift contains only the blocks used for parking. But a cabinet facility in the sense of **TrainController™** does not only include these blocks, but also those blocks and routes outside the parking area that are relevant for the control of the cabinet. This means in particular that the entry/exit blocks are often outside of what the user calls a cabinet.

The entry/exit blocks are less the first or last blocks in the cabinet, but more the blocks from which the cabinet is entered or into which the cabinet is left.

Blocks in which trains are to be parked are always holding blocks (red) - never entry/exit blocks (green/orange).

Basic Principle

A cabinet always consists of several levels in which several blocks of the same type are placed on top of the other. The user only defines the blocks of one single level of the cabinet (main level or basic level) as normal blocks in the switchboard. The routes that

connect the blocks on the main level with the neighboring entry/exit blocks are also defined as normal routes in the switchboard. So basically only the main level of the cabinet is defined by the user.

In addition, the user enters in the properties of the cabinet how many levels (one above the other) the cabinet has. **TrainController™** then automatically calculates the other blocks that are in these levels as copies of the blocks of the main level. Copies of the routes of the main level are also generated, with which the artificially created further blocks are automatically connected to the neighboring entry/exit blocks.

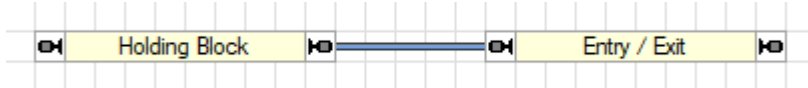


Diagram 82: Main Level of a Cabinet – defined by the end user in the Switchboard

The end user only specifies the blocks and routes in the main level of the cabinet...

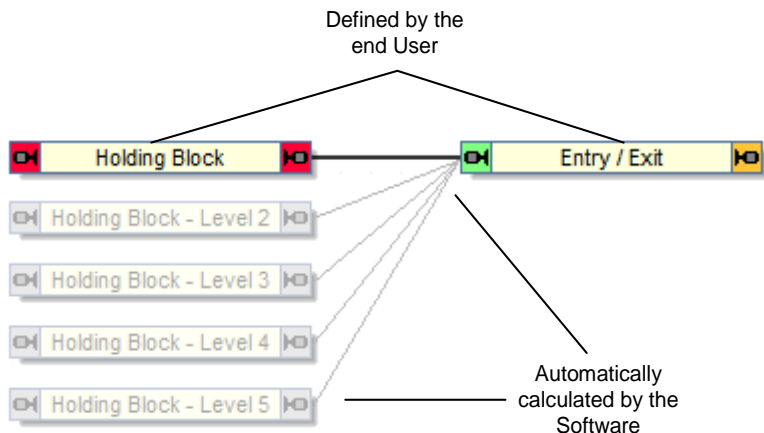


Diagram 83: Levels of a Cabinet

... **TrainController™** then automatically calculates the blocks and routes on the other levels as almost exact copies of the holding blocks and routes of the main level.

Each additional copy of a route is automatically attached to the same entry/exit block as the corresponding original route on the main level. This means that trains can use these routes to get from the entry/exit blocks to the holding blocks on the upper levels and vice versa back to the entry/exit blocks.

In addition to the copies of blocks and routes for the levels in the cabinet, TrainController also automatically creates a cabinet object in the turntable window. This cabinet object represents a virtual or actually existing train lift with which the individual levels of the cabinet can be pushed into the position in which an entry or exit for the relevant level can be carried out.

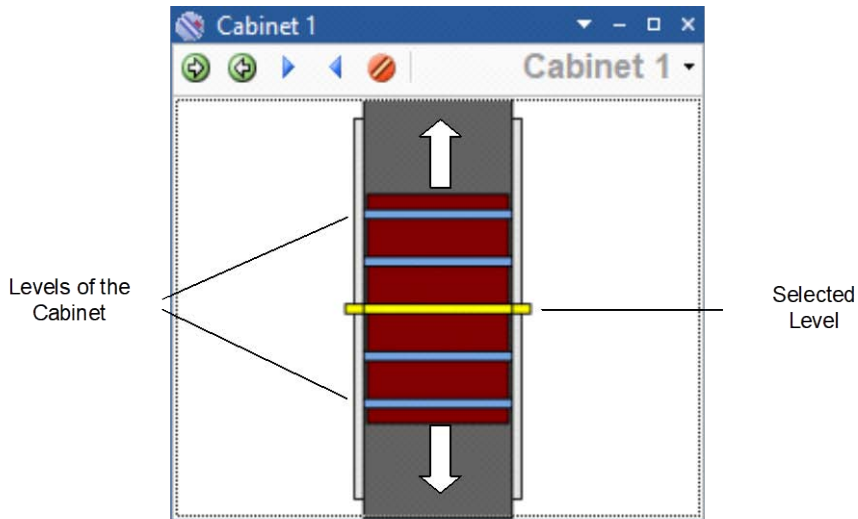


Diagram 84: Cabinet in the Turntable Window

In the turntable window, the cabinet object can be operated manually in a similar way to a turntable or transfer table. All levels of the cabinet are then always shifted at the same time until the level is selected into which an entry or exit is to take place.

For automatic operation, this cabinet object is automatically integrated into the routes of the cabinet with the position belonging to the respective level. If a train is to enter or leave a certain level, the cabinet object is automatically moved to the appropriate position when a route is set to a block on this level. This is particularly important if the cabinet is equipped with a mechanically operated lift that can be controlled by the computer. More on this below.

Train Lifts and Paternosters

Cabinet objects offer many settings that are also available for turntables and transfer tables. If a mechanical train lift is used that can be controlled by the computer, **TrainController™** offers settings for its control. Diagram 85, for example, shows the configuration for a Muet Lok-Lift.

Diagram 85: Setting up the controlling Hardware of a Cabinet

The operation of the cabinet object for the selection of the various levels for entry and exit and thus also the operation of a train lift connected on hardware side is automatically entered by **TrainController™** for all levels of the cabinet in the routes belonging to the individual levels. In this way, a train lift and all associated parking blocks can be integrated into the automatic control of your system in a very simple way and trains can be automatically parked in the individual levels of the lift.

Lifts and paternosters move differently. While lifts always move back and forth between two end positions, paternosters can move around in a circle beyond the end position. With an appropriate selection option, **TrainController™** can be adapted to both behaviors.

Display Cases and Fiddle Yards

Display cases and fiddle yards, which cannot be automatically controlled by the computer like a train lift, can at least largely be included in automatic processes on your model railway as well. This is done as shown in the following diagram.

Cabinet - Cabinet 1

General | **Connection** | Customize | Tracks | Operations | Comment

Type:
 Type: Manual Cabinet / Fiddle Yard

Connection:
 Digital System:
 Address: 0 Output: 1

Timing:
 Switch Time: 100 Step Time: 1000

Position Control:
 Feedback: Cabinet Ready

OK Cancel Help

• • • •

Diagram 86: Setting up the Hardware of a manually controlled Cabinet

Especially for safe integration into automatic operation, there is also the option to specify a push button or feedback indicator, by pressing which the user can inform the program during operation that the cabinet is ready for a train to enter or leave (see option **Position Control** in the above diagram). Instead of a push button symbol, a feedback indicator can also be specified here. This means that a button on the hardware side, which is connected to an input of a feedback module, can also be used.

If a display case consists, for example, of individual shelves equipped with rails, which are removed from the display case for entry or exit of a train and attached to the model railroad, the user can press a button to inform the program that the track is ready for entry or exit. The process is as follows:

- **TrainController™** requests a route to a level of the cabinet during operation.
- The user now ensures that the shelf belonging to the level is removed from the display case and attached to the model railroad.
- When this has happened, he presses the button specified in the **Position Control** option.
- This is the signal for **TrainController™** that the requested route is now ready and the train can move to the track on the attached shelf.

This works in a similar way even if trains are not stored in a display case, but are packed back in their original boxes.

A fiddle yard, in which several parallel tracks are mounted on a sliding board and the track intended for entry or exit is selected by moving the board, is operated in exactly the same way. Here, by pressing the button, the user reports that the movable board is in the right position. The fact that the tracks are arranged next to each other on such a sliding board and not on top of each other like in a display case or train lift does not matter for **TrainController™**. This has no effect on the operation principle.

Entry and Exit Tracks on different Levels

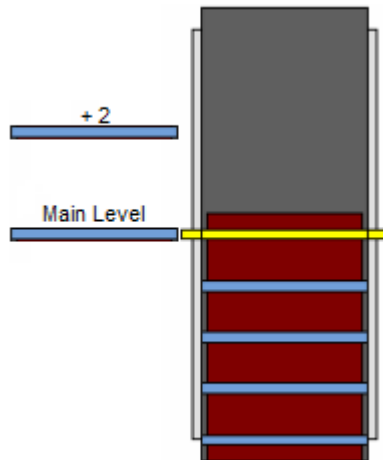


Diagram 87: Entry /Exit Tracks on different Levels

It is also possible that entry and exit tracks are on different levels. Diagram 87 shows a cabinet with entry/exit tracks on different levels. While the cabinet is normally positioned in such a way that the level intended for entry and exit is where the lower track is in the above diagram, a different position must be approached here for entry or exit via

the upper track. For example, if the upper track is 2 levels above the lower track, the cabinet must move 2 positions further for entry and exit via the upper track than for the lower track. This additional offset can be entered in the routes of the cabinet.

Section Settings - Cabinet 1 / Route: Cabinet 1<-->Entry / Exit

General

General
 Cabinet: Cabinet 1
 Route: Cabinet 1<-->Entry / Exit

Use:
 Holding: ☐ ☒ ☐ ☐
 Entry: ☐ ☒ ☐ ☒ Exit: ☐ ☒ ☐ ☒

Track Offset:
 Track Offset: 2

OK
 Cancel
 Help

Diagram 88: Settings for a Route in a Cabinet

If, for example, 2 is entered as **Track Offset** for a route in a cabinet, then the cabinet moves 2 positions further than usual when using the associated route. So a train wants to drive into or out of level 3, the display case is moved to level 3 with a **Track Offset** of 0. With a **Track Offset** of 2, level 5 is approached.

Example: Operation of a Traverser

Diagram 89 shows a traverser with 3 tracks that can be pushed back and forth between two positions to serve two access tracks.

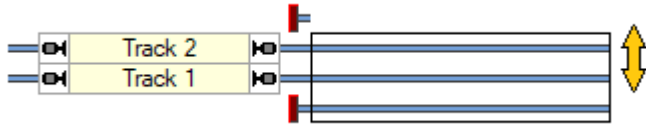


Diagram 89: Traverser

This traverser can also be controlled with a cabinet. For this purpose, a single (!) block is generated for the traverser and entered as a holding block in a cabinet. The two access blocks are also entered in the cabinet, namely as entry/exit blocks.

The two access blocks are each connected to the holding block in the cabinet with a route. Naturally, this is only possible with the help of a turnout. This turnout only plays a role visually. It does not physically exist and is set to **Without Connection** in **TrainController™**.



Diagram 90: Traverser Setup

The cabinet itself is set to 3 levels. This means that **TrainController™** automatically generates two additional blocks for these levels. Including the original block “Traverser” a block is therefore available for each of the three tracks actually present on the traverser.

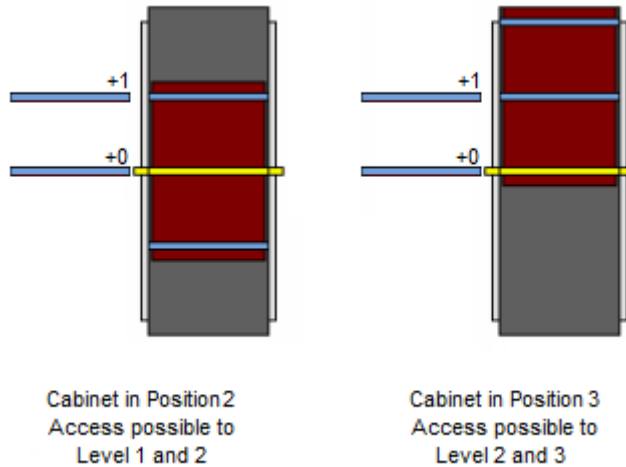


Diagram 91: Traverser Levels

In order to move the middle level (level 2) of the cabinet (=traverser) to "Track 1", the cabinet must be set to position 2 (for level 2). The track offset (see the previous section) for the route to "Track 2" is set to 1. In order to move the main level (top level 1) of the cabinet to "Track 2", position 2 is also set because of this offset (level 1 plus offset 1). That means, if the cabinet is in position 2, the levels 1 and 2 of the cabinet (= upper two tracks of the traverser) can be accessed; if the cabinet is in position 3, the levels 2 and 3 (= lower two tracks of the traverser) can be accessed. The lowest position (position 1) in which the main level (top level) of the cabinet is selected (similar to Diagram 87) and only level 1 can be accessed from "Track 1", is invalid in this case. The cabinet should be prevented from moving to this position by a suitable setting.

Cabinets, Schedules and other Facilities

The blocks and routes automatically generated by **TrainController™** for levels 2 and higher of the cabinet are normally automatically adopted by the software in all schedules and facilities in which the user has entered the original blocks and routes on the main level. This means, for example, that trains with schedules that lead to a holding block on the main level of the cabinet can automatically take these schedules to the corresponding blocks on other levels. In this way, cabinets are automatically integrated into the operation with schedules.

It is also possible to combine cabinets with other facilities. For example, fully functional hidden yards can be created in cabinets with all the functions that were described above for the hidden yard facility.

Alternatively, the user can do without this automatic transfer. He then has to enter the automatically generated blocks and routes himself in the desired schedules and facilities, but then also has the option of making individual settings for the individual blocks and routes - e.g. when a certain level should be given preference for the selection of entry or exit.

9 The Traffic Control and the Simulator

Manual operation of feedback indicators

The traffic control allows you to simulate the movement of running trains by turning on or off the contact indicators, that belong to the particular blocks, by mouse clicks.

This is not only possible by opening a traffic control window, but also directly in the switchboard or dispatcher window. If the ALT key is pressed and held in offline mode while the mouse is moved over a block in a switchboard or the dispatcher, the status of the individual feedback indicators in this block is displayed. By clicking on an indicator with the ALT key pressed, the status of this indicator can be switched on or off.

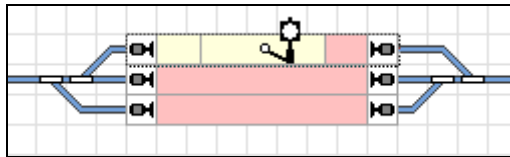


Diagram 92: Manual Operation of Feedback Indicators in the Switchboard

Diagram 92 shows three occupied blocks. With the ALT key pressed, the mouse is over the top block. There are three occupancy sections in this block, of which the right one is currently occupied. The mouse pointer is just over the middle section. This section can now be switched on or off with a click of the mouse.

11 Advanced Train Control

11.3 Approved Trains

Attributes of Train Descriptions

Train descriptions offer additional possibilities to specify the following characteristics of the affected trains:

- the minimum and maximum length
- the minimum and maximum weight
- the range for the current speed of trains

Unlike in version 9, these settings are available in all train descriptions, including those that are not created as independent objects.

By default, all minimum values are set to 0 and all maximum values to unrealistically high values. This means that these default settings apply to practically all trains and are virtually ineffective. However, if the maximum value, for example for the speed, is also set to 0, the train description only applies to stopped trains. This is different than in version 9, where a minimum and maximum value equal to 0 overruled the corresponding setting and train descriptions for stopped trains could only be created with tricks.

Conditional Train Descriptions

For train descriptions a condition can also be specified. If this condition is not fulfilled, no train fits the train description. So even a train, that meets all other criteria of the train description, will not fit the train description, if the condition is not satisfied.

In this way it is for example possible, to disable or enable certain blocks, routes, etc. for specific trains with a condition. Assume separate train descriptions for freight and passenger trains. Assume furthermore, that a certain toggle switch in the switchboard is contained in the conditions of the train description for freight trains. If both train descriptions are assigned to a block, then the block can be either enabled for passenger trains only or for both types of trains. This depends on the position of the toggle switch. This results in additional interesting opportunities to intervene in the ongoing operation of the layout.

Unlike in version 9, conditions are available in all train descriptions, including those that are not created as independent objects.

Using Train Descriptions to describe Similar Trains

Especially when starting a schedule as a successor to a previous schedule or when controlling entries and exits in a hidden yard or engine shed, it makes sense to be able to specify that the schedule to be started should be carried out with a train of the same type as the previous schedule. This is possible by specifying train descriptions for similar trains.

For this purpose train descriptions can also be used to specify similar trains.

With simple train descriptions, two types of similar trains can be defined. These are the trains to which the simple train description applies and those trains to which the train description does not apply. So if you e.g. only want to differentiate between freight trains and passenger trains at the start of a successor schedule, specify a simple train description, that e.g. applies to all passenger trains. If the schedule is carried out with a passenger train, this train description applies. The successor schedule is then only carried out with a train to which the simple train description also applies - i.e. also with a passenger train. If the schedule is carried out with a freight train, this train description does not apply. The successor schedule is then only carried out with a train to which the simple train description does not apply, too - i.e. also with a freight train.

With extended train descriptions, however, more than two types of similar trains can be defined. It is well known that every extended train description consists of several lines. Each line describes a certain composition of trains. If at least one line of a train description simultaneously applies to two different trains, these trains are considered to be similar in the sense of this train description. By specifying several lines in such a train description, different train compositions can be defined, each of which describes a certain type of similar trains. In the first line of a train description you can e.g. specify all freight trains, in the second line all slow passenger trains and in the third line all fast passenger trains. If this three-line train description is then used for a hidden yard in order to cause a similar train to leave when a train arrives, then another slow passenger train leaves the hidden yard when a slow passenger train arrives and another freight train leaves the hidden yard when a freight train arrives (see also section “Full automatic Hidden Yards” on page 92).

Since wherever similar trains play a role, you can enter a specific train description, you can also specify the similarity of trains in different ways. For a hidden yard it can be specified, that when a passenger train enters, another passenger train leaves. For an engine shed (see section 6.18, “Engine Shed”), on the other hand, you can specify that when a steam locomotive comes in, another steam locomotive also leaves; likewise when a diesel locomotive enters another diesel locomotive leaves, etc.

In this way, the similarity of trains can be determined in different ways and always adapted to the respective purpose.

14 Extended Control and Monitoring Functions

14.4 Operations

In **TrainController™ Gold**, operations can also be performed by engines and cars. They are executed, when the vehicle starts, stops, toggles its direction or is about to be decelerated to threshold speed.

19 Appendix

19.1 Migrating Existing Data Files from TrainController™ 9

Project files generated with **TrainController™ 9** are automatically converted into the format of the new version when opened in **TrainController™ 10**. The following things should be noted, however.

Open existing Files

Existing files are opened with the **Open** command in the **File** menu. After calling this command, the usual procedure under Microsoft Windows for opening files is followed.

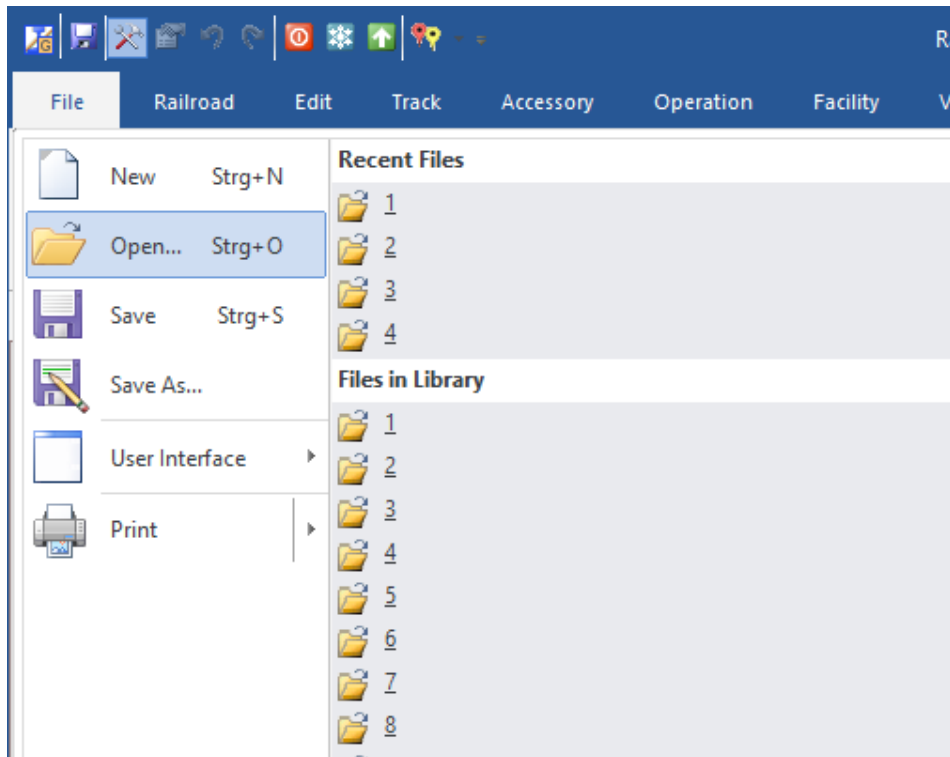


Diagram 93: File Menu

Line Up and Move Up

In **TrainController™ Gold 9** it was possible to line up trains in the destination block of a schedule and move them up later with the help of a schedule rule. This rule is no longer supported in version 10. It was replaced in version 10 by a far superior method.

For further details see section 6.12, “Line Up”.

Index

- acceleration
 - minimum acceleration time 19
- AutoTrain 20
 - direct mode 22
- block wizard 18
- booster 47
- branching 34
- Cabinet 97
- common speed limit
 - in a speed restriction section 30
- control block
 - of a hidden yard 85
 - of a siding 44
- direct mode of AutoTrain 22
- Display Cabinet 97
- empty a hidden yard 93
- engine change 69
- engine shed 94
- entry block
 - of a branching 34
 - of a cabinet 97
 - of a gradient 75
 - of a hidden yard 85
 - of a hump yard 80
 - of a run-round loop 65
 - of a siding 44
 - of an engine change 70
- exit 38
- exit block
 - of a cabinet 97
 - of a gradient 75
 - of a hidden yard 85
 - of a hump yard 80
 - of a run-round loop 65
 - of an engine change 70
- facility 24
- Facility View 24
- Fiddle Yard 97
- flow control 47
- gradient 75
- hidden yard 85
- holding block
 - of a cabinet 97
 - of a gradient 75
 - of a hidden yard 85
 - of a run-round loop 65
 - of a siding 44
 - of an engine change 70
- hump yard 80
- individual speed limit
 - in a speed restriction section 31
- instant move up
 - in a hump yard 83
- line up 58
- line up area 58
- line up marker 58
- line-up 59
- marker
 - for line up 58
 - for shunting 51
- minimum acceleration time 19
- one-way track 27

- origin block
 - in a hidden yard 92
- passing block
 - of a branching 34
 - of a hidden yard 85
 - of a one-way track 27
 - of a run-round loop 65
 - of a siding 44
 - of a speed restriction section 30
- priority of trains 40
- run-round loop 65
- separation point
 - of a hump yard 81
- shunting 50, 51
- shunting area 50
- shunting marker 51
- siding 44
- similar trains 110
 - in a hidden yard 93
- single track 24
- speed restriction section 29
- start behaviour 15
- start delay 15
- station 47
- track offset
 - of a cabinet 103
- Train Lift 97
- train priority 40
- traverser 104
- tunnel 31