RAILROAD & CO.®

TrainController™

Users Guide

for all Windows Systems

Version September 2005

Contact: Freiwald Software
Kreuzberg 16 B
D-85658 Egmating, Germany
e-mail: contact@freiwald.com
http://www.freiwald.com

Table of Contents

About this Document **9**
RAILROAD & CO. TrainController™ Users Guide 9
Help Menu 10
Quick Start - Step 1: Installation and Program Start **12**
Installation 12
Program Start 13
Quick Start - Step 2: Controlling a Train **16**
Preparing a Train for Model Railroad Computer Control 16
Controlling a Train 19
Quick Start - Step 3: Controlling Switches – The Switchboard **20**
Creating a small switchboard control panel 20
Preparing a Switch for Model Railroad Computer Control 23
Quick Start - Step 4: Creating Blocks - Tracking Train Positions **25**
Equipping the layout with feedback sensors 25
Dividing the layout into Blocks 26
Entering the locations of Blocks into the Switchboard 26
Assigning Feedback Sensors to Blocks 28
Displaying train positions on the Computer Screen 30
Quick Start - Step 5: Controlling Trains Automatically **33**
AutoTrain™ 33
Creating a Commuter Train 34
1 Introduction **39**
1.1 Overview **39**
Supported Digital and Control Systems 39
Modes of Train Operation 40
Use 41
Components 41
Automatic Operation 43
1.2 Fundamentals of Use **43**
The Overall Principle 43
File Handling 44
Window Handling 44
Edit Mode 45
Further Steps 45
Switchboards 46
Train Windows 47
The Visual Dispatcher 48

2 The Switchboard **50**
2.1 Introduction **50**
2.2 Size and Appearance **52**
2.3 Drawing the Track Diagram **55**
2.4 Connecting the Switches **55**
2.5 Signals and Accessories **58**
Signals 59
Accessories 59
Connecting Signals and Accessories 60
2.6 Routes **60**
Manual Routes vs. Automatic Routes 60
Recording of Routes 61
Signals in Routes and Protection of Routes 62
Operation of Routes with Start and Destination Keys 63
Interlinking Routes 64
2.7 Text Labels **64**
2.8 Images **64**
2.9 Highlighting occupied track sections **65**
2.10 Displaying Train Names and Symbols in the Switchboard **66**
2.11 Using the Computer Keyboard as a Control Panel **66**
3 Train Control **67**
3.1 Introduction **67**
3.2 Engines and Trains **69**
3.3 Throttle and Brake **71**
3.4 Speedometer and Odometer **72**
3.5 The Speed Profile **72**
Preparing the decoder 72
The simplified Profile 73
Advanced Fine Tuning of the Speed Profile 74
Measuring with Momentary Track Contacts 76
Measuring with Occupancy Sensors 77
3.6 Headlights, Steam and Whistle **80**
3.7 Multiple Units **81**
Operation of Additional Function Only Decoders 83
Example: Automatic Car Lighting 83
3.8 Acceleration and Train Tonnage **84**
3.9 Coal, Water and Diesel **85**
3.10 Monitoring the Maintenance Interval **86**
3.11 Passing control between Computer and Digital System **87**
4 Contact Indicators **88**
Momentary Track Contacts vs. Occupancy Sensors 89

5 The Visual Dispatcher I ... 92
5.1 Introduction ... 92
5.2 Blocks ... 96
Blocks on the Layout ... 96
The Main Block Diagram ... 98
Links and Routes between Blocks ... 101
5.3 Direction of Travel vs. Engine Orientation ... 102
Direction of Travel ... 102
Engine Orientation ... 103
5.4 States of a Block ... 103
Occupied Block ... 104
Reserved Block ... 104
Current Block ... 104
Display of Train Positions ... 106
Locking of Blocks ... 106
Locking the exit of Blocks ... 106
5.5 Train Detection and Train Tracking ... 106
Train Detection ... 107
Train Tracking ... 111
5.6 Blocks and Indicators ... 112
5.7 Stop and Brake Indicators ... 114
5.8 Arranging Indicators in a Block ... 117
Arranging Momentary Track Contacts and Occupancy Sensors in a Block ... 117
One Sensor per Block: Combined Brake/Stop Indicators ... 121
5.9 Block Signals ... 122
General ... 122
Signal Aspects ... 123
Colour ... 123
How to use Signals on the Model Railroad Layout ... 124
How Block Signals Work ... 125
Additional Notes ... 126
5.10 Schedules ... 127
Schedule Diagrams ... 127
Start and Destination of a Schedule ... 128
Alternative Paths ... 130
5.11 Execution of Schedules ... 133
Starting a Schedule ... 134
Reservation of Blocks and Routes ... 134
Path Selection ... 137
Release of Blocks and Routes ... 138
Simulation of Train Movements without Connection to a Model Railroad ... 138
Restricted Speed, Wait Time and additional Operations ... 139

Type of a Schedule - Shuttle and Cycle Trains 140
Shunting..140
Running Trains manually under Control of a Schedule 141
5.12 AutoTrain – Start of Schedules made Easy..142
5.13 Successors of a Schedule 144
5.14 Schedule Selections 145
6 The Traffic Control...146
7 The Inspector...147
8 The Message Window 149
9 A Sample Layout...150
General 150
Step 1: Creating the Switchboard 151
Step 2: Defining the Engines 152
Step 3: Creating Blocks 154
Step 4: Contact Indicators...156
Step 5: Creating Schedules 159
Manual Operation 160
Further Steps..160
10 The Clock..164
11 Indicators and Semi-Automatic Control...165
11.1 The Memory of Indicators 165
Example: Preventing an Indicator from Flickering 166
11.2 Protection and Locking with Conditions 167
11.3 Operations 168
Example: Automatic Reset of Signals 170
Example: Emergency Stop Button...171
11.4 Semi-Automatic Control Mechanisms using Flagman Elements 171
The Flagman 171
Flagmen and Operations...172
Flagmen and Conditions...173
Example: Detecting Train Direction..173
Example: Detecting uncoupled Cars..174
Example: Simple Track Occupancy Detection 176
11.5 Macros 177
Example: Automatic Engine Whistle 177
11.6 External Control Panels 178
12 The Visual Dispatcher II 180
12.1 The manually created Main Block Diagram 180
Editing the Block Diagram 181
Routes and Links 183

12.2 Nodes **185**
12.3 Virtual Contacts and Virtual Occupancy Indication **187**
General 187
Using Virtual Contacts as Indicators in a Block 189
Stopping in the Middle of a Platform 192
Virtual Occupancy Indication 192
12.4 Controlling the traffic flow in Schedules **193**
Limiting the Reservation of Blocks and Routes in certain Schedules 193
Preferring Blocks 194
Critical Sections 194
Long trains to Long Tracks – Short Trains to Short Tracks 196
Routes with separate occupancy indication 198
12.5 Examples **198**
Example: Manual Control of Station Entry 198
Example: Manual Control of Station Exit 200
Example: Hidden Yard with Train Length Control and Automatic Bypass 201
12.6 Timetables **204**
13 The Turntable Window **207**
13.1 Introduction **207**
Supported Turntable/Transfer Table Commands 208
13.2 Configuring a Turntable or Transfer Table **209**
13.3 The Type of a Turntable/Transfer Table **210**
Digital Turntable 210
Analog Turntables/Transfer Tables 211
Generic Turntables 212
13.4 The Track Layout of a Turntable/Transfer Table **212**
13.5 Turntables and Feedback **213**
13.6 Automatic Operation of Turntables **214**
Turning Engines Automatically - The Turn Boundary of a Turntable 216
Example: Turntable and Roundhouse 220
13.7 Turntable Operations **229**
Example: Indexing of an Analog Turntable 231
14 Special Applications **234**
14.1 Mixing manual and automatic Operation **234**
Passing trains from manual to automatic control 235
Passing trains from automatic to manual control 235
Passing trains from manual to automatic control without the use of a train detection system 236
14.2 Running Conventional Engines without Decoder **236**
Stationary Block Decoders 236
Computer Command Control 237

Computer Section Control 237
Computer Cab Control...238
Adjusting the Polarity of each Block...240
Running conventional and digital Engines on the same Track...........................242
Notes...242
Additional Options 243
14.3 Operating Several Digital Systems Simultaneously...................................245
14.4 Migrating Data Files created with Version 4.x or earlier to Version 5....246
Blocks..246
Routes..246
Lines and Schedules 247
Other Elements 248
Free-style Schedule Diagrams 249
Suggestions for the subsequent manual Data Migration......................................250
List of Examples 252
Index 253

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 computer controlled model railroads.
- **TrainProgrammer**™ is the program, which makes programming of DCC decoders as simple as a few clicks with your mouse.
- **+Net**™ is a module, that allows to control your layout with a network of several computers running **TrainController**™.
- **+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.
- **TrainMonitor**™ is the world's first program, which is especially made for indication of train positions on the computer screen based on train detection and train tracking.
- **RAILROAD & CO. Handheld** is the world's first remote control especially designed for computer controlled model railroads.

RAILROAD & CO. TrainController™ Users Guide

An overview of the basic concepts of **TrainController**™ is provided in this Users Guide. By reading this document one can obtain information about the many features of the product. Additionally you are provided with background information necessary for model railroad computer control with **TrainController**™.

The document is divided into three parts. Part I provides a quick start tutorial for users, who are in a hurry and are in a fever to start quickly. Part II explains the fundamentals of use. Knowing the contents of this part you will be able to control your switches, signals, routes and trains manually and to perform basic automatic operation. Novice users should focus to this part first and put its content into practice before proceeding with Part III. Part III explains extended features of the software for professional use of all possibilities by advanced users.

Details of usage are mentioned only if they are necessary to understand the related issues, or to point to important features of the program. If you want to know in detail, how specific functions are to be used, refer to the **Help** menu of **TrainController**™, please.

Several sections or paragraphs show additional markings for novice or advanced readers or to indicate important notes. The markings and their meaning are:

B

Basic content. Novice readers should focus on these parts.

X

Extended content for advanced users. Novice readers should ignore these parts in the beginning.

!

Important note.

Help Menu

The help menu installed with **TrainController™** contains detailed reference information necessary for using the program. All menus, dialogs and options are completely described and can be referred to in case of questions or problems.

!

Please note: no document is complete without the other. If you want to know, what a certain term means or what a certain function does, refer to the Users Guide, please. If you want to know, how a certain object is to be edited or how a specific function is to be executed, call the help menu.

Part I

Quick Start

B

Quick Start - Step 1:
Installation and Program Start

You have obtained **TrainController**™ to control your model railroad with your computer. It is easily understood, if you are eager to control your layout with your computer as soon as possible. If you are in a hurry about starting without reading the complete Users Guide first, you can also reconstruct the following quick start tutorial about **TrainController**™.

Detailed explanations about the concepts, that are the fundamentals of the following, can be found in Part II of this document. It is strongly recommended to study the contents of this part prior to working seriously with **TrainController**™.

Now let us start:

Installation

The installation file of **TrainController**™, its name is SETUP.EXE, can be downloaded from the download area of the Internet home page of the software (www.freiwald.com) or started from a CD ROM.

After starting SETUP.EXE a self-explaining window is displayed, that guides you through the steps, that are necessary to install **TrainController**™ on your computer.

Diagram 1: TrainController™ Setup Screen

Ensure, that you select the right language, because the selected language will also appear later, when running **TrainController™**.

Before you start **TrainController™** you should connect your digital system, with which you are controlling your model railroad, to the computer. Please refer to the instructions provided by the manufacturer of your digital system, how this is done.

Program Start

After correct installation of **TrainController™** there should be an entry in the **Start** menu of your Windows system, with which you can start the software.

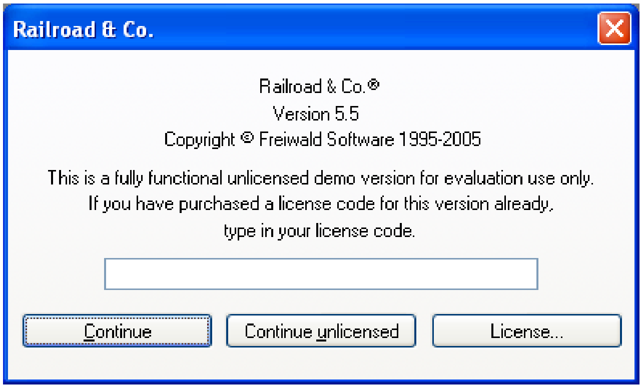


Diagram 2: License Inquiry

After start of the program the software first inquires your license key. Do not be concerned, if you are not yet in possession of such key. Press **Continue unlicensed**, if you want to try the software before buying it.

In the next step the connected digital system is to be configured. Usually the following screen appears automatically, when the program was started the first time. If the program starts without displaying the screen shown below, then call the **Setup Digital Systems** command of the **Railroad** menu.

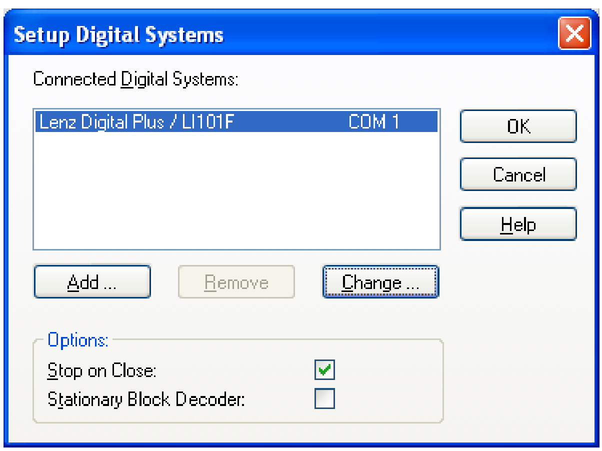


Diagram 3: Setup Digital Systems dialog

If your digital system and/or the serial port of your computer, to which your digital system is connected, is not displayed accordingly, press **Change** to select the right settings.

In order to test, whether the connection to the digital system is properly established, play around a little bit with the **Power Off** and **Power On** command of the **Railroad** menu. These commands stop or start your digital system, respectively. Your digital system should respond accordingly to these commands. If your digital system does not respond or if there are even some error messages, then do not proceed any further, until this problem is resolved. In case of problems in this area, check very thoroughly, that the digital system is properly connected to the computer according to the instructions of the manufacturer.

If the steps outlined above have been performed correctly, you are ready to do the first steps into model railroad computer control.

Quick Start - Step 2:
Controlling a Train

Preparing a Train for Model Railroad Computer Control

First put a train onto the tracks of your layout and run it with your digital system. This step is recommended to verify, that the digital system and the train are correctly running and also, to bring the digital address of the train back to your mind. This is needed a few moments later.

Now ensure, that the **Edit Mode** option in the **View** menu is active.

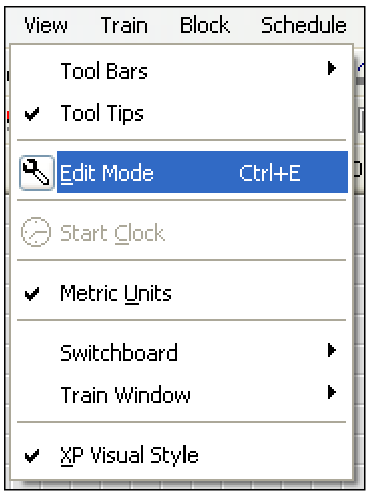


Diagram 4: View Menu

In this mode it is possible to enter new data into the software or to change existing data. This is what we want to do next.

Call the **New Train Window** command of the **Window** menu. If this is done correctly, the following window will appear on your computer screen:

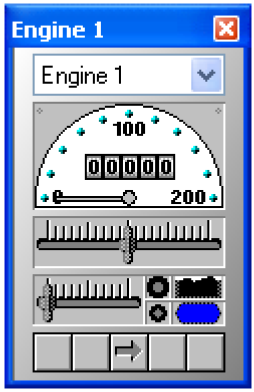


Diagram 5: Train Window

If you want to learn more about the various controls of this window, refer to chapter 3, “Train Control”, please.

Now select the **Properties** of the **Edit** menu. This is one of the most important commands of **TrainController**™. It is used for all objects contained in the software (trains, turnouts, signals, routes, etc.), whenever it is required to change the settings of a particular object. The following window is displayed now:

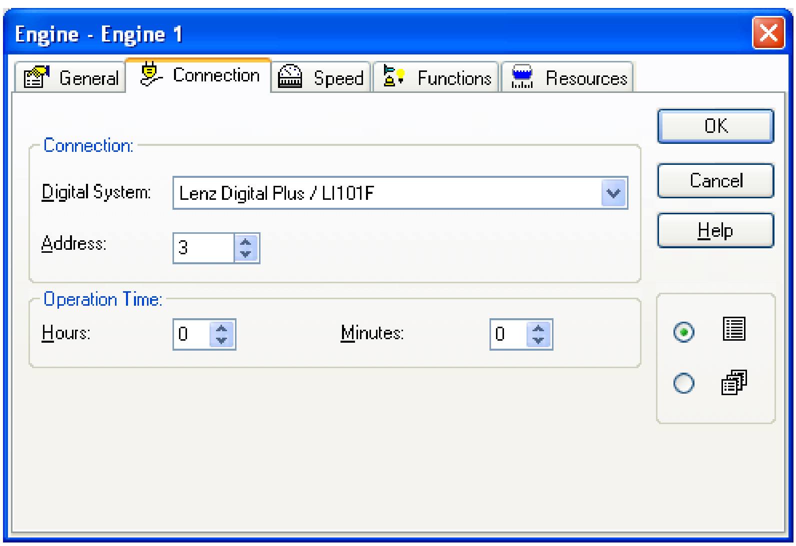


Diagram 6: Specifying the Digital Address

Specify the same address, that you have been using previously to control the train with your digital system, in the field labelled **Address**. If you want to give your engine a name, that can be more easily remembered, select the tab labelled **General** and enter an appropriate name. In the following we want to call this train “Passenger Train”.

You can see this name entered into the program in the image displayed below:

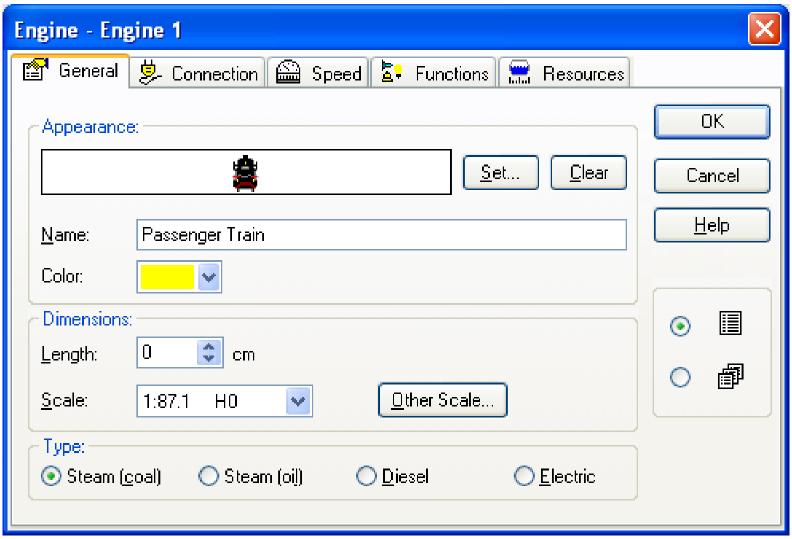


Diagram 7: Entering a Name

You may have noticed, that the term “train” is being used here, while the images show the term “engine”. If you want to read more about this difference, refer to section 3.2, “Engines and Trains”, please. In the following we will continue using the more general term “train”.

Now press **OK** to close the dialog and to commit these changes. We are returning now to the main screen and are ready to control the train:

Controlling a Train

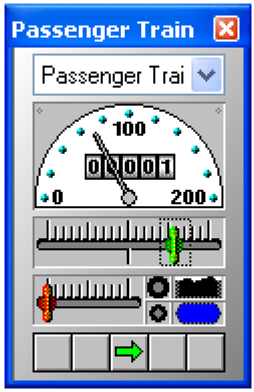


Diagram 8: Train Window

You may notice, that the colour of some controls in the train window changed. This happened due to the fact, that we entered a digital address for our train. Now the software knows, how to control the train. To prove this move the mouse to the green control in the centre of the window. Click to it and drag the green control to the right. If everything has been done correctly so far, the train will slowly start to move. We have done the first successful step into model railroad computer control!

Before continuing I suggest to enjoy playing with the train. Play around with the green control, which is actually an on-screen throttle, drag it to the right and back to zero, then to the left and watch, how your train responds to these actions. See, how the speedometer needle top of the throttle indicates the scale speed of your running train. Watch the odometer increasing. By clicking the green arrow at the bottom of the screen you will reverse the direction of your train. Dragging the red control, which is located between the throttle and the green arrow, will slow down the train. This control is actually a brake control. It can be used by experienced users, to apply the brake to a running train.

There are much more things, that **TrainController**™ can do for realistic control of your trains. You can operate auxiliary functions (light, whistle, coupler, etc.), simulate the consumption of resources, adjust the momentum to your personal needs and scale the speed and distance measurements to the physical characteristics of your train. This is discussed in detail in chapter 3, “Train Control”.

Quick Start - Step 3:
Controlling Switches – The Switchboard

Creating a small switchboard control panel

So far the area in the background of the main window of **TrainController™** is still empty. It contains a number of cells, that are arranged in rows and columns. These cells are still empty. We want to fill this empty area with a small switchboard control panel for the following small track layout:

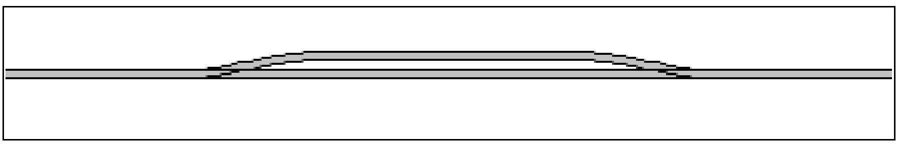


Diagram 9: Small Sample Layout

In the first step we draw the track diagram in the switchboard window. First ensure please, that **Edit Mode** in the **View** menu is still turned on (see Diagram 4). Next select the **Draw** tool in the **Tools** menu.

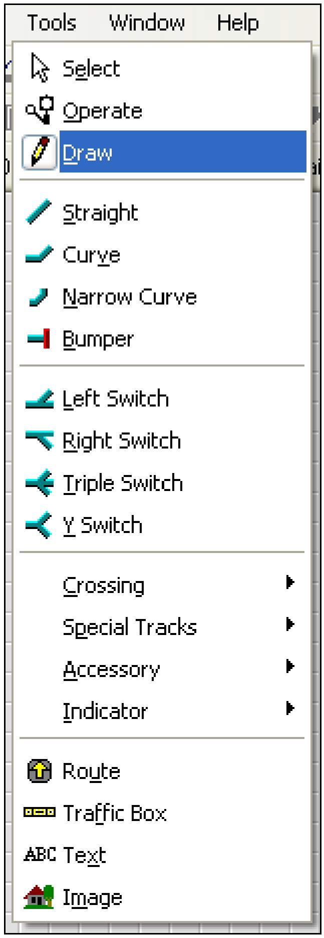


Diagram 10: Tools Menu

Now move the mouse to a cell in the switchboard window, where the left end of our track diagram shall be located. Click and hold the left mouse button and drag the mouse about 25 cells to the right. Then release the left mouse button. The following image should be visible now in the switchboard window:

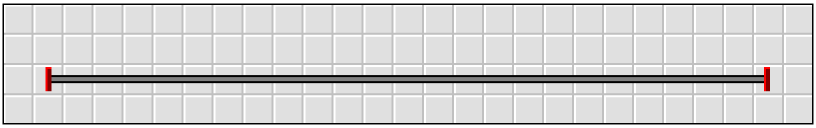


Diagram 11: Straight track section

We have drawn a straight track section. Now move the mouse to a cell on this track section, that is located about one third right of the left end. Click the left mouse button and drag the mouse one cell to the right and one cell up. Then release the left mouse button. Now you should see something similar to the following:

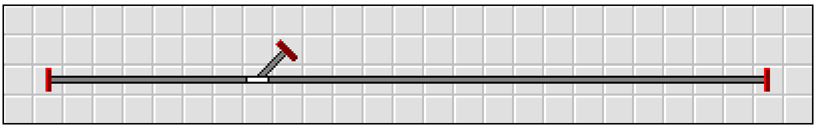


Diagram 12: Track section with switch

The first switch (or turnout, respectively) in the switchboard is created now. Now click to the cell, where the diverging route of this switch ends and drag the mouse to the right to a cell, that is located about one third left of the right end of the straight track section.

Diagram 13: Extending the track diagram

Finally click to the cell, where the last mouse movement ended, and drag the mouse one cell to the right and one cell down.

Diagram 14: The complete track diagram

The track diagram of our small sample layout is now completed and should look similar to Diagram 14.

If you want to operate real switches of your existing model railroad with the track diagram control panel just created, try to identify a small area of your layout, that contains a similar track structure with two switches as shown above. Now operate these switches with your digital system. This step is recommended to verify, that the digital system and the switches are correctly working and also, to bring the digital addresses of the switches back to your mind. This is needed in the next step.

Preparing a Switch for Model Railroad Computer Control

Ensure, that the **Edit Mode** option in the **View** menu is still active (see Diagram 4).

Now click to the symbol of the left switch in the track diagram and select the **Properties** of the **Edit** menu. Do you remember? This command is used for all objects contained in the software (trains, turnouts, signals, routes, etc.), whenever it is required to change the settings of the particular object. The following window is displayed now:

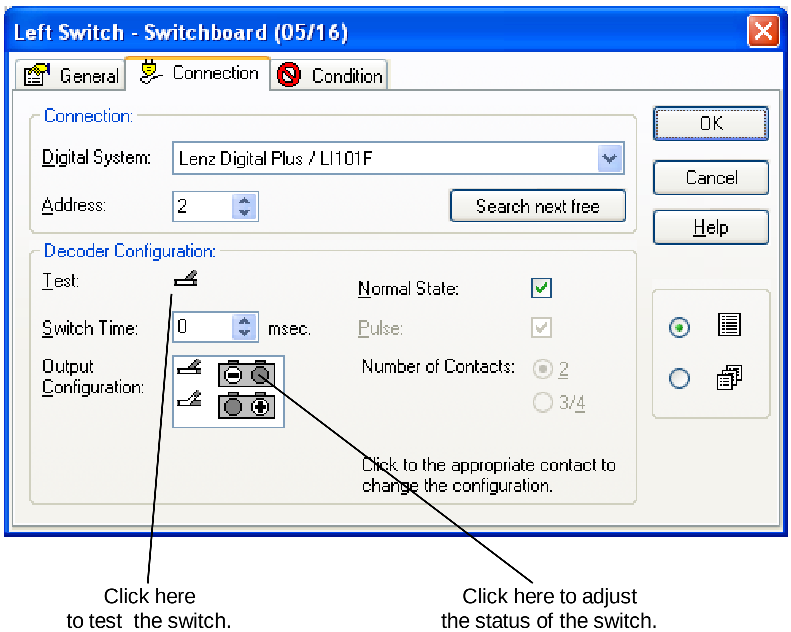


Diagram 15: Specifying the Digital Address

Specify the same address, that you have been using previously to control the corresponding real switch with your digital system, in the field labelled **Address**. Now click to the symbol of the switch, that is located right of the label **Test**. The real switch on your model railroad layout should respond now. Depending on the wiring of your switch it may happen, that the image in the software and the physical switch do not show the same status (closed vs. thrown). If this is the case click to the grey circle in the upper row of the **Output Configuration** to adjust the displayed status (see Diagram 15). The highlighting in the **Output Configuration** should change now and the displayed image of the switch and the status of the switch should be in sync, when you test the switch again.

Some advanced background information: in many cases and depending on the used digital system the highlighting in the **Output Configuration** reflects the keys, that are to be pressed on the handheld of your digital system to set the switch (or any other accessory, that is operated by switch commands) to the corresponding state. Whenever the display of the switch on the computer screen and the status of the switch on your layout are not in sync, then you should operate the switch first with your handheld and remember the keystrokes, that are to be pressed to achieve a certain state. Then translate these keystrokes to the **Output Configuration** of this switch.

If you want to give your switch a name, that can be more easily remembered, select the tab labelled **General** and enter an appropriate name.

Now press **OK** to close the dialog and to commit these changes. We are returning now to the main screen and are ready to control the switch. To do this, turn off **Edit Mode** in the **View** menu (see Diagram 4), move the mouse to the symbol of the switch in the track diagram of the switchboard window, click to this symbol and watch, how the real switch on your layout responds.

Finally perform the same for the right switch symbol in the track diagram.

We are now able to control a train and a small layout manually with the computer. I suggest to run the train back and forth on this small layout a little bit and to play with different routes by changing the positions of each switch prior to each run of the train.

In the next step we want to learn, how trains can be operated automatically under control of the computer.

Quick Start - Step 4:
Creating Blocks - Tracking Train Positions

Equipping the layout with feedback sensors

The most important prerequisite to control trains automatically with your computer or to monitor the movements of trains on the computer screen is equipping the layout with feedback sensors. These sensors are used to report train movements back to the computer. Based on this information **TrainController**™ is able to take the right decisions to direct automatically running trains to their destination or to monitor the movement of trains.

Feedback sensors differ in occupancy sensors and momentary track contacts. Details of this difference and more detailed information about feedback sensors can be found in chapter 4, “Contact Indicators”.

In the following we assume, that occupancy sensors are used to control our small layout and that our layout is divided into four detection sections according to the following image:

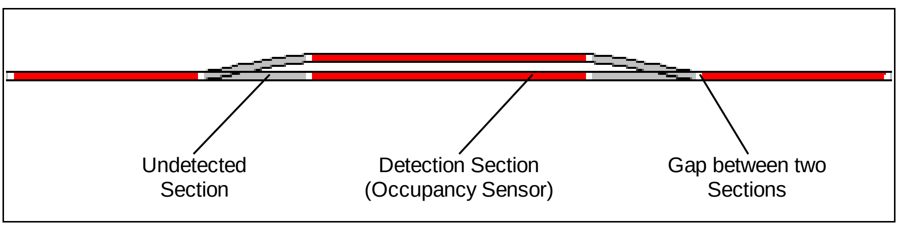


Diagram 16: Detection Sections and Occupancy Sensors

There are other possibilities to divide a layout into detection sections or to control it with momentary track contacts. The scheme displayed above is also not necessarily the optimal solution, too. The above scheme has been chosen for this tutorial for reasons of simplicity and because it is sufficient to perform a quick start. Other variants to equip your layout with feedback sensors are outlined in more detail in section 5.8.

Dividing the layout into Blocks

Another important prerequisite to control trains automatically with your computer or to monitor the movements of running trains is separating the layout into logical blocks. Blocks are the base elements for automatic train control and tracking of train positions. There is a close relation between feedback sensors and blocks: each block is associated with <u>one or more</u> feedback sensors.

There are certain guidelines for creation of blocks. They are outlined in detail in section 5.2, "Blocks". According to these guidelines we divide our small sample layout into blocks as shown below:

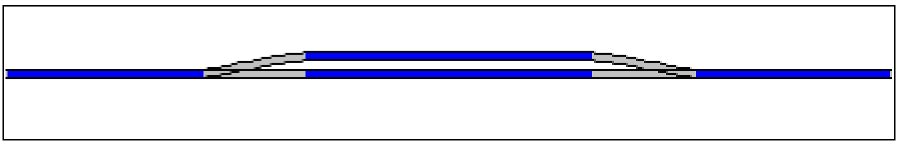


Diagram 17: Dividing a layout into Blocks

As you can see we have applied a 1:1 relation between blocks and detection sections here. Please note, that this is not always the case. In many cases more than one detection section or feedback sensor will be associated with one block. However, it is also possible to control your layout or appropriate parts of it with one feedback sensor per block. For reasons of simplicity and because it is sufficient for the quick start we go with one detection section per block here, too. Please keep in mind, however, that blocks and detection sections are <u>not</u> the same thing.

More details about this topic are outlined in detail in section 5.6, "Blocks and Indicators".

Entering the locations of Blocks into the Switchboard

Blocks are represented by **TrainController™** on the computer screen by rectangular symbols. These symbols are also called *traffic boxes*, because they usually show more than just a block. To enter the blocks or traffic boxes, respectively, that are needed to control our train on our sample layout, turn on **Edit Mode** in the **View** menu and select the **Traffic Box** command of the **Tools** menu.

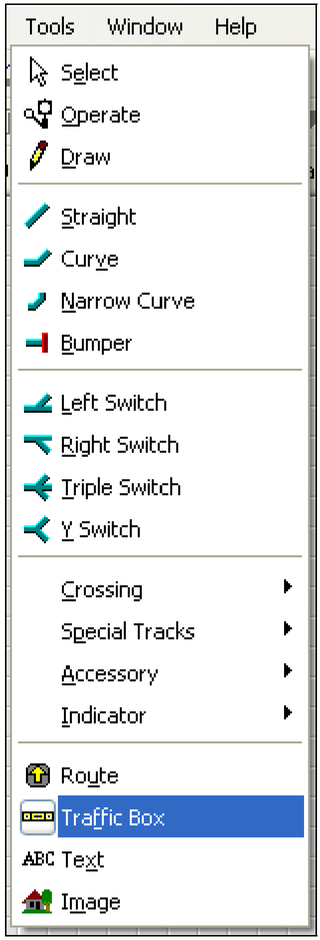


Diagram 18: Tools Menu

Now click to the cell, that is located right of the cell, that contains the left end of our track diagram. A traffic box, that represents the first block, will appear at this location.

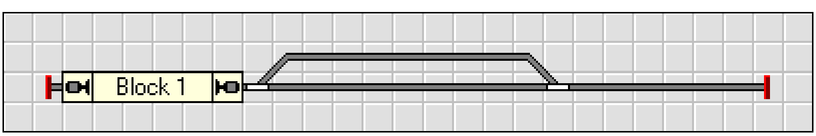


Diagram 19: Traffic Box in the Switchboard

Please do the same for the three other blocks. Note, that the cell, where you click, determines the leftmost end of the traffic box. Ensure also, that you click into a cell, that contains a piece of straight track.

You can change the size of each traffic box by dragging its left or right border.

If everything was done correctly, the track diagram should look like the following image:

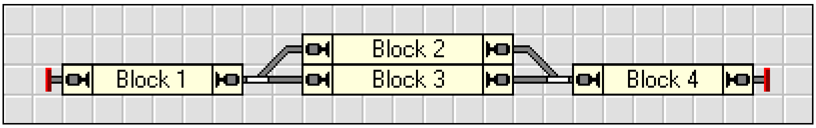


Diagram 20: The complete Track Diagram with all Traffic Boxes

Assigning Feedback Sensors to Blocks

There is a close relation between feedback sensors and blocks: each block is associated with one or more feedback sensors. Feedback sensors are represented in **TrainController™** by contact indicators. To create a contact indicator for a particular feedback sensor and to assign this indicator to a block, select "Block 1" in the switchboard track diagram and call the **Create Contact Indicator** command of the **Block** menu according to the following image:

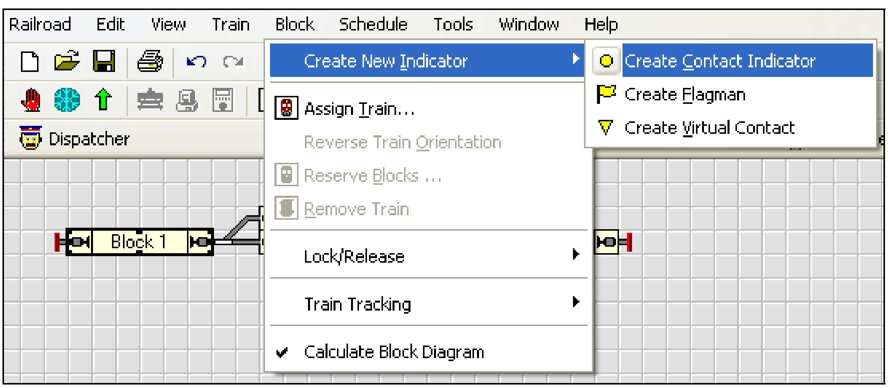


Diagram 21: Creating a Contact Indicator for a Block

The following dialog box is displayed automatically after creation of the contact indicator:

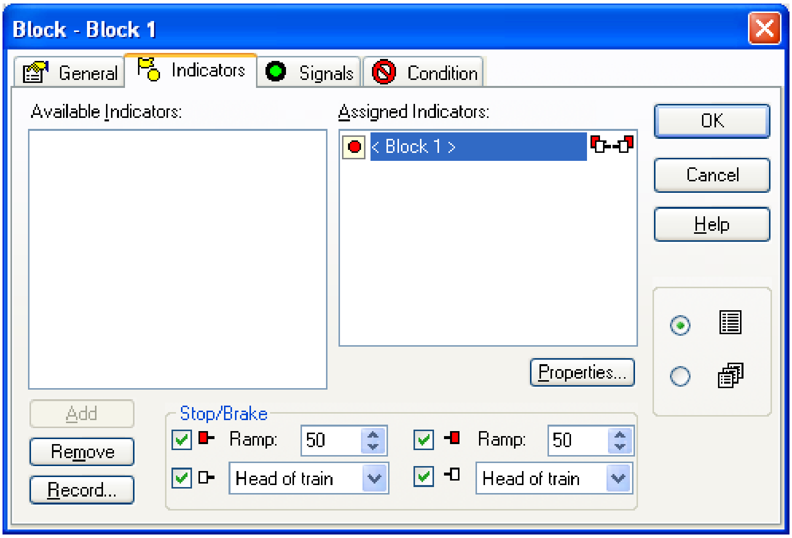


Diagram 22: Assigning a Contact Indicator to a Block

It shows the properties of the block and indicates, that one contact indicator is now assigned to this block. Double click the top entry in the right list box labelled **Assigned Indicators** or press **Properties**. The following dialog box appears:

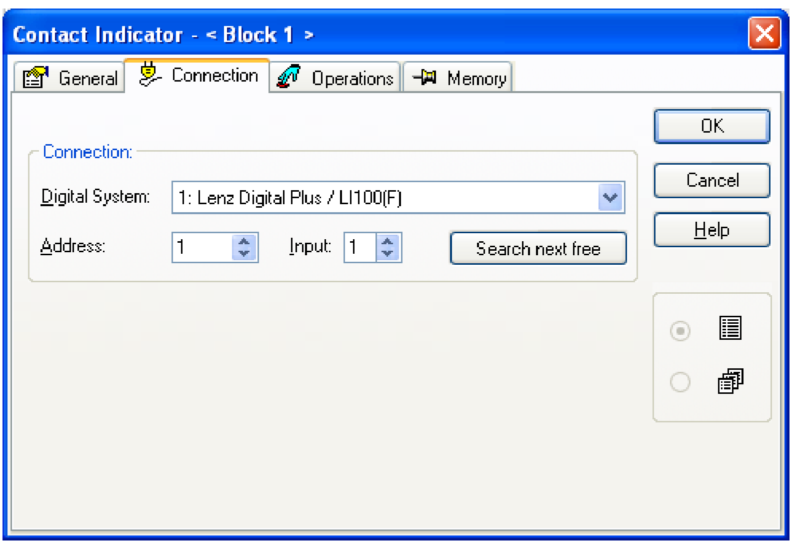


Diagram 23: Specifying the Digital Address of a Contact Indicator

Now specify the digital address of the feedback sensor, that belongs to this contact indicator. In most cases this is the digital address of the feedback decoder and the number of the contact input of this decoder, to which the sensor is connected.

To test your settings, put a train or anything else, that is suited to trigger a feedback event, into the detection section, that corresponds to “Block 1”. The traffic box in the track diagram in the switchboard should now change its colour to red:

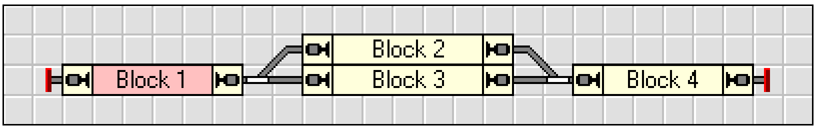


Diagram 24: Indication of an occupied Block

Now create and assign contact indicators to the other three blocks, too.

If this has been done correctly, the traffic boxes in the switchboard will change their colour according to the movements of your train on the layout. Play around a little bit with your train and watch how the blocks in the switchboard are indicated.

Displaying train positions on the Computer Screen

Now we are ready for *train tracking*, i.e. displaying of train positions on the computer screen.

To do this move your real train into “Block 1”, if it is not located there already. Ensure, that the train is heading towards the other blocks, i.e. that it has to run forward, in order to go to “Block 2” or “Block 3”, respectively.

Then turn off **Edit Mode** in the **View** menu (see Diagram 4). Next select “Block 1” in the switchboard and call the **Assign Train** command of the **Block** menu according the following image:

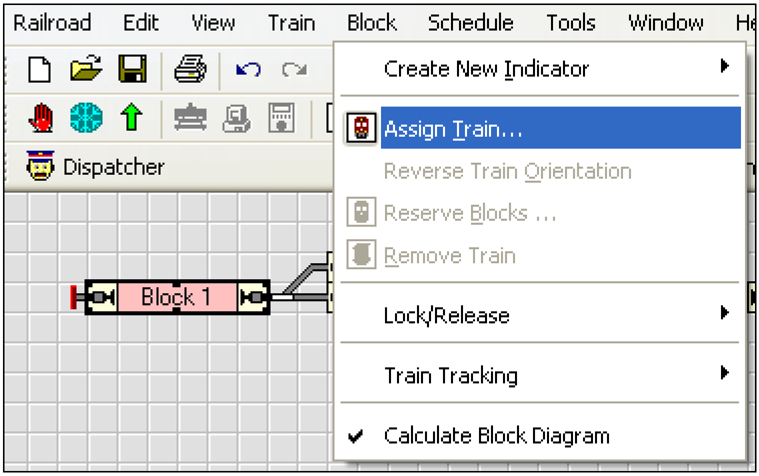


Diagram 25: Block Menu

In the following dialog select the “Passenger Train” and mark the option near the arrow pointing to the right.

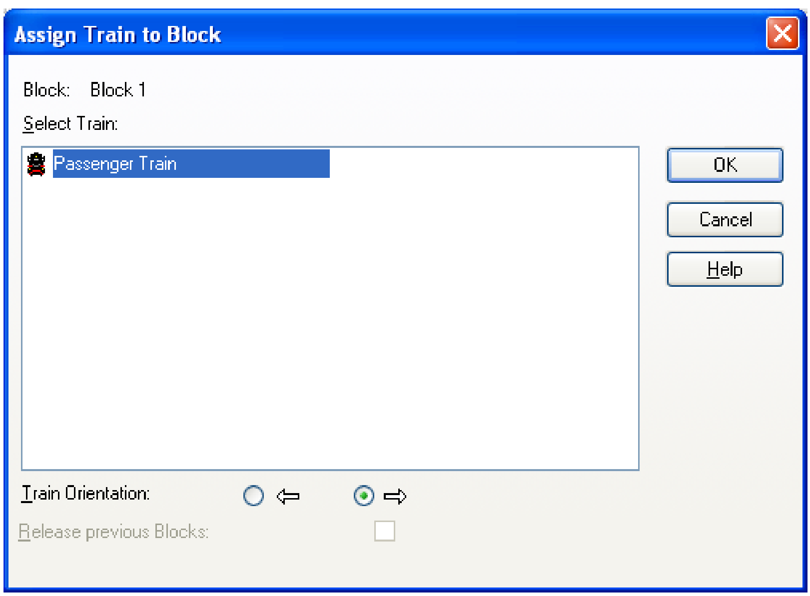


Diagram 26: Assigning a Train to a Block

After pressing **OK** the symbol and the name of the train will appear in “Block 1” in the switchboard control panel:

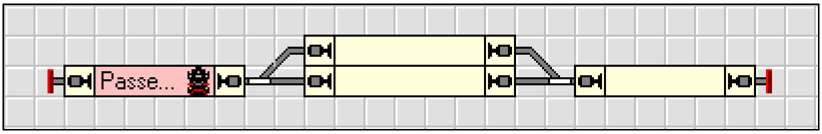


Diagram 27: Display of Train Positions on the Computer Screen

Instead of using the **Assign Train** command you can also drag and drop the train symbol with the mouse from another place on the computer screen to “Block 1”, if the train symbol is visible somewhere else.

Now run the train with the on-screen throttle of the train window displayed in Diagram 8. When the train travels to another block, then the display should be updated accordingly and the name and symbol of the train should move to the symbol of the other block, too. If you are testing this on a bigger layout ensure please, that the train does not leave the area, that is controlled by blocks and feedback sensors as described so far.

If all steps are performed correctly so far, then you are able to control the movement of your train and operate your switches with **TrainController™**. You are also able to track the positions of moving trains on the computer screen.

Quick Start - Step 5:
Controlling Trains Automatically

AutoTrain™

The last part of our quick start tutorial is automatic control of running trains. In the first step a train located in “Block 1” of our small sample layout shall run to “Block 4” and stop there. To do this run our train manually back to “Block 1”. Train tracking should ensure, that the display reflects this movement and finally looks like Diagram 27. Ensure also, please, that **Edit Mode** in the **View** menu is turned off (see Diagram 4).

Now press the key ‘A’ on your computer keyboard and move the mouse pointer to the train symbol located in “Block 1” while pressing ‘A’. The mouse cursor should now show an ‘A’ and an arrow pointing to the right:

Click to the train symbol and drag the mouse to “Block 4”, to be precise into the right half of “Block 4” until the mouse pointer shows again the same sign as displayed above. Now release the left mouse button and the key ‘A’. The display in the switchboard should now change and show something similar like the following:

Diagram 28: Running a train automatically with AutoTrain™

Simultaneously the real train on your layout should start to move now and run through “Block 2” or “Block 3” to “Block 4”, where it should slow down and stop.

After the train stopped, you can let it run back to “Block 1” automatically by dragging the train symbol back to “Block 1” while pressing the ‘A’ key as outlined above. Please ensure, that the mouse pointer now points to the left before clicking to the train symbol and before releasing the left mouse button, since the train should now run to the opposite direction.

If the train does not stop at the desired location in “Block1” or “Block 4”, then you can adjust the ramp, which is applied to the deceleration and stop of the train, by using the

Ramp option (see Diagram 22). More information can be also found in section 5.7, “Stop and Brake Indicators”.

Creating a Commuter Train

As a final step of our tutorial we want to run the train automatically back and forth between “Block 1” and “Block 4” several times. The train shall always select the right block with regard to direction of travel, i.e. when running to the right, the train shall pass “Block 3”, when running to the left the train shall pass “Block 2”. Additionally the train shall perform a short intermediate stop in “Block 2” and “Block 3”, respectively, during each pass.

To do this run the train manually back to “Block 1”. Train tracking should ensure, that the display reflects this movement and finally looks like Diagram 27. Ensure also, please, that **Edit Mode** in the **View** menu is turned off (see Diagram 4).

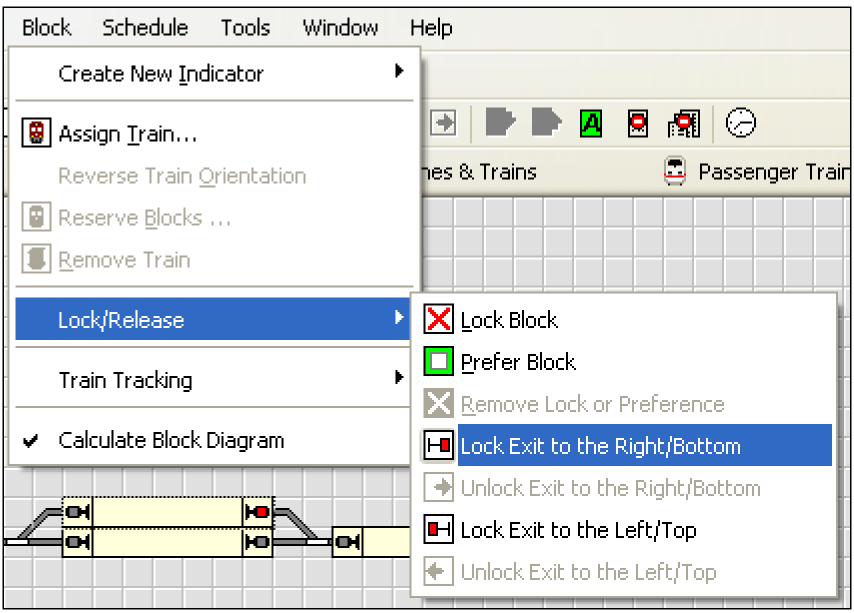


Diagram 29: Locking the exit of a Block

Now select “Block 2” and call the **Lock Exit to the Right/Bottom** command of the **Block** menu. This ensures, that the train will not pass through “Block 2” on its way to

"Block 4". Then select "Block 3" and call the **Lock Exit to the Left/Top** command of the **Block** menu.

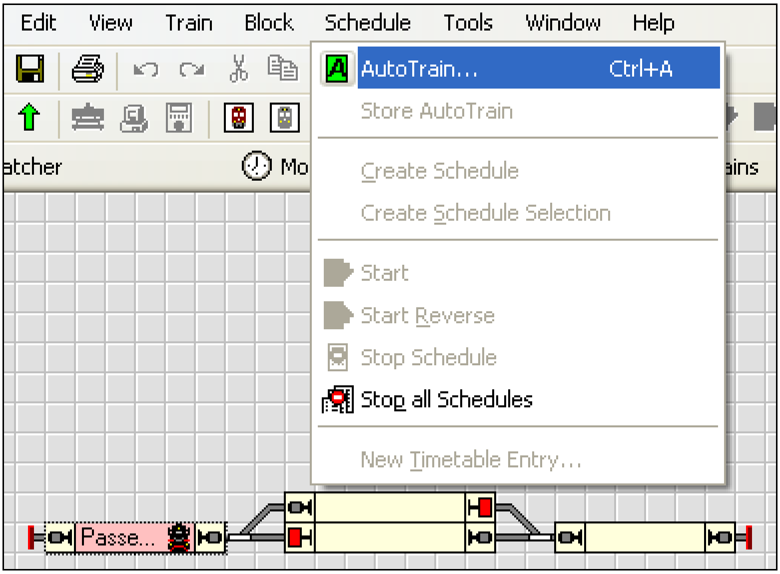


Diagram 30: Schedule Menu

Next select "Block 1" and call the **AutoTrain** command of the **Schedule** menu. This opens the **AutoTrain™** tool bar as displayed below:

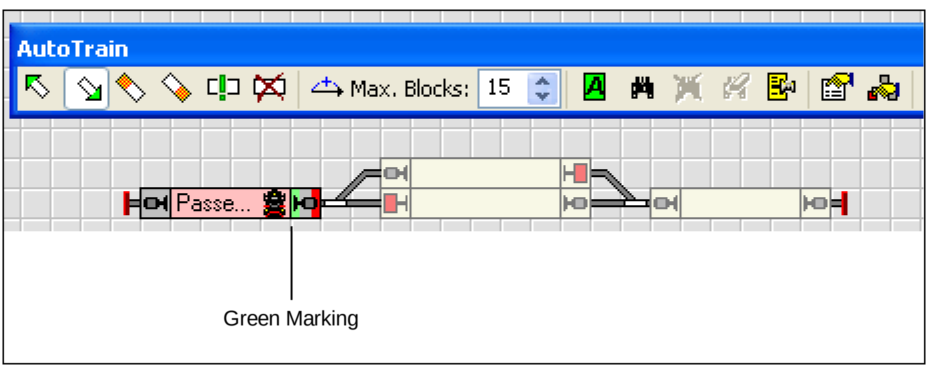


Diagram 31: AutoTrain™ Tool Bar

Ensure, please, that a green marking appears at the right side of “Block 1”. This indicates, that we want to start our train in this block with direction to the right. If this marking is not set, select “Block 1” and press .

Next select “Block 4” and press . This indicates, that the train shall enter “Block 4” from the left to the right and stop here. Now press . The software now checks, whether there is a path from “Block 1” to “Block 4”. As a result “Block 2” and “Block 3” are displayed on the screen with the same intensity as “Block 1” and “Block 4”. This indicates, that there is a path from “Block 1” to “Block 4”, that passes “Block 2” or “Block 3”, respectively.

Now press . **TrainController**™ opens the following dialog:

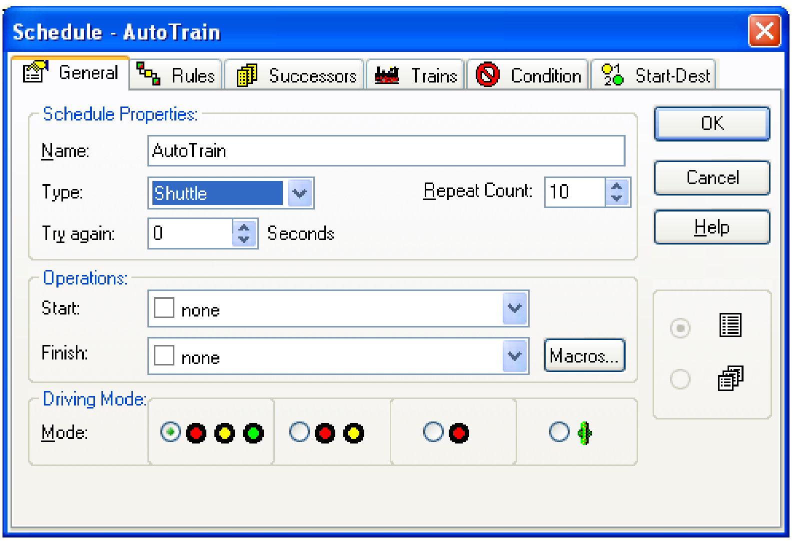


Diagram 32: Specifying a Shuttle Train

Here select **Shuttle** as **Type** and **10** as **Repeat Count**. This tells the software, that you want to create a train, that is running back and forth (shuttle) ten times. You can certainly specify any other number as **Repeat Count** as well. Commit your settings with **OK**.

Now select “Block 2” and press . **TrainController**™ opens the following dialog:

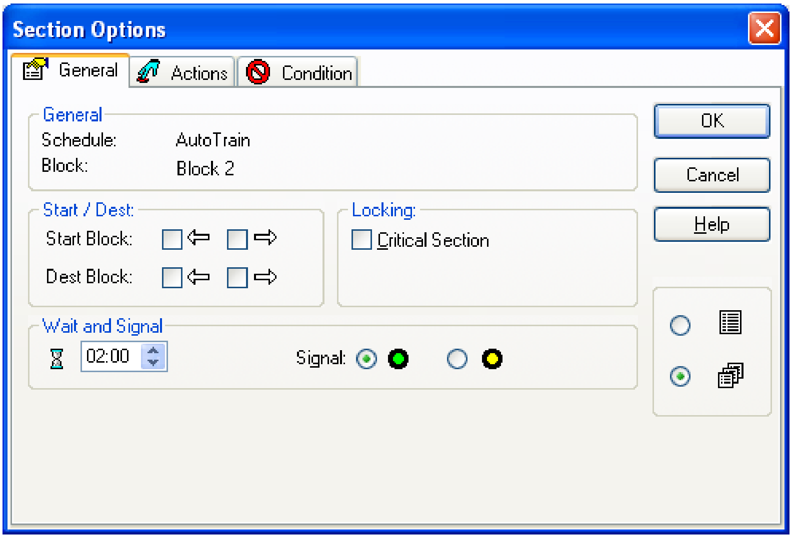


Diagram 33: Specifying a Wait Time

Enter **02:00** in the box below **Wait and Signal**. This tells the software, that the train shall wait 2 simulated minutes in “Block 2”. The duration of simulated minutes is scaled with the speed of the internal fast clock. For more information about the fast clock refer to chapter 10, “The Clock”, please. Commit your settings with **OK**. Perform the same steps for “Block 3” to specify a wait time for “Block 3, too.

Now press . The train now starts to move towards „Block 3“. In “Block 3” it slows down and stops for a few moments. Then it starts again and enters “Block 4”. Here it slows down again, stops and starts to the opposite direction. In “Block 2” it slows down and stops again. After a few moments it starts again and runs to “Block 1”, where it stops. Then the complete cycle is repeated again.

You are now able to configure control of automatically running trains. However, **TrainController**™ is certainly able to perform much more complex train control on much more complex track layouts. **TrainController**™ cannot only control perpetually running commuter trains or trains, that run perpetually around a loop. **TrainController**™ can perform intermediate train stops or execute train functions automatically, such as switching on lights or playing sounds. **TrainController**™ can operate hidden yards automatically or control trains according to time tables. Proceed with reading of part II of this Users Guide, please, to learn, how these amazing things can be done with **TrainController**™.

Part II

Fundamentals

B

1 Introduction

1.1 Overview

B

TrainController™ is a system to operate a model railroad layout from a Personal Computer running MS Windows 98 or 95, Windows ME, Windows XP, Windows 2000 or Windows NT.

TrainController™ provides you with the ease of point and click to operate your switches, signals, routes and other accessories displayed on track diagram panels. Track diagram panels are individually created for each yard or section, as desired. You can run your trains with on-screen throttles, external hand held throttles connected to your computer, or with your favourite throttles or hand held throttles supported by your digital system. You can operate digital engines equipped with their own decoders, as well as conventional models without decoders. Digital and conventional engines can run on the same track. Far-reaching automation features make railroad operations manageable by one person and match those found on the largest club layouts. You can see on the screen which engine/train is on which track.

Supported Digital and Control Systems

The software supports all digital and control systems which provide a computer interface:

- Digitrax LocoNet
- Wangrow System One
- North Coast Engineering Master Series
- Lenz Digital Plus
- Roco-Digital
- EasyDCC
- CTI
- RCI
- Märklin Digital
- Trix Selectrix
- Müt Digirail
- Rautenhaus Digital
- Fleischmann FMZ

- Uhlenbrock Intellibox
- Fleischmann Twin Center
- HSI-88
- Zimo
- MpC (Gahler & Ringstmeier)
- HELMO train identification system
- Edits
- TracTronics SECSI
- MegaDecoder
- Lionel TrainMaster
- rail4you
- PC Interface Board 8255

You can run different systems simultaneously on different serial ports. This increases the maximum number of trains, switches, signals and feedback indicators that can be operated. If your favorite digital system is not able to report the state of feedback sensors, then you are able to enlarge this system with a second system that is able to do this.

TrainController™ also supports an offline mode that allows trial operation without a connection to a real model railroad. Up to nine digital and control systems can be connected simultaneously.

Modes of Train Operation

With **TrainController**™ it is possible to run digitally equipped engines as well as conventional engines that don't have digital decoders. The operation of conventional engines is done with stationary block decoders; i.e. decoders or computer controlled throttles that are mounted at fixed positions on your model railroad rather than in each engine.

This feature is useful if,

- you have a large collection of engines and not all are digitally upgraded.
- you have a conventional - i.e. non-digital - operated model railroad and want to control your layout with your computer without installing a digital decoder in each engine first.
- the models of your engines are very small and the decoders do not fit into the engines (e.g. when you run Z scale).

In all, **TrainController**™ provides three methods of operation:

- Operating trains using mobile decoders installed in the engines ("*Computer Command Control*").
- Operating conventional trains using stationary block decoders with static assignment to track sections ("*Computer Section Control*")
- Operating conventional trains using stationary block decoders with dynamic assignment to track sections ("*Computer Cab Control*").

Additionally, it is possible to use these methods simultaneously, i.e. it is possible to run conventional engines and digital engines on the same track - even if your digital system does not support this feature.

Use

TrainController™ is easy to use. It provides an easily learned, intuitive, graphical user interface that is developed according to the following guidelines:

- Use of **TrainController**™ is possible without the need to be a computer expert or programmer.
- Graphical items are provided instead of an abstract command syntax.
- Operation is based on natural objects like trains, switches, signals, etc. instead of digital addresses or something else.
- Activities are natural - point to a signal and set it to red with a simple mouse click instead of issuing a command like "set contact output of decoder 35 to 1". Accelerate a train to speed 35 mph instead of typing "set speed level of train decoder 16 to 7".
- Automatic Operation can be arranged within minutes without the need to learn a programming language first.

Components

Each component of **TrainController**™ has its own special functionality and most of them can be used separately. You only need to concentrate on the components you choose to use. The control of trains and the operation of switches and signals is separated. However, even though the particular components can be used without the others, they can also be joined together with the help of the *Visual Dispatcher* to form a tailor-made operating system for your model railroad.

These are the components of **TrainController**™:

- **The Switchboard:** easy to use control panel editor for the operation of switches, signals and other accessories with point and click ease. It allows manual, semi-automatic and full automatic operation of your accessories.
- **The Train Window:** on-screen throttles and various cab instruments for realistic train operation
- **The Visual Dispatcher:** intelligent monitoring and operation of your entire model railroad, or just parts, that can be arranged within minutes
- **The Turntable:** manual or automatic operation of all turntables and transfer tables
- **The Clock:** fast clock with a simulated perpetual calendar

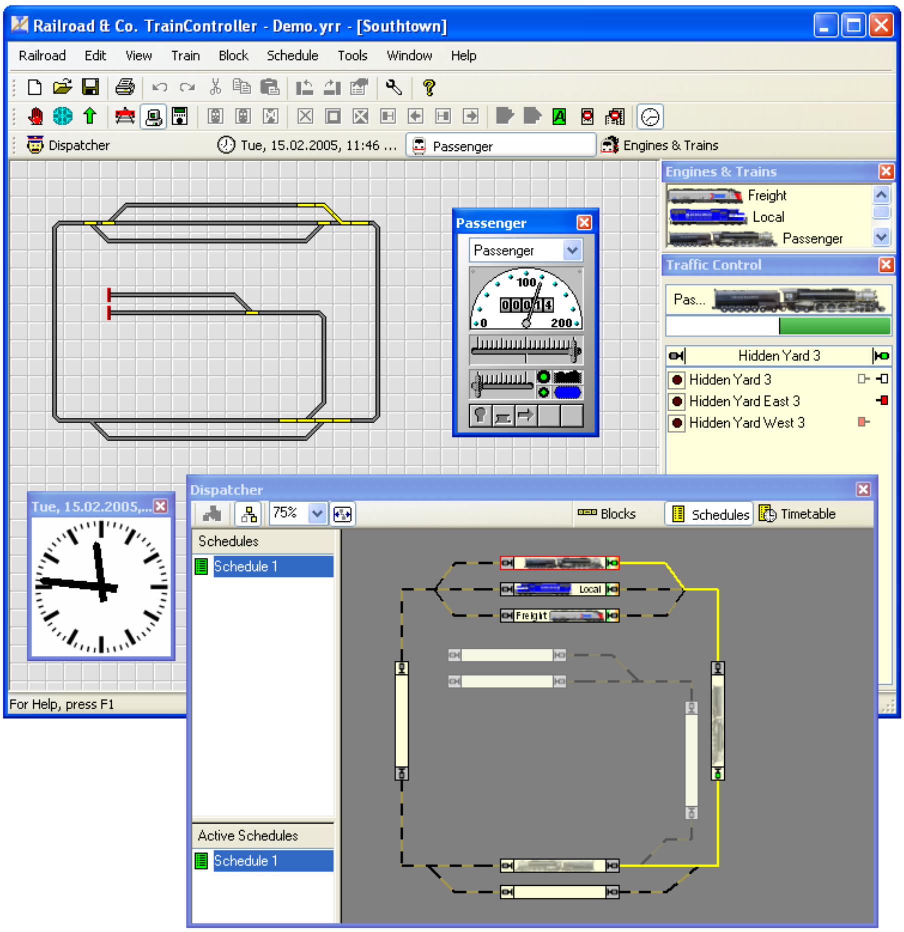


Diagram 34: RAILROAD & CO. TrainController™

Automatic Operation

Because you want to control your model railroad with your computer, you are probably interested in operating parts (or all) of your layout automatically. **TrainController™** does not require you to be an experienced programmer or computer expert in order to do this. For this reason, **TrainController™** does not require to learn a special railroad programming language with a new syntax. Automatic operation can be accomplished by a simple point and click to the objects which are to be operated or monitored. No abstract syntax must be learned. Configuration of automatic operation is as easy as drawing a track diagram.

The number, range and complexity of activities that can be managed by one person is extended substantially. A broad range of operating flexibility is provided that extends from a completely manual operation through a completely automatic operation (e.g. hidden yards control). Manual and automatic operations can be mixed simultaneously. This applies not only to trains on different areas of your railroad, but also to different trains on the same track, and even to the operation of a single train. The automatic processes are not bound to specific trains. Once specified, they can be performed by each of your trains. Timetable and randomizer functions increase the diversity of your model railroad traffic.

1.2 Fundamentals of Use

B

The Overall Principle

The concept of **TrainController™** is intended to support manual, semi-automatic and automatic operation of your model railroad as well as mixing of manual and automatic operation.

Switchboards, *Train* and *Turntable Windows* provide the controls to operate switches, signals, routes, trains and turntables, etc. These controls can be operated manually by the human operator or automatically by the computer.

A human operator is normally only able to operate one or two switchboards and at most two trains at the same time. If multiple control panels or a certain number of trains are to be operated at the same time, then either support of additional human operators is required, or a computer running **TrainController™**. The software contains a special component called the *Visual Dispatcher*, which is able to take the place of additional human operators.

Like a human operator the Visual Dispatcher is able to operate switches, signals, routes and trains. This is called *automatic operation*.

Manual and automatic operation can be mixed like several human operators can cooperate to control the same layout.

You can also decide to do without the *Visual Dispatcher*, if you want to control everything yourself.

File Handling

The complete data of your model railroad layout is stored in one single file on the hard disk on your computer. This file is called *layout file*. You can create as many layout files as you like. This is for example useful if you have different model railroad layouts or if you want to try and store several variants of the same layout file.

The layout file contains the complete description of your layout, i.e. all track diagrams, routes, trains and all data specified for automatic control of the layout, if any. Please note that all data of the same layout is to be stored in the same layout file.

Layout files are created, opened and stored through the **Railroad** menu of the software.

Whenever a session is terminated by closing a layout file or by terminating the complete software then an additional file is automatically created called *status file*. The status file contains the current status of your model railroad layout, i.e. the current state of all switches and signals, the status and positions of trains, or the current time of the clock, etc. The status file is loaded again when the program is started the next time. Using the content of the status file the software is able to start with the status of the model railroad valid at the end of the previous session.

Window Handling

Normally you will open several windows within the same layout file. If you want to split the track diagram of your layout into two or more switchboard windows or if you want to control different trains with different train windows then you can open additional windows within the same layout file.

Additional windows (switchboards, trains windows, clock, etc.) are created and destroyed through the **Window** menu of the software. Each window can be closed or made invisible, respectively, at any time without loss of data.

The main window always contains a switchboard. Each additional window can be docked to the frame of the main window. In this case it is always moved or resized with the main window. Alternatively it is possible to place each additional window independently from the main window at arbitrary positions on the computer screen.

Please note the difference between *windows* and *files*. Only one layout file can be opened at the same time and the layout file contains all data and windows that belong to the same layout. The windows belonging to the same layout are contained in one layout file.

Diagram 34 shows an open layout file that contains several windows. The file contains among others a switchboard window, two train windows, a clock window and a Dispatcher window for automatic operation.

Edit Mode

All changes to be made to the content of your layout file require that **TrainController™** is running in *edit mode*. While edit mode is turned on you can change data, add new data or delete data, that is no longer needed. During operation edit mode is turned off. This protects your data during operation against unintentional changes.

Edit mode can be turned on or off at any time. When edit mode is turned on all automatic operation of your layout, if any, is stopped, though.

! **In order to input new data as well as to edit or delete existing data edit mode must be turned on.**

Further Steps

In order to control your model railroad layout with **TrainController™,** you need one or more of the digital systems listed in the previous section. These digital systems are connected to an available serial port of your computer. It is also possible to connect the digital system to a USB interface of your computer by using an appropriate USB-Serial-Converter.

In the following it is assumed that you are already familiar with the usage of your digital system. For details regarding your digital system, please refer to the documentation provided by the manufacturer.

To create a computer control system with **TrainController**™ the following steps are usually performed:

- Creation of *Switchboards* containing control panels based on track diagrams of specific areas
- Entering the data of existing *engines* and *trains*
- Optionally drawing the *main block diagram* in the Dispatcher
- Creation of automatic *schedules* with the *Dispatcher*

It is not necessary to perform all steps listed above to control your model railroad with **TrainController**™ For model railroad clubs, it may be sufficient to arrange the *Switchboards* only. In this case, one person may be responsible for controlling the traffic by operating switches, signals and routes while other persons are using handheld throttles to control the trains. If you have an existing control panel, then you can use the *Train Windows*, independently, to take advantage of the realistic train control features of the program.

Switchboards

Usually, you will start configuring **TrainController**™ by creating one or more *Switchboards*. Like at real railroads, *Switchboards* are control panels to be used to control switches, signals, routes and other accessories like uncouplers or crossing gates. *Switchboards* are made using symbol elements representing *tracks, switches, crossings, signals, routes* and more.

Switchboards are usually created for those parts of the layout that contain switches, and signals. Examples of such areas are stations, sidings or hidden yards.

You first insert track elements into the *Switchboards* to create a track diagram that represents the track plan of your entire layout, the main station or any yard etc. For small and medium size layouts it is recommended to create only one switchboard, which is always located in the main window of the software. If desired, then this main switchboard can be used as a base for quick and easy setup of automatic operation. In the case of larger and more complex layouts you will probably create a separate *Switchboard* per yard. You can create as many additional switchboards as you like.

After you have placed all tracks, switches, crossings and bridges in the correct positions, you specify the *digital addresses* of your switches.

When this has been done, you are already able to control the switches of your model railroad with **TrainController™** and your computer.

Your model railroad may contain not only tracks and switches but also signals and other accessories. If so, the next step is placing the *signals* at the appropriate locations of your control panel. **TrainController™** provides symbols for *two*, *three* and *four aspect signals*. Uncouplers, lights, crossing gates or other accessories can be controlled with symbols representing *push buttons, toggle switches* or *on-off switches*.

After you have placed all the signals in the correct positions, you specify the *digital addresses* of your signals and other accessories.

Once you have specified the *digital addresses* of your signals and other accessories, you are able to control these objects manually with **TrainController™**, also.

Another possibility is to control your switches and signals by using *route elements*. Route elements are able to control groups of switches and signals. Additionally, route elements are locked until the specific route is released.

Text elements can be inserted at arbitrary positions to label your control panel. Images can be placed in the Switchboard as well.

If you want to quickly setup automatic operation for your trains or display train positions in the switchboard, then you will insert *traffic boxes* into your switchboard, that represent the blocks of your model railroad.

Switchboards provide even more features for controlling, monitoring and semi-automatic operation. These elements are discussed in more detail later.

Train Windows

The *train window* enables the operation of your *engines* and *trains*. To control several trains simultaneously, you can open as many train windows on your computer screen as desired.

After the selection of the current engine, or train, in the train window, you are able to control the train and monitor its operations with the control instruments.

As with real railroads, there is also a difference between *engines* and *trains* in **Train-Controller™** In order to be able to run your engines, it is sufficient to enter them as *engines* in **TrainController™** and to specify their *digital addresses*.

To operate a certain engine on your layout, just create a *Train Window* and specify the digital address of the engine. You do not have to bother with all other options until you want to add more realism to the operation of your trains.

Trains are used to obtain more realistic effects like *multiple unit* operation or in the consideration of car tonnage for speed calculation. A *train* represents a consist of one or more engines and a couple of cars. If a certain engine is sometimes running a light and fast passenger train, and at other times a heavy and slow freight train, then you can create different trains in order to reproduce the behavior of the engine in either situation.

The Visual Dispatcher

The *Visual Dispatcher* is a component that makes large scale railroad operations manageable by one person, matching operations found on the largest club layouts. *Engines* and *trains* can be operated manually or automatically.

Like a human operator must know the overall structure of the model railroad layout the *Visual Dispatcher* needs to know this, too. This structure is represented by a diagram that contains blocks and routes and the track connections between them. This diagram is called *main block diagram* of the layout. The main block diagram describes the track layout of your entire model railroad in rough outline.

The *Visual Dispatcher* manages traffic flow using a blocking system. Blocking ensures that trains do not collide and supports the tracking of train positions. For this purpose, the railroad layout is divided into virtual, logical blocks. That means, you define blocks at locations where traffic control will take place (e.g. scheduled stops in a station).

Usually each track in a station or hidden yard, each siding and appropriate sections of the connections between two yards will form a block.

Dividing the layout into logical blocks does not necessarily imply, that your blocks must be electrically insulated. **TrainController™** does not require such electrical insulation. It solely depends on the used hardware, whether your blocks must be insulated or not.

Blocks, *routes* and connecting *links* are arranged graphically in the *main block diagram* to specify on which paths trains will travel. *Schedules* describe train movements, i.e.

how trains travel. This includes start and destination blocks, scheduled waits, speed limits, etc.

AutoTrain™, an outstanding feature of **TrainController**™, allows you to start trains automatically without the need to define a schedule before or to create new *schedules* while playing with your trains – programming while playing!

Trains can run under full manual control, in which case the human operator will be responsible for obeying the block signals set by the *Dispatcher*; or under full control of the computer; or even with an intermediate level of automation.

For shunting special types of schedules are provided.

Schedules and *timetables* can be arranged with a broad range of flexibility. Since a timetable can be created for each day of the year, up to 365 timetables can be used.

Randomizer functions increase the diversity of your model railroad traffic.

2 The Switchboard

2.1 Introduction

B

TrainController™ displays a *switchboard* in the main window of the software. Additionally it is possible to display as many additional s*witchboards* on your computer screen as desired. A switchboard represents a track diagram control panel of specific parts of your model railroad, i.e. those parts that contain switches, and signals. Examples of such areas are stations, sidings or hidden yards. If you intend to control a large layout consider creation of a separate panel for each yard.

Switchboards are used to operate the *switches, signals, routes* and other *accessories,* like crossing gates, on your model railroad. Switchboards are created using different symbol elements that are arranged in rows and columns.

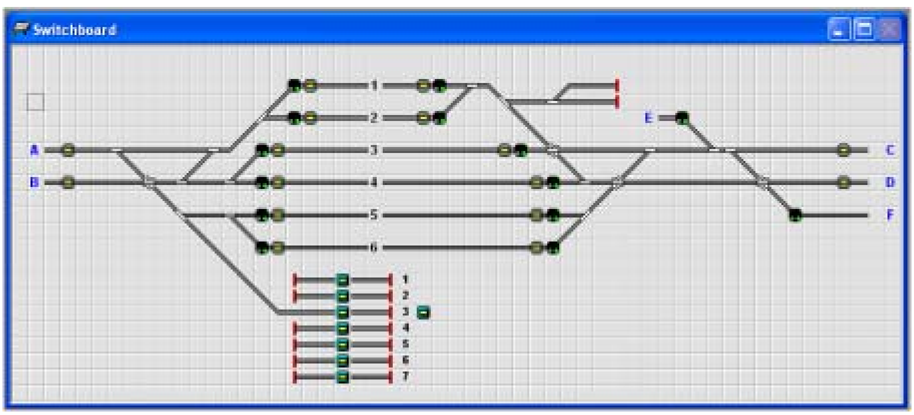


Diagram 35: Switchboard Example

Several types of symbol elements are provided to facilitate in the creation of switchboards:

- *Track elements* are used to represent the tracks of your model railroad such as *straight* and *curved* tracks.
- *Switch elements* are provided as special track elements to enable operation of different types of switches like *normal*, *triple* or *slip switches*.

- *Signal elements* are used as *two*, *three* or *four aspect signals* to represent and to operate the signals on your model railroad.
- *Accessory elements* of several types – *push buttons*, *toggle switches* or *on-off switches* – operate additional accessories such as uncouplers or lights or can be used to trigger other actions like playing of sound files.
- *Route elements* enable manual route operation and locking.
- *Traffic Boxes* can be used for quick setup of automatic operation and display of train positions.
- *Text elements* can be used as labels, e.g. for tracks in stations.
- *Images* can be inserted into your track diagrams to display landscapes, buildings, streets or other objects of your model railroad.

The following elements can be alternatively added to switchboards or to the *Visual Dispatcher*:

- *Indicator elements* are provided as *contact indicators* or as more intelligent *flagman indicators* to allow monitoring of the model railroad, creation of semi-automatic and automatic control mechanisms, or tracing of train positions.
- *Virtual Contact Indicators* can be used to reduce the number of track contacts which are needed for automatic operation.

The following steps are performed to create a full functioning switchboard:

- Drawing the track diagram of the related area
- Connecting switches and signals
- Placing signals and accessory elements
- Creating manual routes
- Inserting traffic boxes, if desired
- Adding text labels and images

The following steps are mainly performed in the switchboard in those cases where it is desired, to monitor the traffic on the layout to a certain degree, or to achieve semi-automatic operation of the layout without running the *Visual Dispatcher*. If the *Visual Dispatcher* is being used, you will likely perform the following steps in the *Visual Dispatche*r rather than a switchboard.

- Inserting contact indicators
- Arranging semi-automatic control mechanisms

2.2 Size and Appearance

For each switchboard it is possible to customize the size, i.e. the numbers of rows and columns, and the appearance individually.

The elements in the switchboard are arranged in a grid based system consisting of rows and columns (see Diagram 35).

The individual preferences with regard to the appearance of track diagram control panels are very different. For this reason **TrainController™** provides many options to customise the appearance of the control panels individually to your convenience and taste. There are options to select the background and track colours, to apply 3 dimensional light and shadow effects to background and tracks and to select the colours in which the states of certain elements are highlighted.

The possibilities are virtually not limited, a few examples are given below:

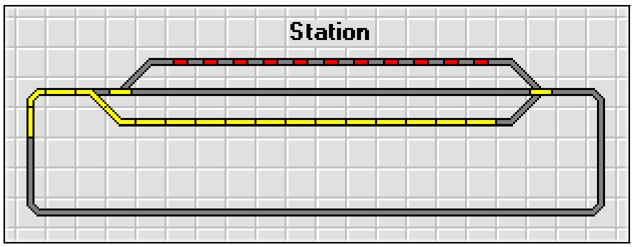


Diagram 36: Standard Format

Diagram 36 shows the standard format for display of control panels. A few examples of the unlimited possibilities to customize the appearance are given in the following:

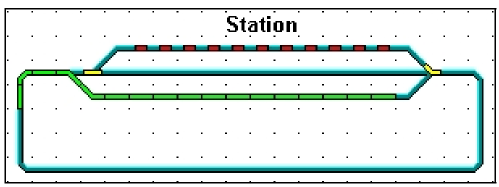


Diagram 37

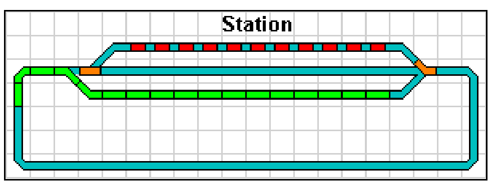


Diagram 38

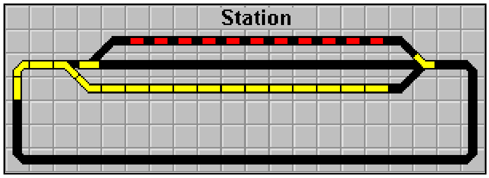


Diagram 39

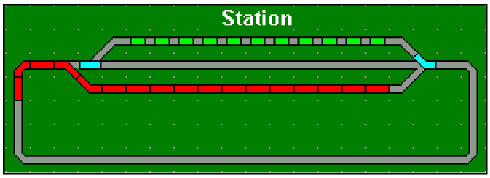


Diagram 40

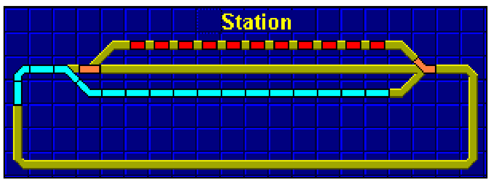


Diagram 41

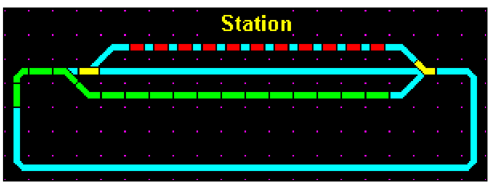


Diagram 42

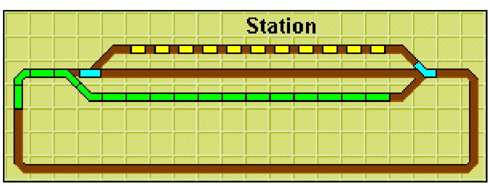


Diagram 43

2.3 Drawing the Track Diagram

B

Creating a *switchboard* starts with drawing the track diagram of the related station, yard or sidings. Using the available *track elements* a schematic image of the tracks of the area is drawn on the computer screen.

The following track elements are available:

- *Straight*
- *Normal* or *narrow Curve*
- *Bumper*
- *Diagonal* or *vertical crossing*
- *Diagonal* or *vertical bridge*
- *Turntable symbol* without electrical function
- *Left* or *right switch* as well as well as *Y-Switch*
- *Triple switch*
- *Single* or *double slip switch*

You can draw your track diagram in various ways. First, though, the edit mode of the switchboard must be turned on.

Then you have the following possibilities:

- **Inserting single elements:** You can draw your track diagram by inserting single elements successively.
- **Drawing a straight track section with the mouse:** You can draw a straight track section consisting of more than one element very quickly by dragging the section you want to draw with the mouse.
- **Drawing the track diagram with the keyboard:** An additional and fast way to draw the track diagram is the use of the numeric keypad of your computer.

These methods are explained in detail in the help menu.

To adjust the track elements precisely, additional edit facilities such as *copy*, *move* or *turning* of track elements are available.

2.4 Connecting the Switches

B

When the track diagram is completely drawn, the *digital address* of each switch, or slip switch, must be specified. This is the address of the stationery decoder or output device controlling the specified switch. If several *digital systems* are used, then the system that controls the particular switch must also be selected.

This is done by selecting the switch element and using the **Properties** command of the **Edit** menu.

For each switch you can specify a *name*. This is useful in identifying the switch when it is referred to later.

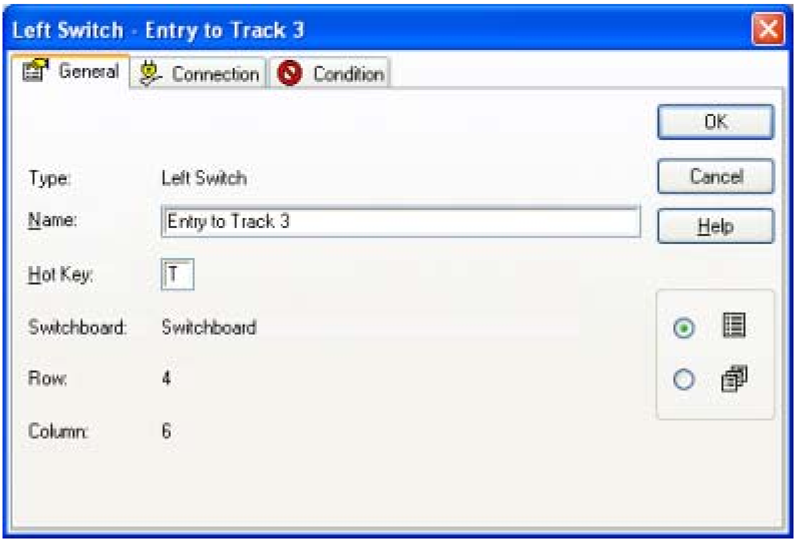


Diagram 44: Specifying the name of a switch

! Switches with more than two states such as *triple switches* or *single* or *double slip switches* with four solenoids, occupy two digital addresses. For simplicity, **TrainController™** always uses the <u>subsequent address</u> here, too.

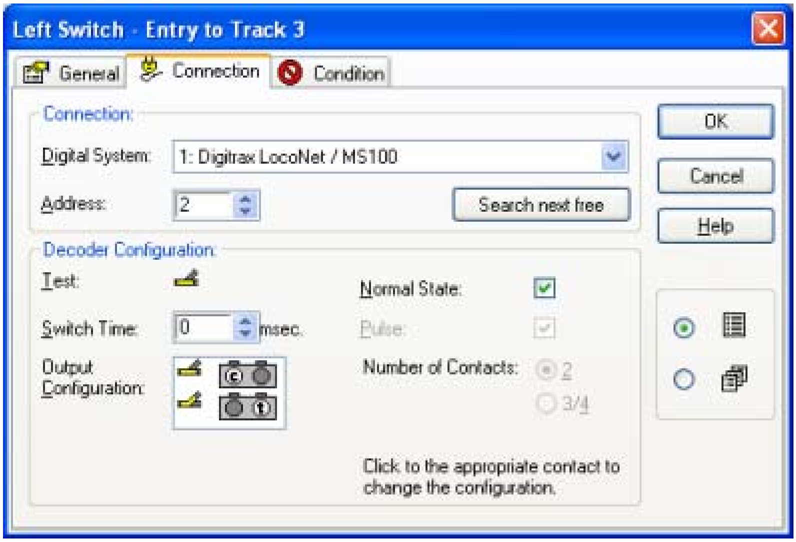


Diagram 45: Specifying the digital address of a switch

For *double slip switches* it is possible to specify whether the switch is operated by two our four solenoids.

Depending on the digital system being used, or the way the switch is wired, the switch element in the switchboard may not reflect the correct position of the real switch. To correct this problem, you are not required to rewire your switch. The software allows to setup the configuration of the decoder outputs in any way as required to operate the switch accordingly.

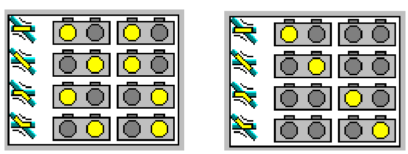


Diagram 46: Decoder Configurations for a double slip switch

The image above displays two possible configurations for a double slip switch. In both cases it is assumed that the switch is operated by two double-solenoid devices with four solenoids in total. For this reason the switch occupies four output contacts of an accessory decoder. The left image displays a situation, in which both double-solenoid devices must be operated in order to throw the switch. The right image displays a situation, in which only one double-solenoid device is to be operated to throw the switch.

The bright circles represent the contact outputs of the accessory decoder which are turned on in order to throw the switch to the corresponding state. The dark circles correspond to the decoder outputs, which remain turned off during operation of the switch.

For certain digital systems the bright circles are drawn in a colour or show an additional sign, which reflects the colour or label of the key, that is to be pressed on the keyboard or handheld of this digital system in order to activate the related contact output. If you are familiar how to operate a certain switch with your keyboard then these additional markings help you to map the keyboard operation to the correct configuration in the software.

These images display only two possible situations. The decoder outputs can be configured very flexible as required to operate the switch accordingly.

2.5 Signals and Accessories

After completing the track diagram, the next step is to place the signals in the diagram, as well as the accessory elements such as operating lights, uncouplers or other accessories.

The following elements are provided:

- *Two, three* and *four aspect signals* of different styles
- *Push buttons*, *toggle switches* or *on-off switches* to operate your accessories

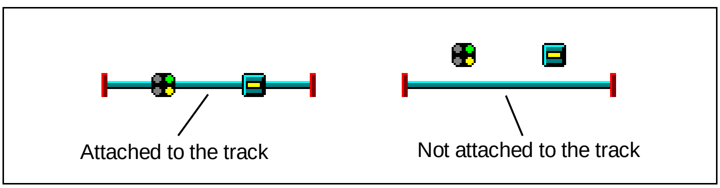


Diagram 47: Attaching signals and accessories to the track

If you want to visualize that a signal or accessory element located in the track diagram is associated with a piece of track on your railroad (for example a signal that controls a track section or a push button that operates an uncoupler), you can *attach* this element to a track element. For the operation of the signal or accessory element, however, it is not important if it is attached to the track or not. The purpose of the attachment is only to visualize the relation between the signal, or accessory element, and the corresponding track.

Signals

Signals are available in different styles. The styles are light or form signals as used by the Deutsche Bundesbahn or light signals of international railroad companies. Additionally, different styles are available for main and advance signals.

The purpose of these different styles is only the indication in the switchboard. For the operation of a signal, it is not important if you select an American or German light signal, or if you select a light or form signal. Feel free to use the style that best fits the respective signal on your model railroad.

It is important for the operation of the signal that you differentiate between the symbol for a *two*, *three* or *four aspect signal*.

For each signal a special style can be selected, that allows rotated display of the signal symbol. With another style it is possible to display multiple signals in adjacent switchboard cells as if the were tied to the same mast. With two four aspect signals combined in this way it is possible to display 16 different signal aspects.

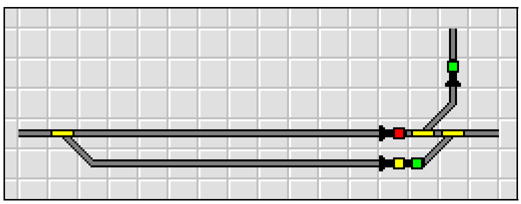


Diagram 48: Rotated and multiple signals

Diagram 48 shows signal symbols that are rotated according to the track elements they are attached to. There is also a multiple signal showing a green over a yellow light. This multiple signal is actually composed by two separate signal symbols. One of them uses a special style that let it look like mounted on top of the other signal at the same mast.

Accessories

Accessory elements are used to control accessories like uncouplers, light or crossing gates. They are available in three different types:

- *Push buttons* are used to turn on a certain contact temporarily – e.g. to control an *uncoupler*
- *Toggle switches* are used to change permanently between two related contacts
- *On-off switches* are used to turn on and off a certain contact permanently – e.g. to turn on and off lights

Push buttons and on-off switches not only can be used to operate a certain contact, but also to control other elements. It is possible, for example, to operate a group of related switches and signals with a single click to a push button. More details are outlined in section 11.3, “Operations”.

Connecting Signals and Accessories

Signals and accessories are connected to their real counterpart on the model railroad much like the switches as outlined in section 2.4, “Connecting the Switches”. This is also done by selecting the symbol of the signal, or accessory, in the switchboard and using the **Properties** command of the **Edit** menu.

For push buttons and on-off switches, which will be used to control other elements, a set of *operations* instead of a digital address needs to be specified. More details are outlined in section 11.3, “Operations”.

2.6 Routes

B

TrainController™ provides *route elements*, which are used to operate and lock the *tracks*, *switches* and *signals*, that belong to a certain *route*. Routes are represented in the switchboard by route symbols that are operated like an *on-off switch*. If the route is turned on, then all switches and signals of the route are operated. All track elements and signals along the path of the route remain locked in this position until the route element is turned off again. As long as these elements are locked, they cannot be operated or used by other routes.

Manual Routes vs. Automatic Routes

TrainController™ distinguishes between *manual routes* and *automatic routes*. Automatic routes can be operated automatically by the *Visual Dispatcher*. Manual routes can only be operated through their route symbol. They cannot be operated automatically by the *Visual Dispatcher*.

A manual route is created by inserting a route symbol into a switchboard at an arbitrary location. The location of the route symbol in a switchboard does not matter. Especially the location of the route symbol must not relate to the location of the tracks, switches and signals contained in this route. Manual routes are created, if the *Visual Dispatcher* is not being used at all or for those areas of your layout, which are only controlled manually with switchboards but not with the *Visual Dispatcher*.

An automatic route is created by inserting a route into the *main block diagram* of the *Visual Dispatcher* (for more details see page 98).

The location, where a route is inserted, determines, whether the route is manual or automatic. Please note the difference between route symbol and route in the switchboard. The route describes, which tracks, switches and signals are contained in a route. These elements are always located in switchboards. The route symbol represents a route.

You can safely create your first route symbols as manual routes in the switchboard in the first stage of your work with the program, when you want to learn, how routes work. If you decide later to operate a certain manual route automatically with the *Visual Dispatcher*, then you can move the route symbol to the *main block diagram* of the *Visual Dispatcher* with the mouse. This converts the manual route into an automatic route. Conversion to the opposite direction is not possible, though.

With the exception, that manual routes cannot be operated automatically by the *Visual Dispatcher*, there are no further differences between manual and automatic routes. The following sections therefore apply to both, manual and automatic routes.

Recording of Routes

The most important action when creating routes is recording the path of the route. This is done by selecting the route symbol and using the **Properties** command of the **Edit** menu. In the following dialog, select the tab labeled **Route** and then press the button labeled **Record**.

This procedure starts the *switchboard recorder* and the path of the route can be recorded. The running switchboard recorder shows the small control panel displayed below:

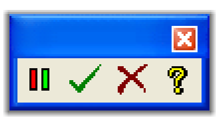


Diagram 49: The control panel of the switchboard recorder

The control panel contains four buttons with the following meaning (from left to right):

- **Break**: Recording is interrupted and no elements are recorded until this button is pressed once more

- **Stop with Save**: Recording is terminated and the recorded elements are stored.
- **Cancel**: Recording is terminated and the recorded elements are not stored.
- **Help**: Display help information about the recorder.

After starting the switchboard recorder, you are able to record the route. First select the switchboard in which the intended path of the route is located. Then, click to the track where the route will begin. Finally, click to the track element, where the route will end. **TrainController™** displays the tracks along the route as if the route were activated, but only if it is possible to reach the destination track from the starting track.

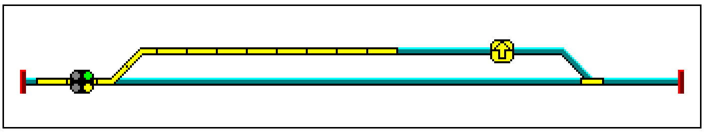


Diagram 50: Active route with switch and signal

If you specify the start and end of a route in this way, then **TrainController™** tries to find an arbitrary suitable path. Alternatively, you can also specify a path from the start to the destination of the route. To do this, move the mouse to the starting track. Press and hold the left mouse button and drag the mouse along the desired path to the destination of the route. After reaching the destination release the left mouse button. Again **TrainController™** indicates the tracks along the route as if the route were activated.

To extend an existing route, additionally press and hold the Shift key during the procedure outlined above.

Signals in Routes and Protection of Routes

If *signals* will be operated in addition to the switches along the route, then you can add the related signals to the *operations* of the route. More details about *operations* can be found in section 11.3, “Operations”. Signals included in these *operations* can be locked if desired until the specific route is turned off again.

In this way you can also *protect* the route. All switches outside the path of the route, which have been additionally assigned to the *operations* of the route, are operated accordingly and can be locked until the route is turned off. In this way you can lock switches outside of routes in appropriate positions to protect trains running on the route against collisions.

Operation of Routes with Start and Destination Keys

On control panels of real railroads, routes are often operated by first pressing a key near the starting point of the intended *route* and then pressing a key near its destination point. The operation of routes in this way can also be done with **TrainController**™. Usually it is sufficient to operate *routes* with a single click to the corresponding *route symbol*. It may be useful, however, to do this by selecting the start and destination of the route.

For this purpose it is possible to assign a start and a destination key to each route. This is done by selecting the symbol of the route and using the **Properties** command of the **Edit** menu. In the following dialog select the tab labelled **Start-Dest**. Here select the desired start and destination key.

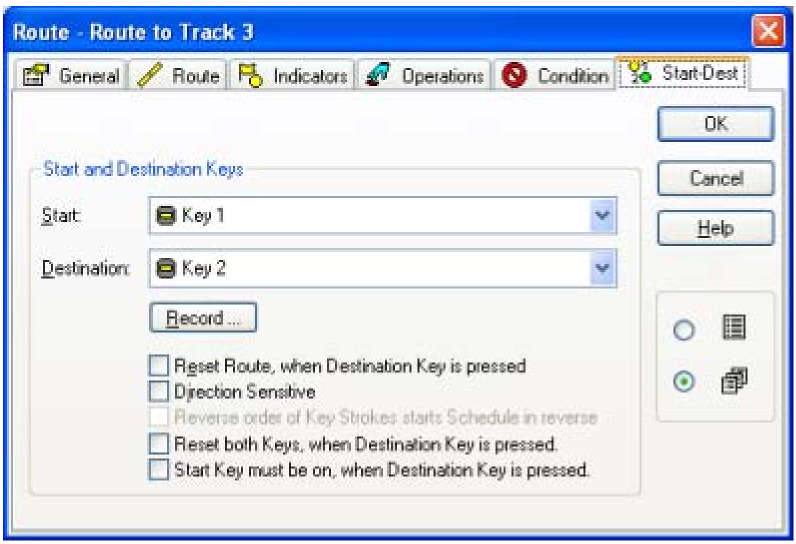


Diagram 51: Assigning start and destination key to a route

- It is possible to select *Push buttons*, *On-off switches* and *Contact Indicators* (see section 4, “Contact Indicators”) as start and destination keys. Especially with contact indicators it is possible to operate routes with start and destination keys in an external control panel (see section 11.6, “External Control Panels”).
- Several options are provided to adjust the operation with start and destination keys to your needs. For example it is possible to specify that the route is released when the destination key is hit again after activation of the route. It is also possible to specify that the start key must be pressed until the destination key is pressed.

Interlinking Routes

The path of a route, i.e. the tracks and switches recorded according to page 61, must be located completely in one switchboard. Nevertheless, it is sometimes useful to create routes across switchboard boundaries. In these situations it is possible to interlink several routes.

Here are different methods to interlink routes:

- **Using routes in routes:** Assign the route symbols that are to be interlinked as *operations* to a single route symbol. The single route symbol is exclusively created for this purpose (here called “main route”). If the releasing of the sub routes is additionally assigned to the “off” state of this main route, then you are able to activate and lock, or to release all sub routes, by simple operation of the main route. The sub routes can be locked with the main route, if desired. In this case the sub routes cannot be released while the main route is active.

- **Routes as part of schedules:** The recommended way for interlinking and operation of routes for automatically running trains is to include the particular routes into a *schedule.* For more details about *schedules* refer to section 5.10, “Schedules”.

2.7 Text Labels

You can place text labels in your control panels. For this purpose, t*ext elements* of the following types are provided and can be used to label switches, signals or tracks.

2.8 Images

It is possible to insert images from bitmap files into your switchboard. The following possibilities are provided:

Images can be arranged in the background, i.e. behind the track diagram, or in the foreground of the switchboard. Images in the background can be covered by track elements or by images laying in the foreground. Such images can be used to display landscape structures like meadows or rivers. Images in the foreground can cover track elements and can be used to display buildings, bridges or tunnels.

It is additionally possible to fade out portions of an image, i.e. to draw portions “transparently”. This is useful if images with irregular shapes are drawn. This is done by drawing the parts of the image, which shall be drawn transparently, with a certain color, which is not used elsewhere in the image.

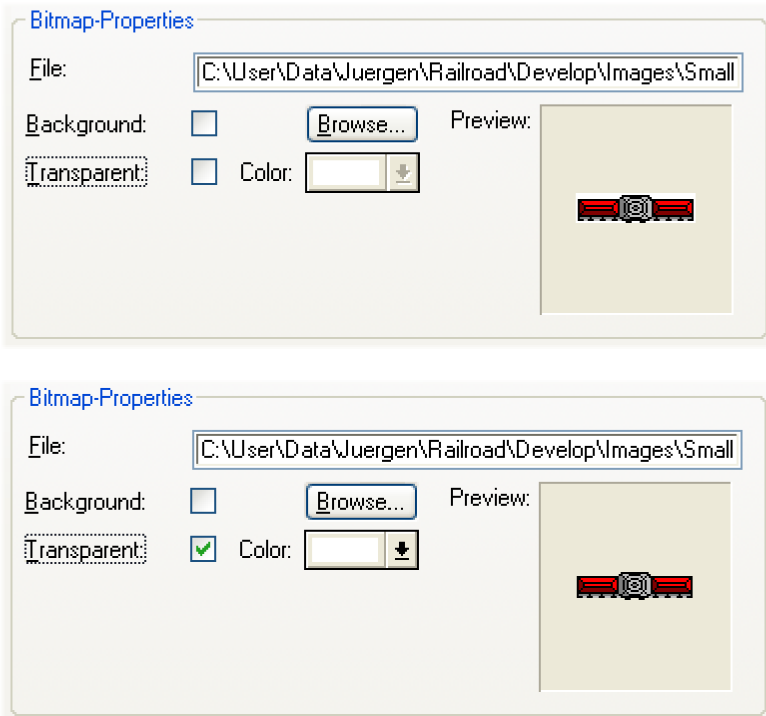


Diagram 52: Arranging an image

In the first example displayed above the image is not drawn transparently. All parts of the image are visible in the preview. In the second example the white parts of the image are drawn transparently and remain invisible.

2.9 Highlighting occupied track sections

It is possible to assign a set of track elements to each indicator (see chapter 11, " Indicators and Semi-Automatic Control"). These track elements are highlighted, when the indicator is turned on. In this way it is for example possible to highlight occupied track sections in the switchboard.

If there is a known engine or train located at the indicator, when the indicator is turned on, then the color of this engine or train is used for highlighting. Otherwise the track elements are highlighted in the same color as the indicator.

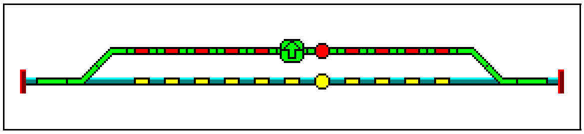


Diagram 53: Highlighting occupied track sections

In the image displayed above both tracks of the station are occupied. They are highlighted in the same color as the corresponding contact indicators. The route to track 1 is activated. Route and highlighted tracks are visible at the same time.

Track highlighting can not only be realized with contact indicators, but also with other types of indicators, which are described in other sections of this Users Guide (for example *flagman indicators* or *virtual contact indicators*).

2.10 Displaying Train Names and Symbols in the Switchboard

The names or symbols of trains located in a certain block can be displayed in the switchboard with so called *traffic boxes*. These are elements, that are associated with blocks. *Traffic boxes* are able to show the status of the related block as well as an image and/or the name of the train that is currently located in the block, if any. For further details please refer to section 5.5, “Train Detection and Train Tracking”.

Traffic Boxes can be used, too, for quick and easy setup of automatic operation of your trains. These boxes mark the location of the blocks of your layout in the track diagram.

2.11 Using the Computer Keyboard as a Control Panel

For the convenient operation of *switches, signals, accessories* and *routes,* it is possible to specify a *hot key* on your computer keyboard. A hot key is one of the keys A to Z or 0 to 9. An element, to which a hot key has been assigned, can be operated conveniently by pressing its hot key.

In Diagram 44, page 56, the hot key “T” is assigned to a switch. That means pressing the key “T” on the computer key board operates the switch.

3 Train Control

3.1 Introduction

The *Train List* and *the Train Windows* provided by **TrainController™** are used to manage and operate your engines and trains.

The *Train List* holds all engines and trains defined in the software.

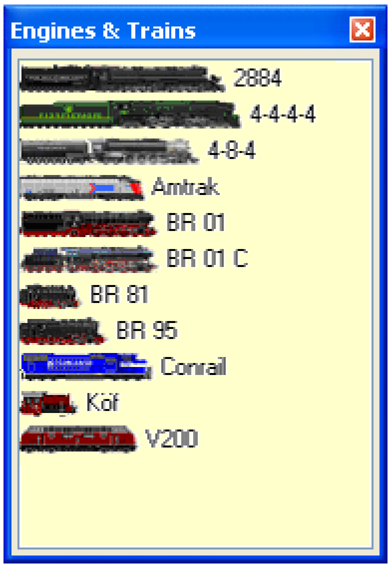


Diagram 54: Sample Train List

Each item in this list shows the name and the image of the train. To prepare train images for display in **TrainController™** a complementary software program called **TrainAnimator™** is available free of charge.

TrainController™ expects the image data to be stored in a certain format and scaled to a certain size. The images must fit to the proportions of the screen display of **TrainController™**. Additionally the images of several trains should fit to each other with regard to their scale, regardless of the origin of each image. **TrainAnimator™** is able to process several image formats, among others bitmap, JPEG or GIF. It is also able to extract images, that are stored in application programs or screen savers. **TrainAnimator™** converts the different data formats and image sizes to a standardized and scaled format, which can be used by **TrainController™** without further conversion.

The images displayed in Diagram 54 have been processed with **TrainAnimator™**, too. Even though the original formats and sizes of the particular images displayed above are very different, they have been converted and scaled to fit to each other.

In the *train list* each engine or train can be selected to change its properties or to operate it. If, for example, a double-click is performed to a train in this list during operation of the layout, then a *Train Window* is being activated with which the train can be operated.

A sample *Train Window* is displayed below:

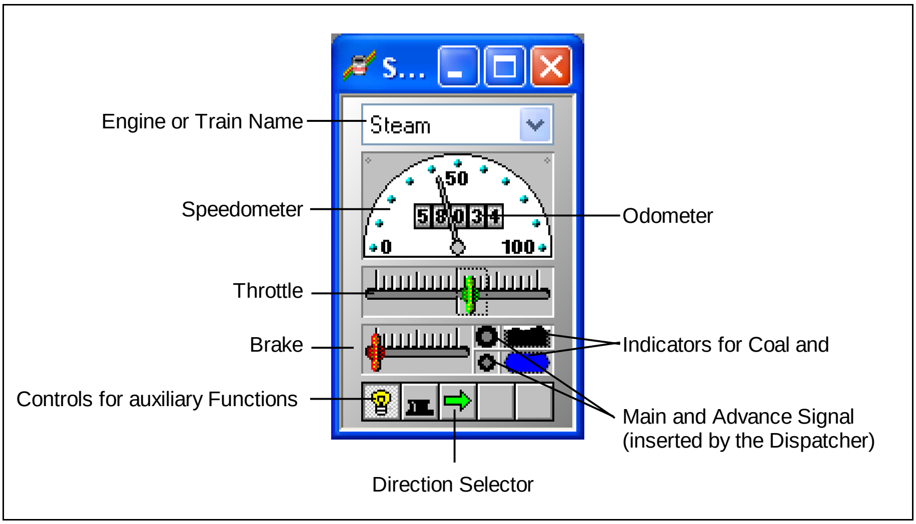


Diagram 55: Train Window

!

All engines and trains defined in the software are listed in all Train Windows, too. If a train is created in a certain Train Window, then this train is listed in all other Train Windows, too. It does not matter in which Train Window the properties of a train are changed or with which Train Window a train is operated. The changes of the properties or of the status are reflected in all other Train Windows also.

The data of engines and trains can be exported to a separate file and imported into another **TrainController™** project. In this way it is possible to exchange train data between different layouts or to import train data created at home into a project file belonging to a club layout.

3.2 Engines and Trains

B

Like at real railroads there is also a difference between *engines* and *trains* in **TrainController™** An *engine* describes different properties of one of your model engines. These are prototypical attributes like maximum speed or horse power, or model related properties like digital address or auxiliary functions. A *train* is powered by one or more *engines* and is used to simulate prototypical aspects such as multiple unit operation or simulation of train tonnage.

For simple operation of your engines it is sufficient to enter each engine with its *digital address* in **TrainController™** To specify the digital address or other attributes mark the appropriate engine in the Train List or in a Train Window and select the **Properties** command of the **Edit** menu. Once an engine is entered with its digital address it is then possible to control it with the Train Window.

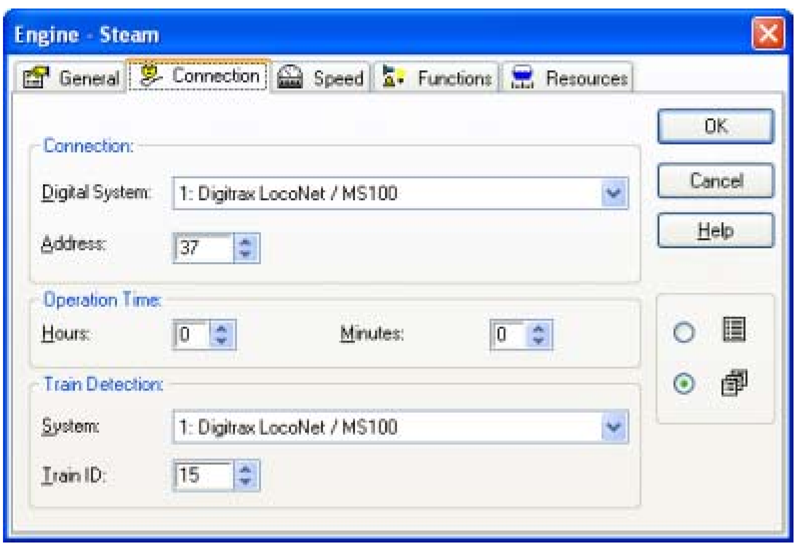


Diagram 56: Digital Address of an Engine

For each engine you can specify its *type*. This describes, how the engine is powered - *steam, diesel* or *electric*. The type is used to simulate the consumption of resources such as coal or diesel.

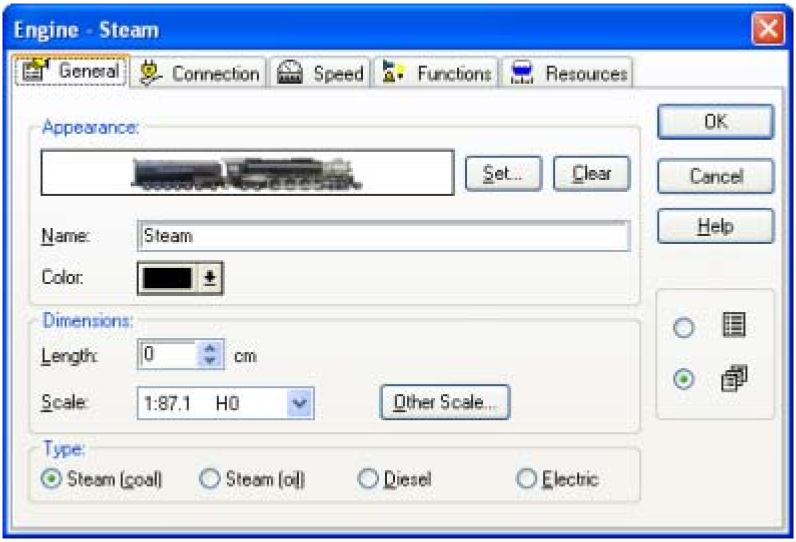


Diagram 57: General Properties of an Engine

Trains are used to obtain more realistic effects like *multiple unit* operation or in the consideration of car tonnage for speed calculation. A *train* represents a consist of one or more engines and a couple of cars. If a certain engine is sometimes running a light and fast passenger train, and at other times a heavy and slow freight train, then you can create different trains in order to reproduce the behavior of the engine in either situation.

Like in real railroads an *engine* can only run with <u>one</u> *train*. Internally **TrainController™** uses a smart “coupling” mechanism. When a train is started, all engines assigned to this train are assumed to be coupled to this train. As long as the train is running these engines cannot be operated individually or with other trains. When the train stops, the affected engines are assumed to be uncoupled. They are then available to be operated individually or with other trains. This internal “coupling” and “uncoupling” is done automatically. The software requires no manual intervention. However the real coupling and uncoupling of the engines on the model railroad is still to be done by the operator.

3.3 Throttle and Brake

B

The *throttle* is used to control the speed of each *engine* or *train*. The zero position of the throttle is located in the middle. When the slider of the throttle is in the rightmost position, the train is running forward with maximum speed. Conversely the maximum backward speed is achieved by pulling the slider to the leftmost position.

An additional instrument to control the speed of a train is the *brake*. Pulling the slider of the brake decelerates the train. The brake is an additional aid. For simplicity it is possible to control the speed with the *throttle* only foregoing the *brake*.

For each *engine* you can specify the *maximum scale speed*. This value is used as the maximum speed with which a train is controlled by **TrainController**™ To run an engine with maximum speed the throttle slider must be pulled to the rightmost or leftmost position.

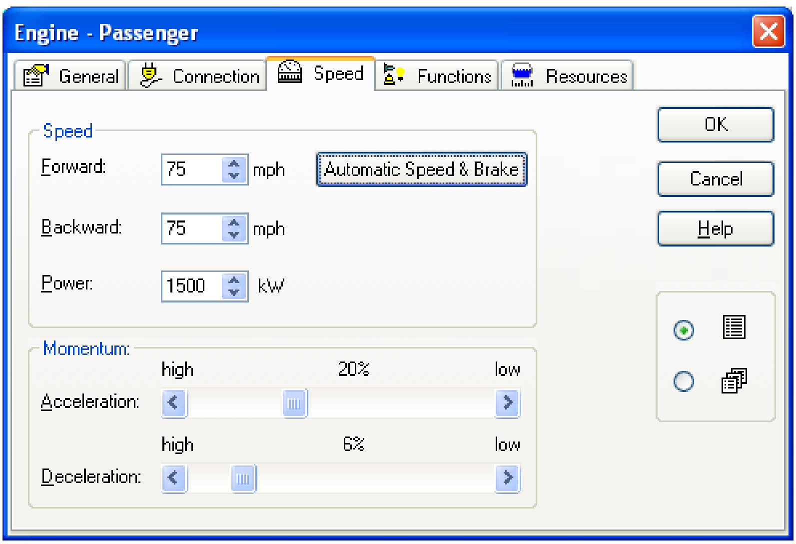


Diagram 58: Speed Properties of an Engine

For each *engine* you can also specify the *threshold speed*. This is the minimum speed at which the engine runs smoothly. The threshold speed is used if the throttle slider is moved out of the zero position. In this way “dead zones” near the zero position of the slider are avoided. For engines which will run automatically under control of the *Dispatcher* (see chapter 5, “The Visual Dispatcher”) it is recommended to adjust the threshold speed accordingly.

3.4 Speedometer and Odometer

The *speedometer* displays the current *scale speed* of an engine or train. The scale speed is calculated using the real speed on the model railroad layout and the scale of the model. If a train with scale 1:87 (H0) is running with a real speed of 1 mile per hour on the model railroad layout, then this speed corresponds to a scale speed of 87 miles per hour.

In conjunction with the scale factor of the *Clock* (see chapter 10, ” The Clock”) the *simulated distance* is calculated. If the scale factor of the clock is 12, then the duration of one “simulated hour” is 5 real minutes. Our train, which runs with a scale speed of 87 miles per hour passes a distance of 87 simulated miles within 5 real minutes. This simulated mileage is displayed on the *odometer*.

In this way it is possible to simulate very large distances which actually do not exist on your model railroad. Our train, which runs with a real speed of 1 mile per hour passes 87 simulated miles in 5 real minutes or around 1000 simulated miles in one real hour, respectively. This is a scale of 1 to 1000!

3.5 The Speed Profile

To enable the program to display the correct *scale speed* on the speedometer and to perform speed calculations correctly it is recommended to adjust the *speed profile* for each engine.

The speed profile is a table that records which *virtual speed step* corresponds to which *scale speed*. **TrainController™** internally works with 1000 virtual speed steps for each direction regardless of the characteristics of the used engine decoder. When a speed command is sent to the decoder, then the virtual speed step is matched to the appropriate physical speed step of the decoder.

Preparing the decoder

B

Before adjusting the speed profile the decoder of the locomotive, if any, should be prepared accordingly. This should be done to achieve the best possible operation. Perform the following steps prior to adjustment of the speed profile:

- Set the start voltage to a value, at which the locomotive begins to run smoothly.

- Adjust the maximum speed of the decoder in a way, that the desired maximum scale speed of the locomotive corresponds to the highest speed step of the decoder. If, for example, your decoder has 28 speed steps and the maximum scale speed of the locomotive should be 100 mph, then adjust the maximum speed of the decoder in a way, that the locomotive runs about 100 mph at speed step 28.
- Set the deceleration momentum of the decoder to a minimum value. This is just the value, at which now abrupt speed change of the real locomotive can be noticed anymore, when changing from one speed step to another.
- Adjust the speed table or the mid voltage of the decoder, if any, and its acceleration momentum to any settings, that you feel convenient with.

!

Note, that the speed profile must be adjusted again, whenever you changed the maximum speed, the deceleration momentum, the start or mid voltage or the speed table of the decoder.

The simplified Profile

B

The software offers two sets of options to adjust the speed profile of each engine. The first set allows editing of a simplified profile. This simplified profile describes the speed characteristics of your engine very roughly only and with identical settings for both directions of travel. It contains the following entries:

- An entry, that describes the threshold speed. This is the minimum virtual speed step (out of 1000) at which the engine begins to run smoothly. This value is adjusted by letting the engine run as slow as possible, but also smoothly. If this is achieved the current speed value is stored into the software.
- An entry describing the speed step, that corresponds to a pre-set slow speed. Let the engine run at this speed (e.g. by measuring the speed with a stop watch) and store this value into the software.
- An entry describing the speed step, that corresponds to the maximum speed of the engine. This value is determined and stored in the same way as the other two values.
- An entry describing the braking ramp, that is effective, when the engine is stopped during automatic operation. If the engine is decelerating too slowly or stopping too late during automatic operation, then this value can be easily adjusted.

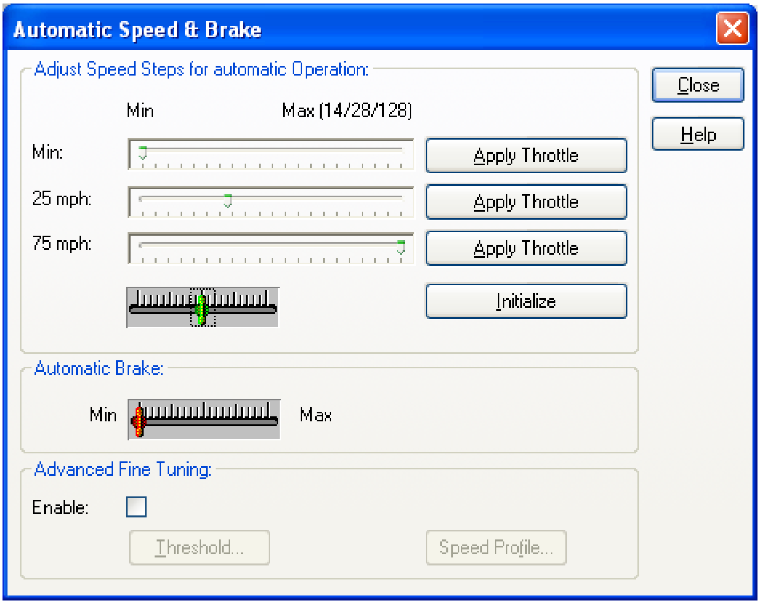


Diagram 59: Adjusting the simplified Profile

This simplified profile describes the speed characteristics of your engine very roughly. This is sufficient for manually operated engines or in many cases during automatic operation, too, if only real stop indicators are being used.

Advanced users, who want to use combined brake/stop indicators (see page 121) or Virtual stop contacts (see section 12.3), should fine tune the speed profiles of their engines as outlined in the following.

! **The settings of the simplified speed profile and the advanced fine tuning affect each other. For this reason it is only possible, to enter options either for the simplified profile or for advanced tuning but not both.**

Advanced Fine Tuning of the Speed Profile

X The advanced speed profile is created by measuring the time needed by the affected engine to pass a certain track section. The *scale speed* is calculated using the length of the section and the scale of the model.

For each direction the speed profile contains at most 15 entries for 15 virtual speed steps of all in all 1000. Intermediate values are calculated accordingly. In this way it is possible to calculate the scale speed for each of the 1000 virtual speed steps.

The entries of the speed profile are distributed equally in the range of the available virtual speed steps. There is no coherence between the number of entries and the number of speed steps of the decoder.

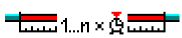
There are five procedures to perform the measurement:

Manual measurement of one single speed (stop watch)

Automatic measurement of one single speed step by running the engine from one momentary track contact to another

Automatic measurement of one single speed step by running the engine on a track section with three occupancy sensors

Automatic measurement of the complete speed profile with momentary track contacts



Automatic measurement of the complete speed profile with three occupancy sensors

You can measure single values of the speed profile manually like using a stop watch. However **TrainController™** provides the possibility to measure all relevant values between the *threshold speed* and the *maximum speed* automatically. For this purpose you have to arrange a track section, which is either limited by a momentary track sensor on each side, or which is monitored by track occupation sensors. To each track sensor a *contact indicator* (see section 4, “Contact Indicators”) must be assigned. To measure the speed profile the engine is run back and forth automatically by **TrainController™** The program starts the measurement with the *threshold speed*. Each time the engine has passed the track section in both directions the engine is accelerated to measure the next speed level. This is repeated until the engine reaches its maximum speed. Monitoring the *contact indicators* **TrainController™** is able to determine, when the engine enters or leaves the track section used for this measurement.

Finally an additional run into each direction is performed to determine the behaviour of the engine when decelerated.

The different methods to perform an automatic measurement with momentary track contacts or occupancy sensors are outlined in the following. Further details about the different types of sensors and their usage can be found in section 5.8, “Arranging Indicators in a Block”.

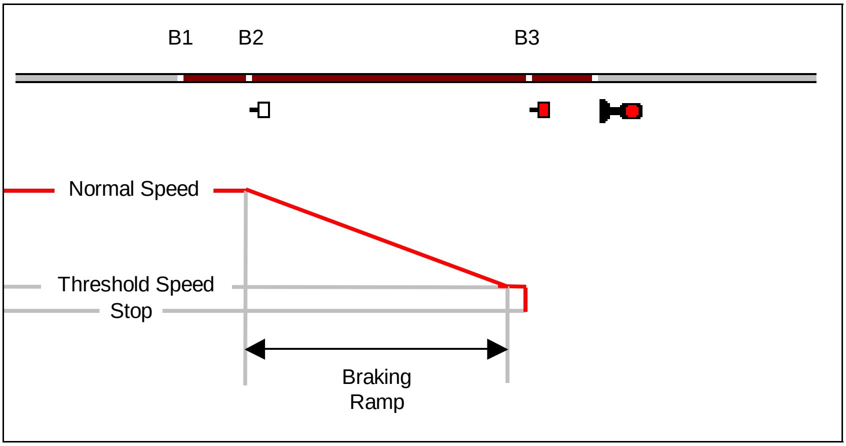


Diagram 60: How Brake and Stop Indicators work – Occupancy Sensors

Measuring with Momentary Track Contacts

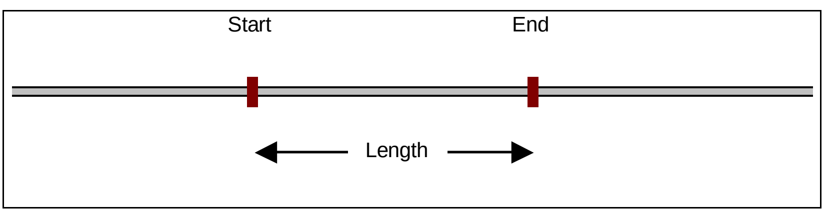


Diagram 61: Measurement with Momentary Track Contacts

For the measurement with momentary contacts two momentary contacts are needed. These contacts are associated with two contact indicators, called “Start” and “End”. The length of the track section used for the measurement is determined by the distance between the two track contacts.

To start the measurement put the engine on the track in a certain distance left of indicator “Start” heading to indicator “Start”. The engine will be started in forward direction. When it reaches the indicator “Start” the measurement of the current speed step begins. When the engine reaches the indicator “End”, then the engine is decelerated and stopped. Now the engine reverses the direction and the measurement of the same speed step in backward direction is performed, now using indicator “End” for the beginning of the measurement and indicator “Start” for the termination. After reaching indicator “Start” the engine is decelerated and stopped and the measurement is repeated for the next speed step in both directions.

The whole procedure is repeated until the speed step that corresponds to the specified maximum speed has been measured.

! **Please ensure, that both indicators are turned off whenever the engine reverses the direction between two runs of the procedure. There is an additional option, with which the run-out can be adjusted. If an indicator is not turned off when the engine reverses the direction then increase the run-out.**

Measuring with Occupancy Sensors

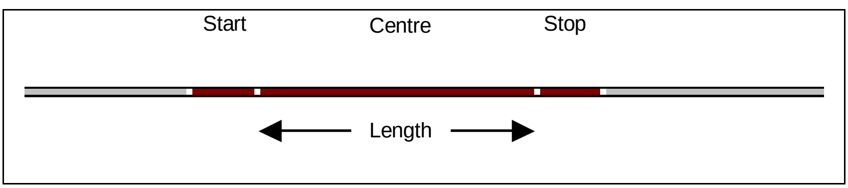


Diagram 62: Measurement with Occupancy Sensors

For the measurement with occupancy sensors three occupancy sensors are needed. These sensors are associated with three contact indicators, called “Start”, “Centre” and “End”. The length of the track section used for the measurement is determined by the length of the occupancy section associated with “Centre”. The length of the other occupancy sections does not matter.

To start the measurement put the engine on the track in a certain distance left of section “Centre” heading to section “Centre”. The engine will be started in forward direction. When it reaches the section “Centre” the measurement of the current speed step begins.

When the engine reaches the section “End”, then the engine is decelerated and stopped. Now the engine reverses the direction and the measurement of the same speed step in backward direction is performed, now using indicator “Centre” for the beginning of the measurement and indicator “Start” for the termination. After reaching indicator “Start” the engine is decelerated and stopped and the measurement is repeated for the next speed step in both directions.

The whole procedure is repeated until the speed step that corresponds to the specified maximum speed has been measured.

! **There must not be any “dead gap” between the occupancy sections. That means the track sections must be located close together. Track section “Centre” must begin where the other track sections end and vice versa.**

! **Please ensure, that the indicator associated with “Centre” is turned off whenever the engine reverses the direction between two runs of the procedure. There is an additional option, with which the run-out can be adjusted. If indicator “Centre” is not turned off when the engine reverses the direction then increase the run-out.**

It does not matter, though, whether the indicator, that is associated with the track section where the engine just reverses its direction, is turned on or off in the moment, when the direction is reversed.

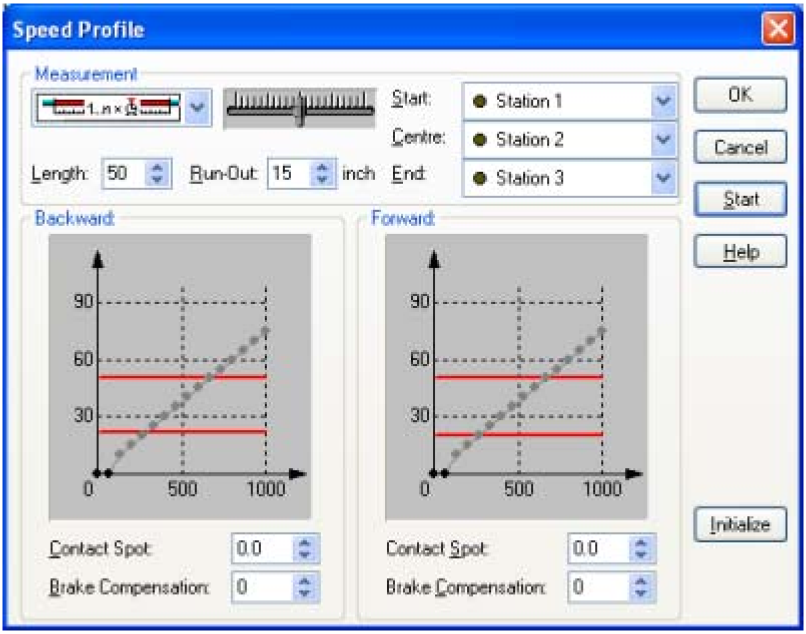


Diagram 63: Measuring the Speed Profile with occupancy sensors

The speed profile can also be viewed and edited graphically.

After running back and forth the engine on the measured section an additional measurement is performed to analyse the braking behaviour of the engine. This analysis leads to a value called *brake compensation*. This is used to compensate additional deceleration delays - e.g. caused by the engine decoder - when an engine is decelerated. If this engine tends to exceed pre-set braking ramps or stop distances when it is slowed down, then increase this value. The default value is 0, which means, that no compensation is performed. Please note: this option has only an effect in conjunction with combined brake/stop indicators or Virtual Contacts and only when engines are decelerated before reaching their location.

Measuring the speed profile is especially important for all engines, which will run under control of the *Dispatcher* (see chapter 5, “The Visual Dispatcher”). The *Dispatcher* uses the scale speed to control the engines. In this way engines with different characteristics pass the same track sections with identical speed, if the speed profile of each engine is adjusted accordingly.

3.6 Headlights, Steam and Whistle

For each engine up to nine auxiliary engine functions (e.g. light, sound, smoke, etc.) can be defined. Each function can perform one of the following:

- activating a built-in function of an engine decoder
- executing a *macro* (see section 11.5, “Macros”)
- playing a sound file

Engine Functions are executed

- manually by using the auxiliary function controls of the train window
- by macros (see section 11.5, “Macros”)
- when a schedule is executed (see section 5.10, “Schedules”)

If engine functions are executed by *macros* or *schedules*, then the particular function is identified by its symbol (e.g. *Light*, *Smoke*, etc.). If for a example a certain schedule has to perform the function symbol *Whistle*, then the function is executed, which is assigned to the engine as *Whistle*. If no such function symbol is assigned, then nothing happens. Some functions are listed more than once (e.g. *Sound 1, Sound 2*, ...). This gives you the opportunity to assign several functions with similar meaning. If for example a certain schedule has to perform *Sound 3,* then a sound is played only if the affected engine has a function *Sound 3* assigned to it.

You can specify an individual *tip* for each function. This is arbitrary text which is displayed in a small popup window, when the mouse is moved over a function button in the *Train Window*. This tip text helps to distinguish between different functions that are associated with similar function symbols (such as *Light 2*, *Light 3* , …).

The actually executed function may be different from engine to engine. This is illustrated by the following example. It is assumed, that a built-in sound function of the corresponding engine decoder is assigned as *Sound 1* to a diesel engine and playing a sound file with a typical sound of a steam engine is assigned as *Sound 1* to a steam engine. If the function *Sound 1* is executed during automatic operation, then the built-in decoder function is activated for the diesel engine and the specified sound file is played for the steam engine.

Each function, which is assigned to a built-in function of an engine decoder can be turned on permanently (e.g. *Light* or *Steam*) or temporarily (e.g. *Whistle* or *Coupler*). For this reason the auxiliary function controls in the train window can be arranged as on/off switches or push buttons.

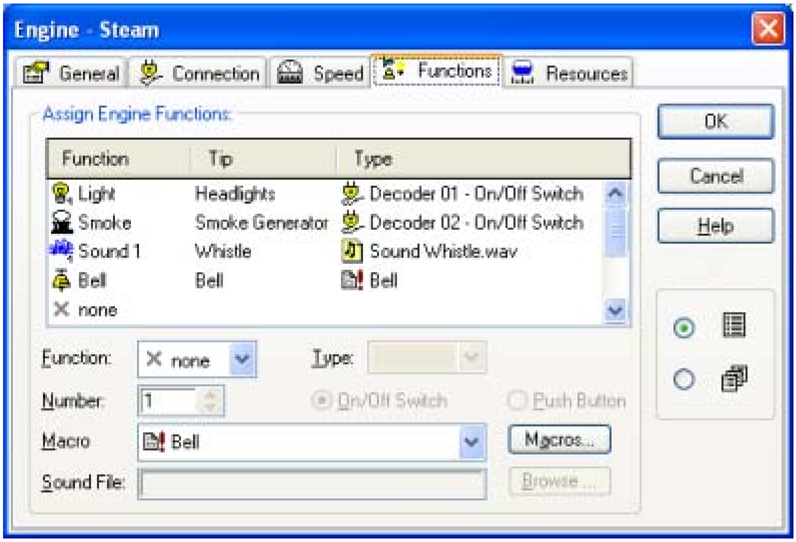


Diagram 64: Arranging Auxiliary Functions

3.7 Multiple Units

TrainController™ supports operation of trains coupled as *multiple units.* To create a multiple unit perform the following steps: first create a new *train*. Then mark this train in the *Train List* or in a *Train Window* and select the **Properties** command of the **Edit** menu. Now select the tab labeled **Engines** and assign the desired engines to this train.

If some of the engines are not headed to the same direction as the first engine of the train then select the option **Reverse** for the affected engines.

When a train is running as a multiple unit then the state of the first engine of the train is displayed in the *Train Window*.

The engines assigned to a multiple unit may have different speed characteristics, i.e. they may run with different speed at the same speed step. However, if the *speed profile* of each affected engine is adjusted accordingly, then **TrainController**™ is able to balance out the different behaviour of the engines.

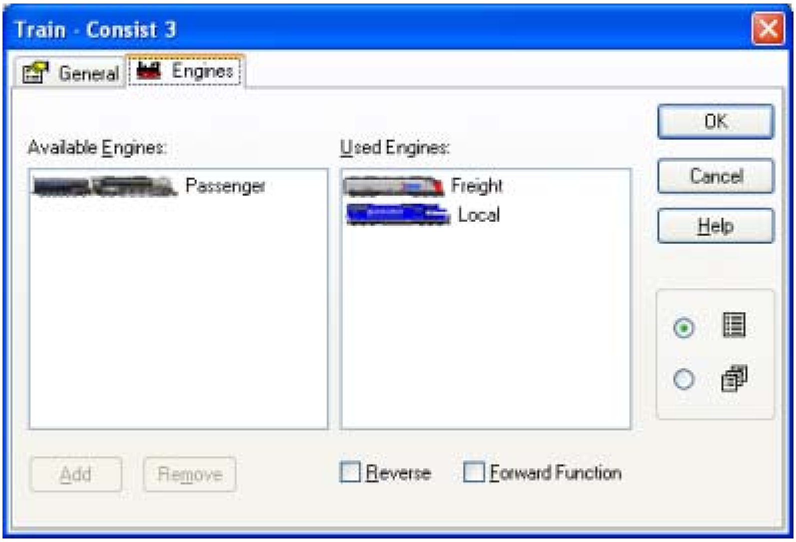


Diagram 65: Creating a Multiple Unit

Like in real railroads an *engine* can only run with one *multiple unit.* Internally **Train-Controller™** uses a smart "coupling" mechanism. When a multiple unit is started, then all engines assigned to this multiple unit are assumed to be coupled to this train. As long as the multiple unit is running these engines cannot be operated individually or with other trains. When the multiple unit stops, then the affected engines are assumed to be uncoupled. They are then available to be operated individually or with other trains. This internal "coupling" and "uncoupling" is done automatically. The software requires no manual intervention, however the real coupling and uncoupling of the engines on the model railroad is still done by the operator.

When a train is selected in the *Train Window* then the window shows the state of the first engine assigned to the train. Especially the function buttons specified for the first engine are visible. If you want to control the auxiliary functions of the second or any other engine of the train manually, select this engine in the *Train Window* and use the function buttons of this engine instead.

Automatic operation of auxiliary functions of a multiple unit are normally performed by the first engine only. Please check the option **Forward Function**, if orders to operate a specific engine function shall be performed by the other engines of the multiple unit, too.

Operation of Additional Function Only Decoders

Function only decoders are often used to add additional functions to a decoder controlled locomotive or to other rolling stock. An example is lighting in passenger cars. These decoders can be controlled with **TrainController™**, too.

This is done by setting up a “dummy engine” with the digital address of the function only decoder. The speed settings of this decoder don't matter in this case. The function setup of this decoder is done as outlined in section 3.6, “Headlights, Steam and Whistle”.

Manual operation of the extra functions provided by the function only decoder is done by selecting the “dummy engine” in the *Train Window* and operating the function buttons of this engine.

For automatic operation of the extra functions provided by a function only decoder it is necessary to define a *train* and to setup the “dummy engine” representing the function only decoder along with the actual engine as multiple unit. Additionally it is necessary to check the option **Forward Function** (see Diagram 65). If different function symbols are used for the functions of the actual engine and the functions provided by the function only decoder, then it is possible to select and activate the specific functions of the function only decoder automatically without affecting the functions of the actual engine.

Example: Automatic Car Lighting

The following example demonstrates how a train can be prepared in order to operate the car lighting in this train automatically. It is assumed that the lighting is controlled by an additional function only decoder. Perform the following steps:

- Create and setup an engine “Loco” for the actual engine heading the train.
- Create an additional engine “Dummy” and specify the digital address of the function only decoder.
- Setup the function symbols for the functions provided by the function only decoder in the engine “Dummy”. Use a unique function symbol for the car lighting that has not been already used for functions of the actual engine “Loco”.
- Create a train and assign the two engines created above to it. Don't forget to check the option **Forward Function**.
- Assign the function symbol representing the car lighting to the operations of a *schedule*, a *macro* or an *indicator* (see also Diagram 104 or Diagram 126) as desired.

3.8 Acceleration and Train Tonnage

An additional feature of **TrainController™** is the realistic simulation of *momentum*, i.e. *acceleration* and *deceleration* of *engines* and *trains.*

For each *engine* you can specify the *horse power* (see also Diagram 58). The horse power affects the acceleration of the engine. An engine with more horsepower is able to accelerate faster. The acceleration is also affected by the *type* of the engine. Usually a electric engine is able to accelerate faster than a steam engine with identical horse power. This fact is also taken into account when the acceleration is calculated.

Trains provide even more realistic simulation of momentum. It is namely possible to specify the *tonnage* of each train. The higher the tonnage of a train the longer is the time needed to accelerate the train to a certain speed or to decelerate the train. The maximum speed of a train is also limited by the tonnage of the train.

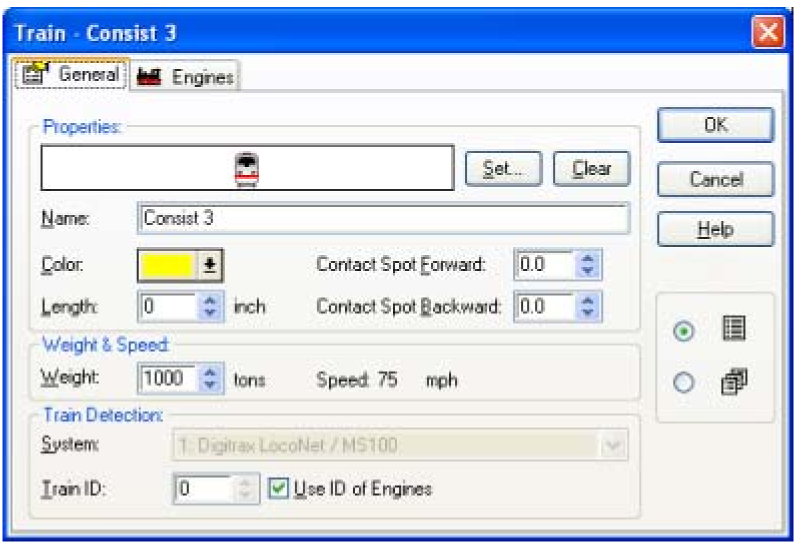


Diagram 66: General Properties of a Train

If several engines are running as a *multiple unit,* then the *horse power* of each engine is added to the total power of the multiple unit. Since the total power is higher than the individual power of each particular engine the multiple unit is able to accelerate faster and to run with a certain train tonnage at a higher maximum speed.

In the example displayed above, a train is arranged. Based on the total power of the assigned engines the program calculates a maximum speed of ... miles per hour.

The time needed to accelerate or decelerate an engine or train is additionally scaled and shortened using the scale factor of the *Railroad Clock* (see chapter 10, ” The Clock”). If for example the scale factor of the clock is 10, then the calculated times are shortened to the tenth part. Even this shortening, however, results in timing, which is often found too slow. For this reason it is possible to adjust the *inertia* of each engine individually. In this way it is possible to accelerate or decelerate an engine without inertia or with the inertia of a real engine. Any setting between these extreme cases can be selected. It is also possible to adjust the inertia for acceleration and deceleration separately (see Diagram 58).

Do not be concerned if this sounds too complicated **-** especially in the beginning. For each engine which is created **TrainController**™ assumes default settings for horse power, train tonnage and momentum. You are not required to set it. The default values result in a moderate behaviour for acceleration and deceleration which can be adjusted with the *inertia* as desired. The additional features discussed in this section are only needed if you want to simulate the behaviour of real trains.

3.9 Coal, Water and Diesel

You can specify the *type* of each *engine*. This attribute describes how the engine is powered. Possible choices are *steam engine, diesel engine* or *electric engine.*

Using this *engine type* **TrainController**™ calculates the consumption of *coal, oil, water* or *diesel*, if desired. It is possible to specify the capacity and the consumption per 100 miles of coal, oil, diesel or water.

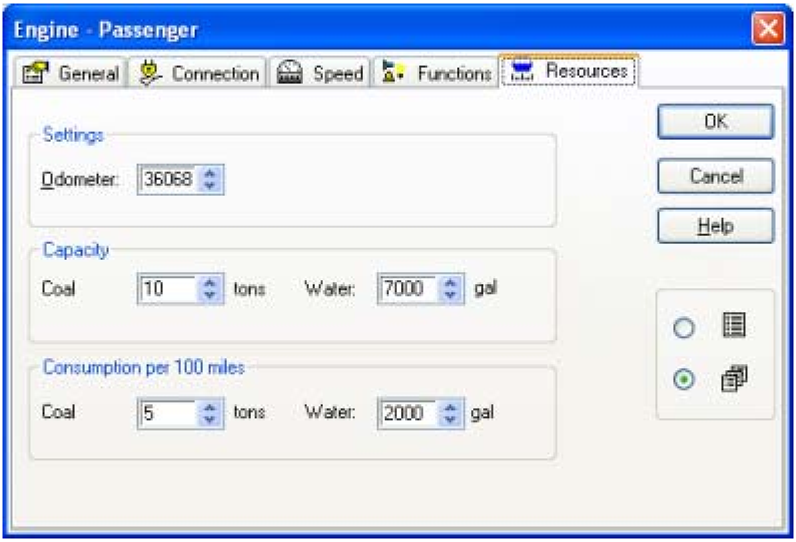


Diagram 67: Arranging Consumption of Coal and Water

This calculation can be toggled on or off as desired. If toggled on, **TrainController™** calculates the consumption of resources while the engine is running. By selecting specific menu items the resources can be reset to full, for example after the affected engine has visited an engine yard.

If an engine runs out of resources it is stopped. The affected resource must be reset to full before it is possible to start the engine again.

For electric engines no consumption of resources is calculated.

3.10 Monitoring the Maintenance Interval

For each *engine* the time it has been in operation since the last *maintenance* is indicated. This time is increased appropriately when an engine is running.

Based on the recommendations of the manufacturer of your engine, you can determine when it is time to lubricate the wheels or to change the carbon brushes. After maintenance you can reset the time indication to 0 (see Diagram 56).

3.11 Passing control between Computer and Digital System

Initially control of each *engine* is assigned to the computer. This means that the software assumes that it has full control over the engine.

With specific menu commands, it is possible to pass control from the computer to the digital system and vice versa.

If control is passed from the computer to the digital system then control of the related digital address is passed to a handheld of the digital system, if necessary. Additionally – if supported by the digital system **- TrainController™** begins to monitor speed and function changes of this engine and reflects such changes in the Train Window accordingly.

For train tracking (see section 5.5, "Train Detection and Train Tracking") of an engine it is important that the software knows the direction and speed of a running engine. If you want to control an engine with a handheld of your digital system under simultaneous train tracking, then it is necessary to assign control of the engine to the digital system before.

If an automatic schedule of the Dispatcher (see section 5.10, "Schedules") is executed with an engine currently under control of the digital system, then control of this engine is temporarily passed to the computer. After finishing the schedule control is passed back to the digital system. Such transfers of control are performed by the software automatically if needed.

Assigning control of an engine to the computer is therefore only necessary, if you want to control the engine <u>manually with the on-screen throttle</u>.

4 Contact Indicators

If your digital system is able to report the state of *track contacts*, *reed contacts*, *optical sensors*, *track occupancy sensors* or other feedback sensors to the computer, then you can indicate the status of these contacts and sensors with *contact indicator* symbols. With these indicators, you are able to monitor the state of the feedback sensors on the computer screen.

Contact indicators are always needed for automatic control of your layout with the *Visual Dispatcher* (this is outlined in detail in the sections 5.6 - 5.8). They can also be used to achieve semi-automatic control as outlined in chapter 11.

A certain indicator symbol can be located in a switchboard and/or in the *Visual Dispatcher*. Locating an indicator symbol in a switchboard is especially useful in cases, where the *Visual Dispatcher* is not being used at all or if the switchboard represents an area of your layout, which is not controlled by the *Visual Dispatcher*. Locating an indicator symbol in the Visual Dispatcher is useful, if the related real contact is located in an area of your layout, for which no switchboard exists, or if the indicator is intended to be assigned to a block (see 5.6, “Blocks and Indicators”).

There is no major difference, whether an indicator symbol is located in a switchboard or in the Visual Dispatcher. Everything written in this document with regard to indicators applies regardless where the indicator is located.

Contact indicators can be used to determine the positions of *trains*. This can be done in or without conjunction with train identification systems (e.g. Digitrax Transponding, Muet or HELMO). For more details refer to section 5.5, “Train Detection and Train Tracking”.

Contact indicators can also be used to operate other objects like switches and signals, semi-automatically, by passing trains. More details about this issue are outlined in section 11.3, “Operations”.

If your digital system is not able to report the state of feedback sensors to the computer, then **TrainController**™ makes it possible to connect a second or more digital systems to your computer. For this purpose, it is not necessary to purchase another complete digital system that is also able to operate trains and switches. **TrainController**™ supports special low cost digital systems that are dedicated to the monitoring of feedback sensors. More details about running several digital systems, simultaneously, are outlined in section 14.3, “Operating Several Digital Systems Simultaneously”.

Feedback sensors are distinguished between *momentary track contacts* and *occupancy sensors*. In **TrainController**™ the same symbol is used for both types of contacts. The difference between both types of contacts does not play an important role as long as trains are not operated under control of the *Visual Dispatcher* (see section 5, “The Visual Dispatcher”).

Momentary Track Contacts vs. Occupancy Sensors

Momentary track contacts are turned on for a short moment, when a train passes a certain point on the model railroad. They stay turned on only for a short moment and are turned off as soon as the train moves any further. In Diagram 68 to Diagram 70 you can see a momentary contact triggered by a passing train. Momentary track contacts indicate that a train is about to pass a certain point. *Occupancy sensors* are turned on when a train enters a certain section on the model railroad. They stay turned on until the train leaves that section completely. Occupancy sensors indicate that a train is located inside a certain track section. In Diagram 71 to Diagram 74 you can see an occupancy sensor turned on and off by a passing train. Occupancy sensors are able to report the presence of a train inside a certain track section even if the train is not moving. Momentary contacts are triggered by moving trains only. Momentary contacts can be made for instance by mechanical track contacts, reed contacts or optical sensors. Occupancy sensors are often based on current sensing in isolated track sections.

Unlike other programs which require occupancy sensors for automatic train control **TrainController**™ is also able to control trains if only momentary track contacts are used. Occupancy sensors are more safe, though, because with momentary contacts special measures against premature release of blocks and routes must be taken.

The following diagrams show the behaviour of a momentary contact in the different phases while a train is passing. The position of the momentary contact is marked with a short vertical line.

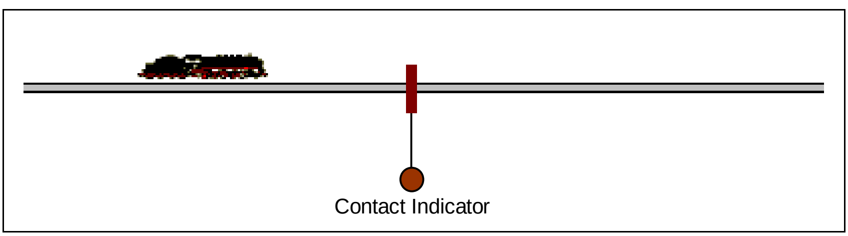


Diagram 68: Train is approaching the momentary contact – the contact is turned off

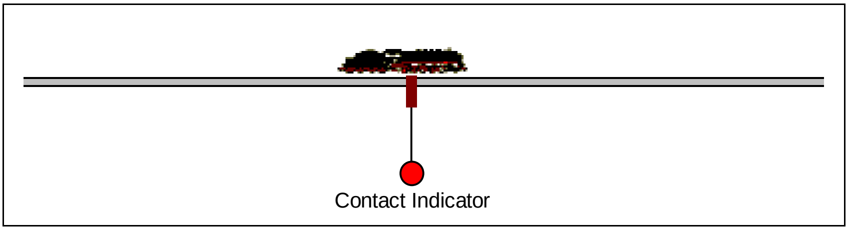


Diagram 69: Train is reaching the momentary contact – the contact is triggered

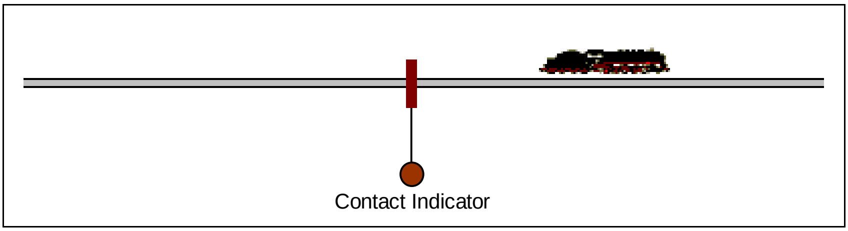


Diagram 70: Train is leaving the momentary contact – the contact is turned off

The following diagrams show the behaviour of an occupancy sensor in the different phases while a train is passing. The track section sensed by the occupancy sensor is marked with a horizontal line. This line is drawn in bright red while the sensor is turned on.

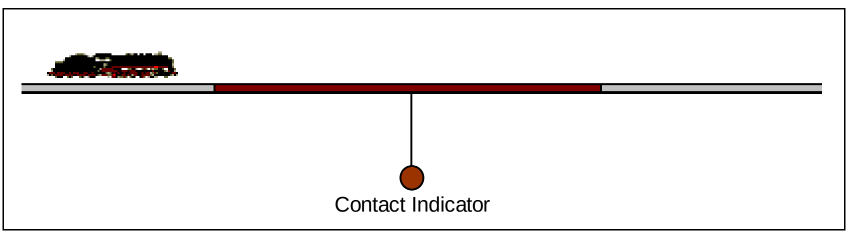


Diagram 71: Train is approaching the occupancy sensor – the sensor is turned off

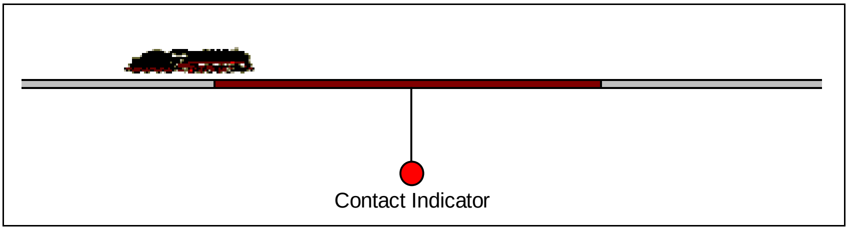


Diagram 72: Train is located inside the sensed section – the sensor is turned on

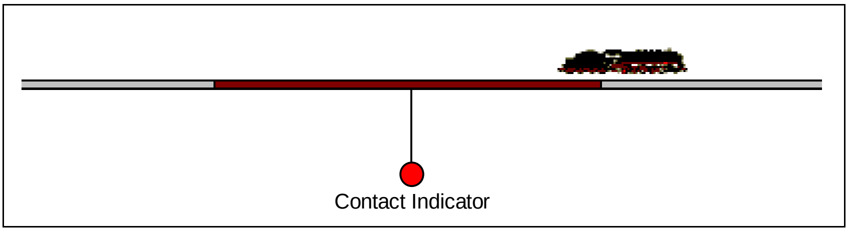


Diagram 73: Train is still located inside the sensed section

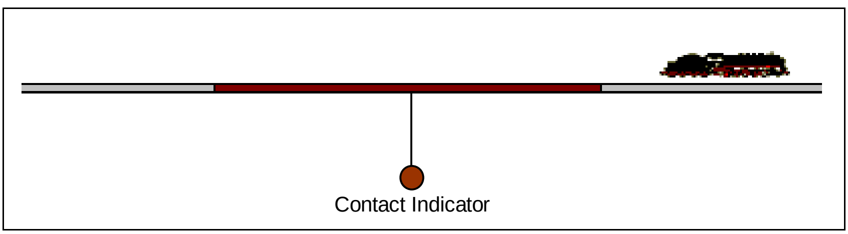


Diagram 74: Train has left the sensed section – the sensor is turned off

There is one major difference between momentary contacts and occupancy sensors to remember: the points at which the indicators are turned on. A momentary track contact is turned on when a train reaches a certain point on the layout regardless of the direction of travel of the passing train. In this way a momentary track contact represents one single sensing point on the model railroad. An occupancy sensor is turned on when a train reaches either end of the sensed track section depending on the current direction of travel of the passing train. In this way an occupancy sensor represents two different sensing points on the model railroad. It depends on the direction of travel of a passing train at which of these two points the train triggers the sensor.

5 The Visual Dispatcher I

5.1 Introduction

A human operator is normally only able to operate one or two switchboards and at most two trains at the same time. If multiple control panels or a certain number of trains are to be operated at the same time, then either support of additional human operators is required, or a component like the *Visual Dispatcher*, which is able to take the place of additional human operators.

The V*isual Dispatcher* (or in a word *Dispatcher*) is a component that makes large scale railroad operations manageable by one person, matching operations found on the largest club layouts.

Like a human operator the Visual Dispatcher is able to operate switches, signals, routes and trains. This is called *automatic operation*.

A broad range of operating flexibility is provided from completely manual through fully automatic operation (e.g. hidden yards control). Manual and automatic operation can be mixed simultaneously. This applies not only to trains on different areas of your railroad, but also to different trains on the same track and even to the operation of a single train. Automatic processes are not bound to specific trains. Once specified they can be performed by each of your trains. There is no need to learn a programming language. Time-table and random functions increase the diversity of your model railroad traffic. Built-in train tracking functions display on the screen which engine/train is on which track.

Like a human operator must know the overall structure of the model railroad layout the *Visual Dispatcher* needs to know this, too. This structure is represented by a diagram that contains blocks and routes and the track connections between them. This diagram is called *main block diagram* of the layout. The main block diagram describes the track layout of your entire model railroad in rough outline.

The *Visual Dispatcher* manages traffic flow using a blocking system. Blocking ensures that trains do not collide and supports the tracking of train positions. For this purpose, the railroad layout is divided into virtual, logical blocks. That means, you define blocks at locations where traffic control will take place (e.g. scheduled stops in a station). Usu-

ally each track in a station or hidden yard, each siding and appropriate sections of the connections between two yards will form a block.
Blocks are arranged graphically and connected by *routes* and *links* to specify on which path a train will travel from certain starting blocks to destination blocks. *Schedules* describe train movements, i.e. how trains travel. This includes scheduled waits, speed limits, etc.

Trains can run under full manual control, in which case the operator will be responsible for obeying the block signals set by the *Dispatcher*; or under full control of the computer; or even with an intermediate level of automation.

For shunting special types of schedules are provided.

Schedules and *timetables* can be arranged with a broad range of flexibility. Since a timetable can be created for each day of the year, up to 365 timetables can be used.

Random functions increase the diversity of your model railroad traffic.

Creating a model railroad operation system with the *Dispatcher* is done by performing the following steps:

- Divide the model railroad layout into *blocks* and enter these blocks into **TrainController™**
- Arrange blocks and routes as well as the links between them in the *main block diagram*. This diagram shall represent the track layout of your entire model railroad in rough outline.
- Arrange *schedules* and optionally create *timetables*

These steps will be outlined in more detail in the following sections. We will do this by looking at the following sample layout:

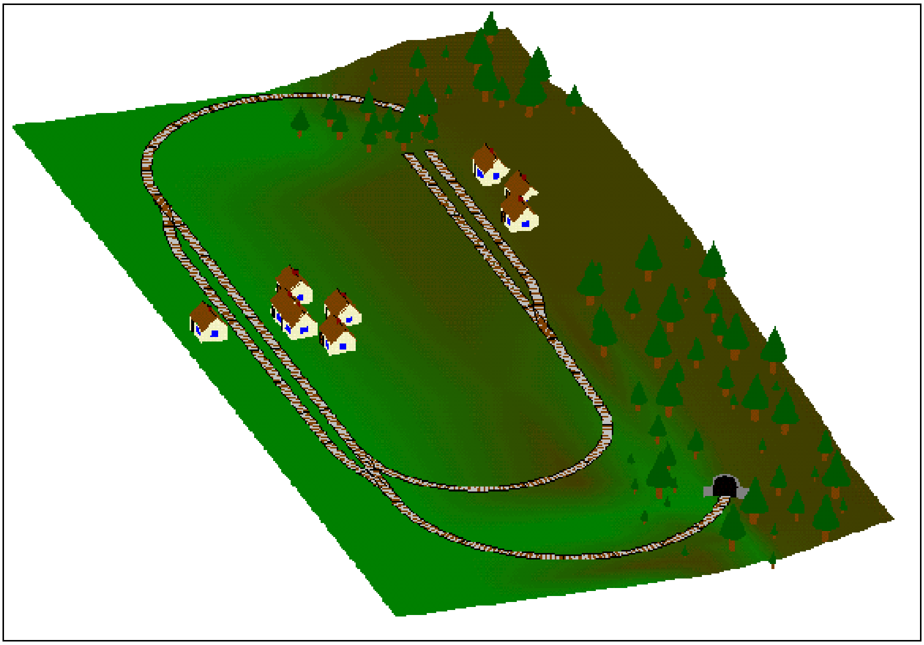


Diagram 75: Sample Layout

The layout has two stations: “Southtown” located on the left side of the layout and “Northville” located at the end of a branch line. There is an additional hidden yard that is covered by the mountain.

This can be seen better in the track plan displayed below:

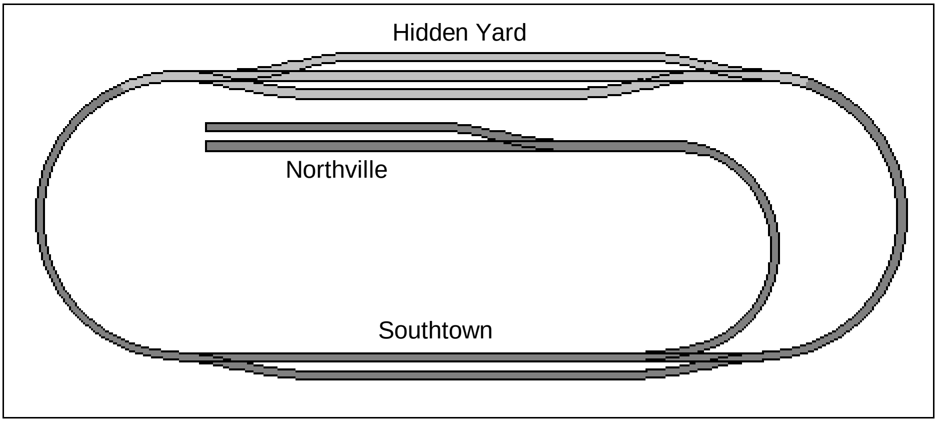


Diagram 76: Track Plan of the Sample Layout

The main line, i.e. the loop that connects “Hidden Yard” and “Southtown”, will be operated automatically under full control of the *Visual Dispatcher*. The branch line from “Southtown” to “Northville” will be operated manually.

The parts of the layout that are covered by structure and therefore invisible are drawn here in a slightly brighter colour.

The first step is drawing of a switchboard for the layout displayed above. It looks as follows:

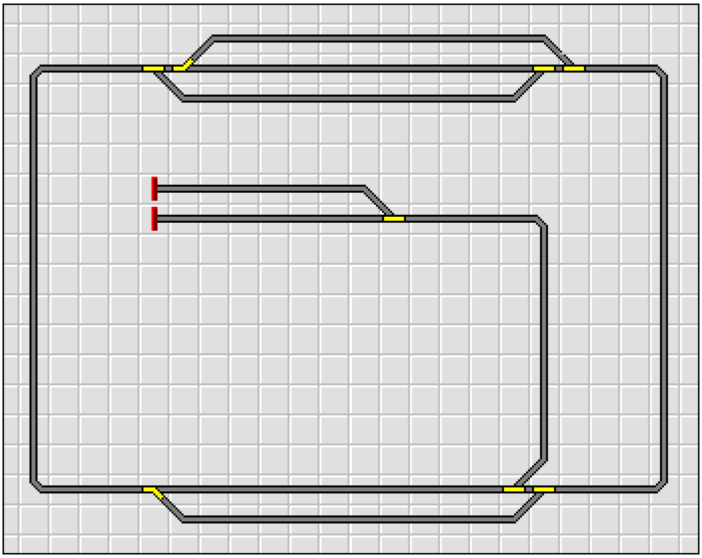


Diagram 77: Switchboard of the sample layout

The next steps, that are required to configure this layout in the *Visual Dispatcher*, are outlined in the following sections.

5.2 Blocks

Blocks on the Layout

B

The *Visual Dispatcher* manages traffic flow using a *blocking system*. Blocking ensures that trains do not collide. For this purpose the railroad layout is virtually divided in logical blocks. That means, you define blocks at locations, where traffic control shall take place (i.e. stopping inside a yard) or where trains are parked. Blocks are also used to determine and to indicate the position of your engines and trains on your tracks.

Typical examples of blocks are

- Tracks at a platform
- Sidings in a (hidden) yard
- Block sections on tracks between two stations

In most cases blocks contain only a straight track section and no turnouts. They are usually limited by two turnouts on both sides or by a turnout and a dead end of the track. Block sections between two stations are often limited by block signals.

Some guidelines for arranging your blocks:

- Blocks may be located anywhere on your railroad.
- Blocks are often limited by turnouts. These turnouts usually do not belong to the blocks.
- Blocks should be long enough to hold each stopping train completely.
- Each location, where the *Visual Dispatcher* shall be able to stop a train automatically (e.g. in a station or in front of a signal), should be located in a separate block, i.e. in order to stop two trains at the same time at different locations, these locations must be arranged in different blocks.
- The more blocks are provided the more trains can be run simultaneously under control of the *Visual Dispatcher*.
- Each block can be reserved by at most one train. A specific train may reserve several blocks. A train, that runs under control of the *Visual Dispatcher*, may enter blocks only, that are reserved for this train.
- Blocks must be provided only for the parts of your model railroad, which shall be controlled by the *Dispatcher*. Parts without blocks are not visible for the *Dispatcher*.

Following these guidelines the block structure of the sample layout looks as follows:

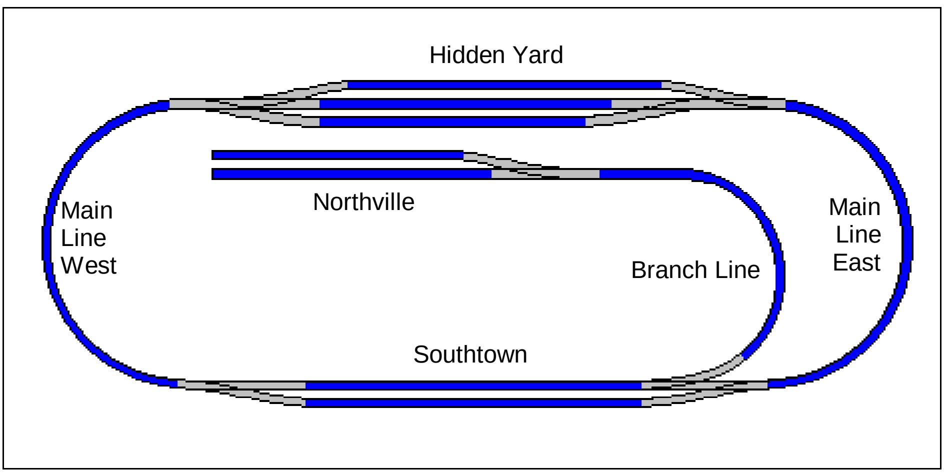


Diagram 78: Block structure of the sample layout

Each blue track section represents a separate block. The blocks on the main line or the branch line between “Southtown” and “Northville” can be subdivided any further into more blocks if each of these blocks is long enough to store the longest train. This is useful if you want more than one train to travel on these tracks at the same time.

The Main Block Diagram

Like a human operator must know the overall structure of the model railroad layout the *Visual Dispatcher* needs to know this, too. This structure is represented by a diagram that contains *blocks* as well as the routes and track connections (*links*) between blocks. If there are turnouts to be thrown in order to let a train run from one block to an adjacent block, then the blocks must be connected by appropriate *routes* (see section 2.6, “Routes”). This diagram is called *main block diagram* of the layout. This diagram describes the track layout of your entire model railroad in rough outline.

The block diagram can be drawn by yourself or automatically created by the software. The latter is especially useful for small and medium size layouts, where the track plan of the entire layout fits into one single switchboard. For this sample layout we decide to let the software calculate the block diagram. Additional hints for manual creation of the block diagram are given in Part III of this Users Guide, Chapter 12, “ The Visual Dispatcher II”.

TrainController™ uses the track diagram of the switchboard, that is contained in the main window of the software, as base of calculation of the block diagram. For this purpose it is necessary to specify the positions of the blocks in this track diagram. This is done with the help of so called *traffic boxes*. Each traffic box represents a block in the track diagram of the switchboard. For each block determined in Diagram 78 a traffic box is created and added to the switchboard in Diagram 77. The result is displayed in the following diagram:

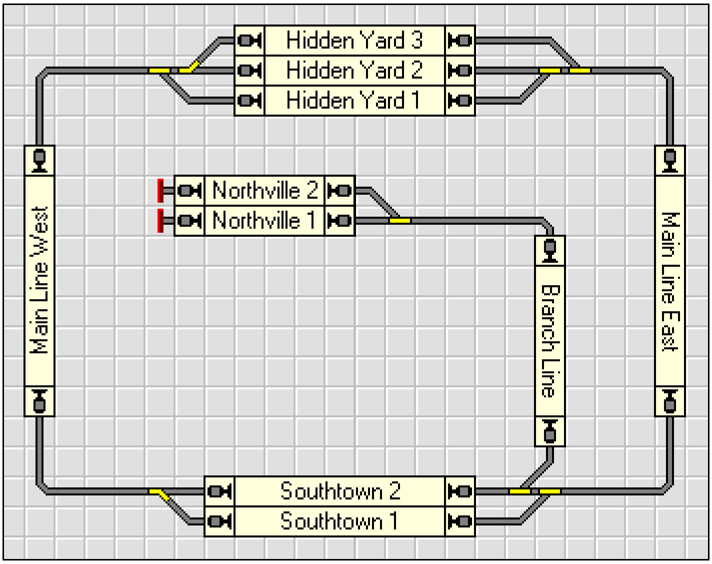


Diagram 79: Switchboard with Traffic Boxes

The software automatically calculates a block diagram, that is displayed in the *Visual Dispatcher* as shown below:

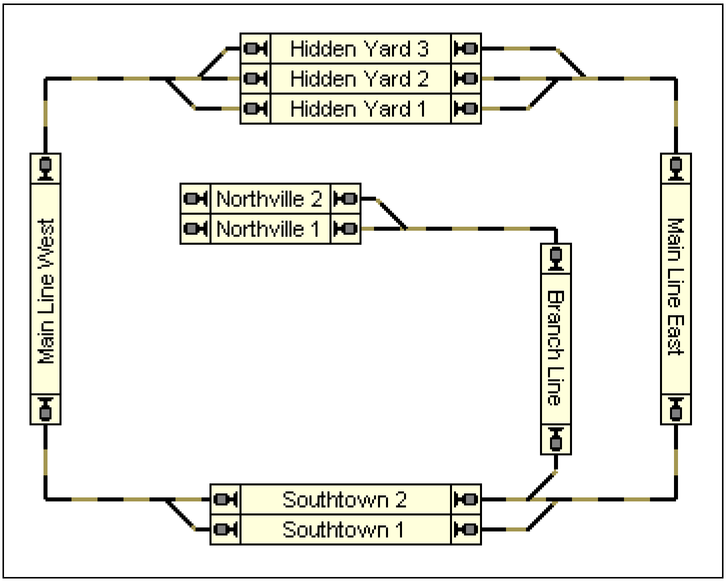


Diagram 80: Main Block Diagram in the Visual Dispatcher

Blocks are displayed on the computer screen by rectangular boxes. The blocks are connected by routes or links. These routes or links are drawn as lines. Since there are turnouts to be operated in this example in order to travel from one block to the next each two blocks are connected by a route. These routes are created and recorded automatically by the software (see also Section 2.6, "Routes").

!

Routes are inserted automatically between two blocks, when the software detects the symbol of a turnout or crossing in the track connection between both blocks.

Please note that the block diagram represents the track layout in rough outline. The actual track connection between "Main Line West" and "Hidden Yard 3", for example, contains two switches. These switches are not drawn in the block diagram in detail or as separate objects. Instead a link between both blocks is created, that indicates, that there <u>is</u> a track connection between both blocks.

In order to enable **TrainController™** to calculate the block diagram automatically note the following:

- Draw the complete track diagram of your layout with all turnouts and crossings and without any gaps in the switchboard, that is contained in the main window of the software.
- Create traffic boxes for all blocks of the layout, place them according to their location on the actual layout and ensure, that they are turned horizontally or vertically according to the track symbols, to which they are attached.
- Ensure that the blocks are connected by track symbols without any gaps.

Links and Routes between Blocks

B

In order to let trains run from one block to another the blocks must be linked together. This is done with the help of *links* or *routes*. In the block diagrams links and routes are represented by lines that connect one block with an adjacent block.

Each block has two entries/exits. If a block is passed horizontally, then the entries/exits are graphically located on the left and on the right side of the block. If a block is passed vertically, then the entries/exits are located at the top and at the bottom. Each link or route begins at the entry/exit of a block and ends at the exit/entry of an adjacent block.

Routes are used to connect two blocks, if turnouts are to be thrown in order to allow a train to travel from one block to the other.

The following image explains the terms once more:

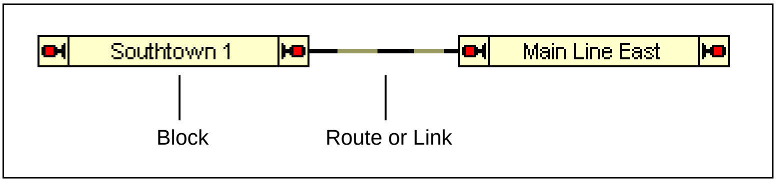


Diagram 81: Blocks, Route and Link

In the diagram displayed above the blocks “Southtown 1” and “Main Line East” are connected with a link or a route.

If you block diagram is automatically calculated, then the necessary links and routes are automatically created according to the track connections between the blocks (traffic boxes) in the track diagram of your switchboard.

!

The main block diagram of the complete layout is drawn within one single diagram. The space provided by the software is not limited. In case of large layouts it

is possible to zoom and scroll the window accordingly, in which the block diagram is displayed.

5.3 Direction of Travel vs. Engine Orientation

B

It is important to understand the difference between *direction of travel* and the *orientation* of an engine.

Direction of Travel

Direction of travel is seen from the passenger's point of view. For the passenger sitting in a train it is important to know, whether the train runs from the east to the west, from the city to the country, or from the sea to the mountains. The direction of travel has a "geographical" meaning. Each *block* can be passed in one of two directions at a time. For each train controlled by the *Dispatcher* the *Dispatcher* must know the train's intended direction of travel. This information is derived by the *Dispatcher* from the arrangement of the blocks in the related diagrams and the links that contain these blocks.

TrainController™ draws each block to represent one pair of corresponding directions. Each block can be either passed horizontally (from the left to the right or back) or vertically (from the top to the bottom or back).

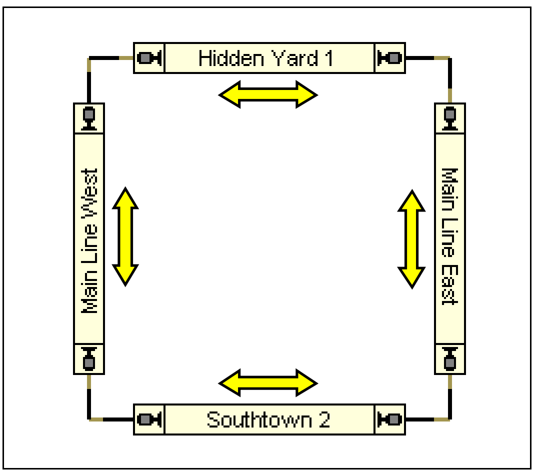


Diagram 82: Block Diagram of a Circular Layout

In the diagram displayed above the direction of travel of each block is here indicated by an arrow. **TrainController™** does not display these arrows, though, but displays small signal symbols on two sides of each block, to mark the direction of travel, that belongs to the block.

The direction of travel will correspond to the drawing of the block in the diagram. A block that is passed horizontally will be drawn as a horizontal rectangle while a block that is passed vertically will be drawn as a vertical rectangle. This is shown in the diagram displayed above.

Engine Orientation

Engine Orientation is seen from the engineer's point of view. It is not important for the passenger. The engine orientation describes the direction of the engine's head. For an engineer, who has to run a train in a certain direction of travel it is also important to know the engine's orientation, i.e. the direction of the engine's head. Depending to the intended direction of travel and the engine's orientation the engineer can decide, whether the engine is to be run forward or backward.

When the *Dispatcher* runs a train, it acts like an engineer. Especially both information - the intended direction of travel and the engine's orientation - must be known by the Dispatcher to start the train accordingly.

The orientation of each engine is specified during assignment of an engine or train to a block. There are several methods to assign trains to a block. The most convenient method is to drag & drop a train icon to the traffic box or the symbol of a block. **Please check always that the current orientation of the engine matches the screen display.** In case both do not match it is possible to revert the screen display with appropriate menu commands.

Another method for automatic assignment of trains to blocks is the use of train detection or train tracking (see 5.5, “Train Detection and Train Tracking”).

5.4 States of a Block

The different *states* of a block are determined by the fact whether the block is *occupied* or whether it is *reserved* for a certain *engine* or *train*.

Occupied Block

A block is assumed to be *occupied*, if at least one of the *indicators* assigned to the block is turned on.

Reserved Block

Each block can be manually or automatically *reserved* for an *engine* or *train* by the Dispatcher. Reservation serves to support the following goals:

- Since a block can be reserved only for at most one *engine* or *train*, train collisions are avoided if blocks are arranged and reserved correctly.
- The program is able to determine, in which block a certain engine or train is located. This enables operations tied to the locality of trains **-** for example stopping a train in front of a red signal.
- The use of *traffic boxes* allows indication of train positions in the *switchboard*.
- Train detection and train tracking is based on dynamic and automatic reservation of blocks, too (see 5.5, “Train Detection and Train Tracking”).

For *shunting* or similar manual operation it is possible to reserve a group of related blocks manually. In this case the *Dispatcher* takes care, that automatically controlled trains do not enter these blocks. If reserved blocks are no longer needed, they can be *released* by yourself or automatically by the *Dispatcher.*

Current Block

Among the blocks, which are reserved for a train, there is a special block, where the head of the train is assumed to be located. This block is called the *current block* of the train. Through the current block all block related operations which affect the speed of a train (like running with restricted speed) are performed.

In the beginning you must manually assign each engine or train to its current block. Afterwards this assignment is adjusted automatically by **TrainController**™ according to the position changes of the affected trains. Even after terminating and restarting of the program this assignment is automatically updated. Only if an engine or train is moved by hand to another track you must assign the engine or train to its new current block again.

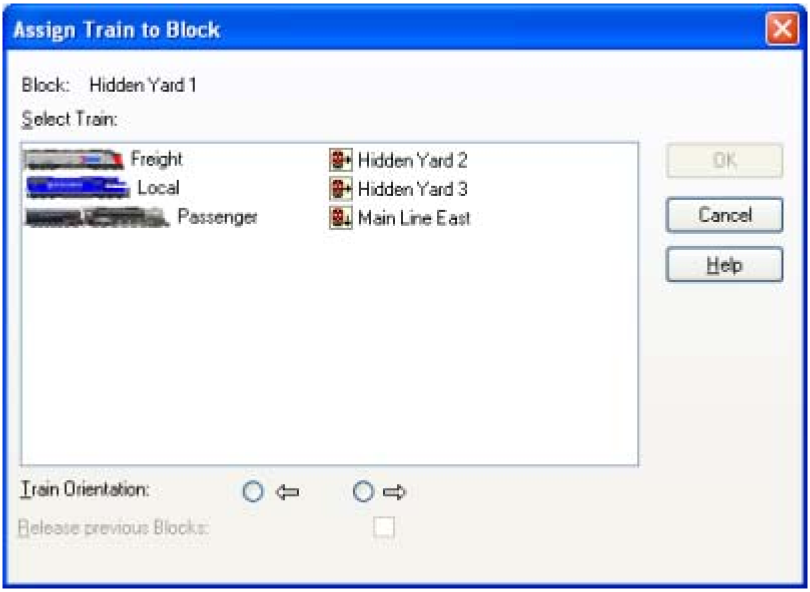


Diagram 83: Assigning a train to the Current Block

When an engine or train is assigned to its current block, then the current *engine orientation* must be specified. **TrainController™** needs to know this orientation to be able to determine, if an engine runs forward or backward. **TrainController™** adjusts the engine orientation accordingly even if an engine changes its orientation by passing a *reversing loop*.

TrainController™ provides several methods to assign a train to a block. The most convenient is to drag a train from the train list to the symbol of a block. The initial assignment of an engine or train to a block can also be done automatically without manual interaction, if a *train detection device* is used (see section 5.5, "Train Detection and Train Tracking"). If this device is associated with a *contact indicator* and this indicator is again assigned to a block then each engine or train detected by the train detection device will be automatically assigned to this block.

A reserved block must not necessarily be occupied. This is also true for the current block. If for example a train leaves its current block and temporarily no other blocks, that are reserved for this train, are occupied, then the current block is not changed, before the train enters another block and this block is indicated as occupied.

Display of Train Positions

The states of a block outlined are indicated by the concerning *traffic boxes* in the switchboard. In this way you can control in the switchboard, too, if a certain block is occupied or reserved. Traffic boxes display the name and/or the image of the train, that is currently located in the related block, in the switchboard. For further details refer to 5.5, “Train Detection and Train Tracking”, please.

Locking of Blocks

Each block can be temporarily locked during operation. Locked blocks cannot be reserved nor entered by running trains. A train, that is already located in a block, when the block is locked, might stay there, though, and leave the block later.

Please note that locking of a block affects all trains. Through the options of schedules it is possible to exclude specific trains from using a certain block.

Locking the exit of Blocks

Each exit of any block can be temporarily locked during operation. A block cannot be left through a locked exit. Trains may enter such blocks and may stay there, but they cannot leave a block through a locked exit.

It is possible to lock either exit of each block individually and independently from the opposite exit.

Please note that locking of a block exit affects all trains. Through the options of schedules it is possible to exclude specific trains from leaving a block.

5.5 Train Detection and Train Tracking

TrainController™ is able to indicate the positions of your engines and trains on the computer screen. This is always and automatically done in the screens of the *Visual Dispatcher*, such as the main block diagram or the particular schedule diagrams.

The *traffic boxes* in the switchboard also display the state of the associated block and optionally the name and/or image of the train that is located in this block.

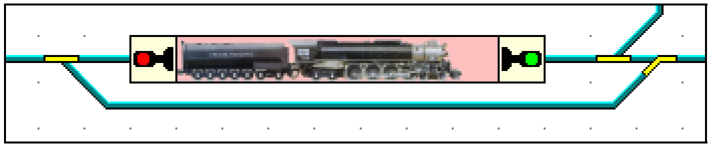


Diagram 84: Traffic Box in the Switchboard

Train Detection

Contact indicators are normally used to determine, if a certain track section is occupied by any *engine* or *train*. Some digital systems are not only able to report occupancy, but also the identity of the occupying train. Examples of such systems are Digitrax Transponding, Muet or HELMO. If a contact indicator is associated with an appropriate train detection device, then this indicator can also be used to determine the train that is occupying a certain track section or detection zone, respectively. Using contact indicators in conjunction with train detection devices is called *train detection* in **TrainController**™.

It is very simple to setup a train detection system in **TrainController**™.

On the computer screen each train detection device/train detection zone is represented by a *contact indicator*. Such indicator is configured in the software as usual as if it were a normal contact indicator without train identification capabilities. Only the digital address of the train detection device/zone is to be specified (Diagram 85).

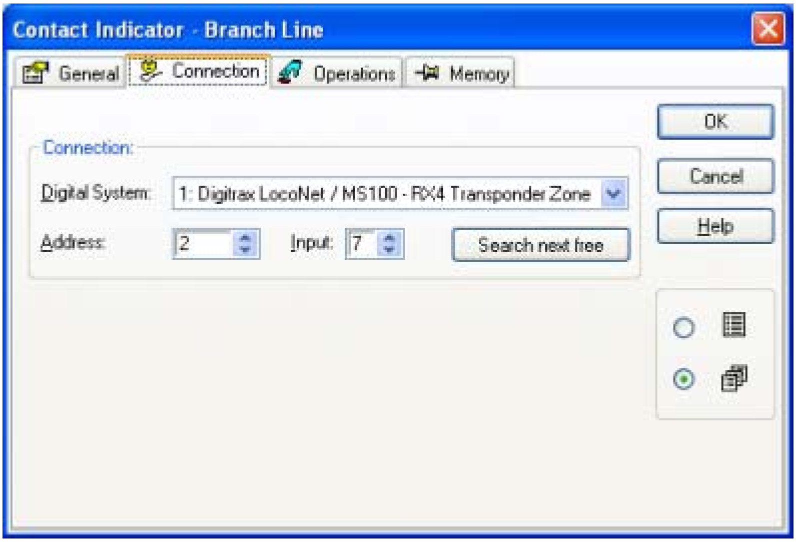


Diagram 85: Specifying the digital address of a train detection device

The next step is the assignment of this indicator to the block, where the section or zone belongs to. The assignment of indicators to blocks is explained in more detail in section 5.6, “Blocks and Indicators”.

Finally the train IDs are entered in the properties of the related engines or trains. The properties of each engine and train (see chapter 3, “Train ”) provide special options to specify an individual train ID for each engine or train. This is displayed in the following image.

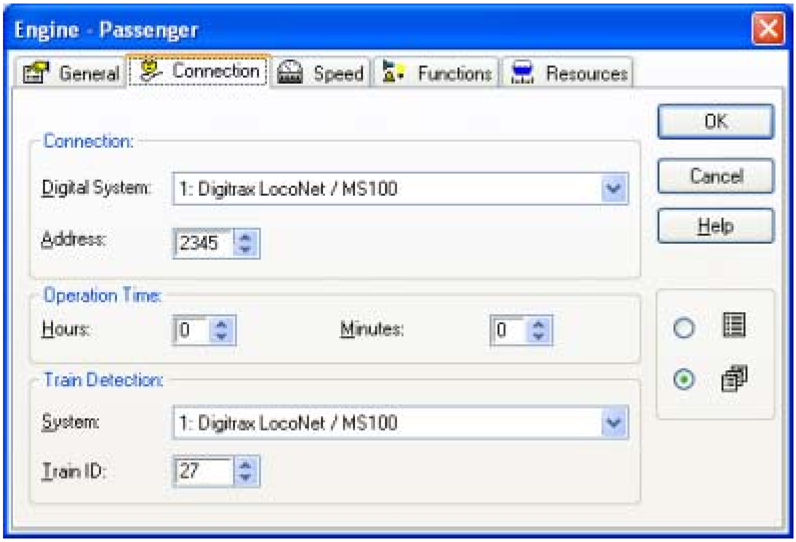


Diagram 86: Specifying the digital connection and the transponder number of an engine

In this example the train ID 27 is assigned to the steam engine with the digital address 2345. Of course digital address and transponder number do not have to be identical. Especially in case of multiple units or if the transponder is mounted on a car rather than an engine it is very useful, that decoder addresses and transponder numbers are treated independently.

For *trains* an additional option, **Use ID of Engines**, is provided. If this option is set then the train is not associated with an own train ID. Instead the IDs of the engines, that have been assigned to this train, are used. If the train is running and the ID of one of its engines is detected then this ID is mapped to the running train.

The screen displayed above might slightly change its appearance dependent on the actual capabilities of the connected train detection system. For some train detection systems you will not explicitly have to assign a train ID, instead there is a kind of auto-capture mechanism, with which train Ids can be automatically read from a passing train.

If more than one digital system is connected (**TrainController**™ allows simultaneous operation of up to 12 digital systems) then it is even possible to use different digital systems for train steering and train detection. In this way it is possible to use a system like Digitrax as an additional train detection system, even if a digital system of another manufacturer is already installed. It is furthermore possible to use a train detection system like Digitrax on model railroads, which are controlled conventionally (in which case only transponder numbers and no digital decoder addresses are to be specified for the particular engines).

Specifying the train ID for each affected engine is the only additional effort with regard to the configuration of engines or trains. Nothing else is to be done.

Here is again a list of the necessary steps to configure train detection:

- Create an indicator symbol for each train detection device or train detection zone, respectively, that you want to use for train detection in **TrainController**™.
- Assign each of these indicators to an appropriate block.
- Specify the engine/train ID of each engine or train you want to use for train detection in the properties of each related engine or train.

When these steps have been done then the name and/or image of the train, that passes a certain train detection device or detection zone will automatically appear in the block symbol of the *Visual Dispatcher*. If there are one or more optional *traffic boxes* in a switchboard window associated with this block, then the train will appear in these boxes, too.

By assigning a train indicator, that is associated with a train detection device/detection zone to a block a relation is established between such detection zone and a block in **TrainController**™. This relation is used to perform an automatic train-to-block assignment, when a train is being detected in a train detection zone.

This relation should also be taken into account, when you wire your layout. Like a regular occupancy sensor the same train detection device or detection zone can also belong to only one block in **TrainController**™ (see also section 5.8, “Arranging Indicators in a Block”). When a train is being detected in a train detection zone, it must be possible to clearly determine the block, to which the train is to be assigned.

TrainController™ does not only use train detection for automatic train-to-block assignment, but also for more complex safety functions. The *Dispatcher* uses train detection as a redundant anti collision protection in addition to the train tracking algorithms implemented in the software (see next section). If a train is reported in a block, which does not correspond to one of the “expected” positions calculated by the software, then the user is alerted and affected trains are stopped if desired.

Train Tracking

The *Visual Dispatcher* uses the *main block diagram* to perform automatic train tracking.

Whenever a block is reported as occupied, because one of the indicators assigned to it is turned on, then the *Dispatcher* checks, whether there is an appropriate train in an adjacent block. An adjacent block is a block that is connected with the current block with a link in the *main block diagram*.

If there is such train, then the train is moved to this block. This is done by automatic assignment of the train to the new block and releasing of the previous block.

As a result of this movement the name and/or image of the engine or train appears in the block symbol of the related block in the *Visual Dispatcher*. Additionally the train disappears from the symbol of the previous block. If there are one or more optional *traffic boxes* in a switchboard window associated with these blocks, then the train movement will be visible in these boxes, too.

If there are more than one train located in adjacent blocks, than the Dispatcher tries to determine the most probable candidate. For this calculation the speed of each train and the direction of travel, if known, or the occupancy state of each adjacent block is taken into account.

In order to achieve precise results it is important to assign the initial position and orientation of each train correctly. Additionally you should always ensure that the software is able to track the direction and speed of each train. The control of trains that you operate with the throttle of your digital system should properly be assigned to the digital system (see 3.11, “Passing control between Computer and Digital System”).

Train tracking can also be disabled for certain blocks or temporarily for the complete layout. This is useful for areas or in situations, when you put new engines or trains manually on the physical track and you want to avoid unintentional train tracking caused by the resulting occupancy messages.

Attention: train tracking is turned on by default. Unintented triggering of indicators, that are assigned to blocks, might cause train assignments to be moved in the block diagram. If this is not desired in certain situations then train tracking can be (temporarily) disabled for the complete layout.

- Under the conditions listed below train tracking works for each engine or train on the layout, which has been previously assigned to a block.

- The initial assignment of trains to blocks can be done manually or automatically by train detection. Train detection rids you from performing the initial assignment manually; train detection is no prerequisite of train tracking; though.
- Train tracking is based on the *main block diagram* of the Visual Dispatcher and follows the specified links between the blocks. The tracking of manually operated trains, such as those trains that you control with the throttle of your digital system, is only possible, if you create an appropriate main block diagram, that contains the proper links between your blocks.

!

For train tracking of an engine it is important that the software knows the direction and speed of a running engine. If you want to control an engine with a handheld of your digital system under simultaneous train tracking, then it is necessary to assign control of the engine to the digital system before (see section 3.11, "Passing control between Computer and Digital System").

5.6 Blocks and Indicators

B

For proper operation the *Dispatcher* must be able to detect, whether a train occupies a specific section of your railroad or when a train passes a specific point on your railroad. This detection is done with *contact indicators* (see section 4, "Contact Indicators").

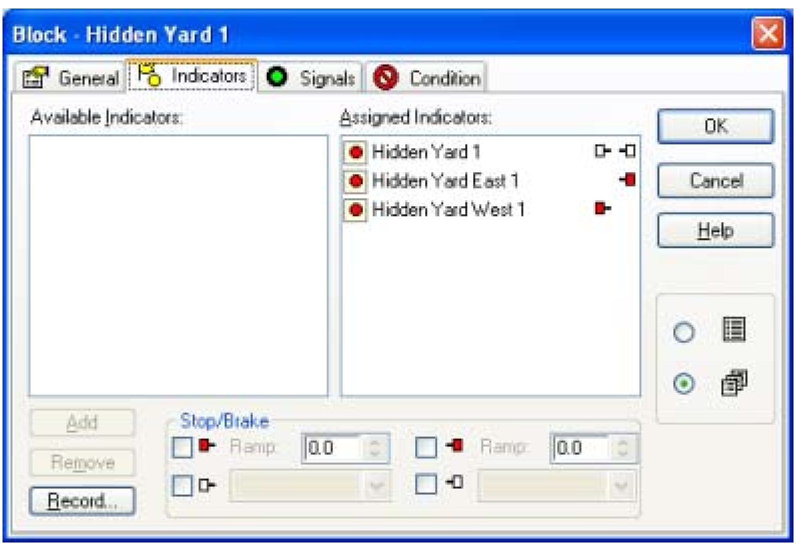


Diagram 87: Assigning Indicators to Blocks

In order to create a block all *indicators* that are located in the block are assigned to the block. If at least one of these indicators is turned on, then the block is assumed to be occupied. The actual layout positions of the sensors assigned to the block determine also the location of the block on your railroad. In order to have control over the exact location, where a train is to be stopped inside a block, you can mark certain indicators as *stop* or *brake indicators* (see section 5.7, “Stop and Brake Indicators”).

There are several methods in **TrainController**™ to create blocks and to assign indicators to them.

To establish a block on your railroad, it is necessary to install the required sensors. Depending on the principle of the used contact sensors it may be necessary to electrically insulate the track section belonging to a certain contact sensor from adjacent sections. Whether electrical insulation is necessary or not depends solely on the contact sensors being used. The software does not require electrical insulation of your blocks.

- The software does not require that a block is electrically insulated from other blocks. The used sensors might require this, though.
- Blocks usually contain several indicators. If these indicators represent isolated or separate track sections then several track sections are contained in the same block (see also 5.8, “Arranging Indicators in a Block”).
- The same indicator <u>cannot</u> be assigned to several blocks. Especially you should install your sensors on your layout in a way, that each sensor section is associated with at most one block. If you use a train detection system (see 5.5, “Train Detection and Train Tracking”) then each train detection section or zone, respectively, should be associated with at most one block.

It is possible to assign indicator symbols, which are already contained in switchboards, to a block. It is also possible to create new, extra indicator symbols within a block, that are not contained elsewhere. Please note the following:

- If an indicator symbol, that is assigned to a block, is also contained in a switchboard, then deleting this indicator from the *Visual Dispatcher* also deletes the indicator symbol from the switchboard.
- If an indicator symbol, that is assigned to a block, is also contained in a switchboard, then deleting this indicator from the switchboard also deletes the indicator from the block and the *Visual Dispatcher*.
- If you want to remove an indicator from a block without deleting it from a switchboard, then edit the properties of the block and remove the indicator from the list of assigned indicators.

- If you want to remove an indicator from a switchboard without deleting it from a block, then move the indicator symbol from the switchboard to the block by drag & drop. Do this, even though the indicator symbol is already listed as assigned to the block. This additional move gives you the opportunity to remove the indicator symbol from the switchboard without deleting it from the block.

5.7 Stop and Brake Indicators

A block is created by assigning several *indicators* to it. If at least one of these indicators is turned on, then the block is assumed to be *occupied*. The indicators are used for indication of occupancy. Additionally each indicator can act as *brake indicator* and/or *stop indicator*.

It may happen, that a train has to stop when passing a certain block. This is for example the case, when the block ahead is not available or when the train shall change its direction or when the train shall stop inside the block for a certain amount of time. In order to give you control over the exact location, where a train is to be stopped inside the block, you can specify certain indicators, which are assigned to the block, as stop or brake indicators.

Let us assume that a train approaches a certain block. That means, that none of the assigned indicators was activated before and that at least one of these indicators is activated now. If this indicator is neither a brake nor a stop indicator, the block is marked as occupied and the train continues with unchanged speed. If the train reaches an indicator, which is assigned as a brake indicator for the current *direction of travel* (see section 5.3, “Direction of Travel vs. Engine Orientation”) and the train has to stop inside this block, then the train is decelerated to its *threshold speed* (see section 3.3, “Throttle and Brake”). The braking ramp can be set as desired individually for each brake indicator. If the train reaches an indicator, which is assigned as a stop indicator for the current *direction of travel* and the train has to stop inside this block, then the train is stopped at once.

Please note that a brake indicator is only effective if the train has to stop in the same block. As a consequence brake and stop indicators that belong together must be contained in the same block.

A stop or brake indicator can be valid for one or both directions of travel. It is even possible, that a certain indicator operates as a stop indicator in one direction and as a brake indicator in the opposite direction.

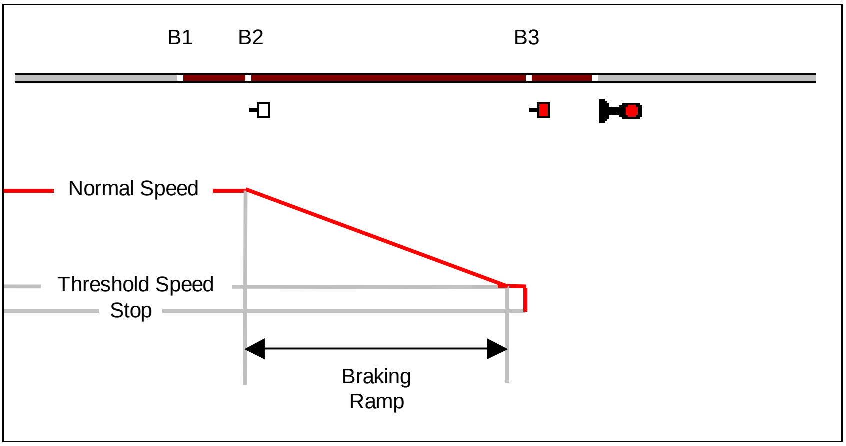


Diagram 88: How Brake and Stop Indicators work – Occupancy Sensors

Diagram 88 shows a block, which is equipped with three occupancy sensors. The left entries to the sensed track sections are marked with B1, B2 and B3.

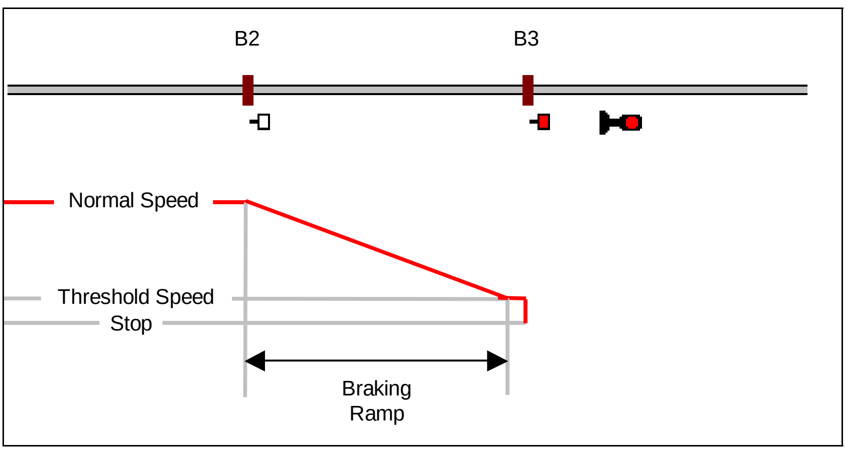


Diagram 89: How Brake and Stop Indicators work – Momentary Track Contacts

An alternative, but for this discussion almost equivalent situation shows Diagram 89. It contains a block, that is equipped with two momentary contacts. These contacts are marked with B2 and B3, too.

B3 is defined as stop indicator ( ) effective for trains travelling to the right. B2 is defined as brake indicator ( ) effective in the same direction. B1, that applies only to the upper diagram, is neither defined as brake nor as stop indicator. B1 is used only for occupancy detection.

The red line shows the speed of the train. It is assumed that the train shall stop in this block, i.e. at B3. When the train enters the block at B1 nothing happens, because B1 is used only to report the entry into the block. When the train reaches B2, it is decelerated to its threshold speed. The braking ramp can be specified individually for each brake indicator. After deceleration the train proceeds at threshold speed until it reaches B3. When the train reaches B3, it is stopped without delay.

If the train does not have to stop in this block, then it passes all indicators without any speed change.

If the stop indicator B3 is missing, then the train will run with normal speed to B2 and stop there without delay. If no stop indicator is assigned to a block, then the first appropriate brake indicator is used as stop indicator. If only the indicator B1 is existing, then the train will already be stopped at B1. If necessary, a train is stopped in a block anyway, even if no brake and stop indicators are assigned.

!

This examples illustrates also, that proper operation of brake indicators requires correct adjustment of threshold speed of each affected train! If this is not the case, the train will be decelerated to an undefined threshold speed. Normally this speed will be too low, to run the train properly and the train will stop when reaching the brake indicator.

It is recommended to locate the sensors which correspond to the brake and especially the stop indicators near the exit of a block, because even long trains should completely fit into a block. If an engine or train passes a sequence of blocks and a certain block is not available or must be passed at restricted speed, then the train is stopped or decelerated in the previous block. Brake and stop indicators control, if a train may exit a certain block and at which speed it must enter and pass the following block. For this reason **TrainController**™ always assumes, that brake and stop indicators are located near the exit of each block with reference to the direction of travel they are effective.

If restricted speed applies in a certain block, then the train is decelerated at the first brake indicator of the previous block. If no brake indicator is assigned to this previous block or a stop indicator is reached first, then the train is already decelerated at the stop indicator. If neither brake nor stop indicators are assigned to this previous block, then the train is stopped when passing the first indicator in this block.

TrainController™ assumes that a train ready to be started is located with its head near the exit of its *current block*. It is also assumed that the train will exit its current block and enter the next block just after being started. For this reason any speed conditions of the first block are ignored and the train is accelerated to the speed, which applies in the second block.

All speed changes take place at the appropriate indicators of the previous block.

When a train enters a block, the Dispatcher checks if there is a route before the next block. In this case, the route is activated if this has not already been done. If the activation is not completed when the train reaches the brake or stop indicator in this block then the train is decelerated or stopped, respectively, in order to wait for the activation of the route. If there is only one indicator in this block, then the same indicator is used for indication of entry into the block, activation of the route and braking or stopping. In this case, the train is always stopped for a short moment because the activation of a route takes some time.

Such short stops can be avoided by adding at least one additional indicator to the block in order to indicate block entry and braking or stopping at different locations. It is possible to use Virtual Contacts for such additional indicators (see section 12.3, "Virtual Contacts").

5.8 Arranging Indicators in a Block

This section describes the different types of sensors and how to use them to operate a block.

Arranging Momentary Track Contacts and Occupancy Sensors in a Block

In the following it is assumed that the track section between the switches in the diagrams displayed in the previous section is a block. Several methods to arrange indicators in a block are discussed below. The pros and cons of each method are outlined as well.

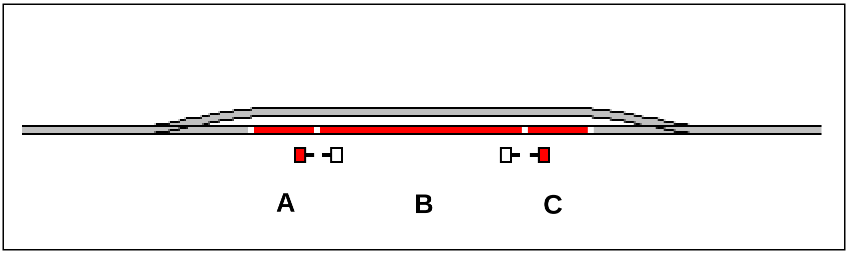


Diagram 90: Block with three occupancy sensors

Diagram 90 shows a block driven by three occupancy sensors. Each of these sensors is associated with a contact indicator in the software called A, B and C. All indicators are assigned to the same block in the software. The block is indicated as occupied as soon as a train enters section A from the left or section C from the right. The block remains occupied until the train leaves the opposite section. The indicator A is additionally used as stop indicator for trains running to the left, C acts as stop indicator for trains running to the right. The trains are stopped at the boundary between B and A or C, respectively. The indicator B is used as brake indicator for both directions. Trains begin to slow down when entering B from either direction. The sections A and C should be long enough, that each train is safely stopped before touching one of the switches. On the other side the longest train should completely fit into the block when being stopped. For this reason the boundaries between B and A or C, respectively, where trains are stopped, must be located close enough to the boundaries of the complete block.

The configuration displayed in Diagram 90 is the optimal solution from a pure technical point of view. The block is indicated as occupied as long as a train is located in one of the three occupancy sections. Additionally it would be even possible to distinguish in which of the three sections A, B or C a train is located. This method is also relatively costly, though, because occupancy sensors are usually relatively expensive and the rails must be cut at the boundaries of each occupancy section.

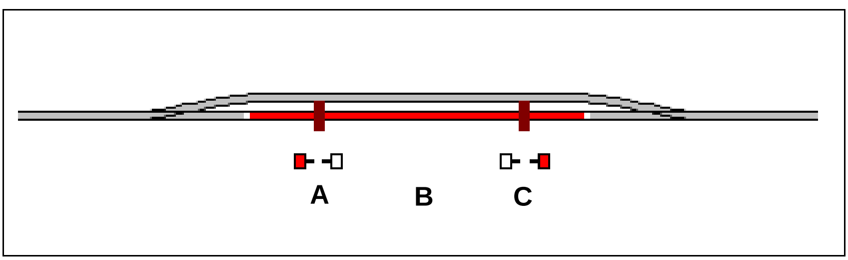


Diagram 91: Block with an occupancy and two momentary sensors

Diagram 91 shows a block driven by one occupancy (B) and two momentary sensors (A and C). Each of these sensors is associated with a contact indicator in the software called A, B and C. All indicators are assigned to the same block in the software. The block is indicated as occupied as soon as a train enters section B from any direction. The block remains occupied until the train leaves section B. The indicator A is additionally used as stop indicator for trains running to the left, C acts as stop indicator for trains running to the right. Both indicators additionally act as brake indicators for the opposite direction, respectively. The location of A and C should ensure, that each train is safely stopped before touching one of the switches. On the other side the longest train should completely fit into the block when being stopped. For this reason A or C, respectively, where trains are stopped, must be located close enough to the boundaries of the complete block.

The configuration displayed in Diagram 91 is usually less expensive then that displayed in Diagram 90, because momentary contacts are usually less expensive then occupancy sensors.

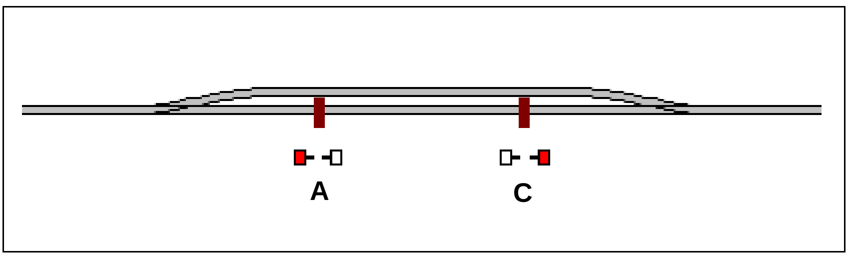


Diagram 92: Simple Block with two momentary sensors

Diagram 92 shows a simple configuration of block driven by two momentary sensors. Both sensors are associated with a contact indicator in the software called A and C. Both indicators are assigned to the same block in the software. The indicator A is additionally used as stop indicator for trains running to the left, C acts as stop indicator for trains running to the right. Both indicators additionally act as brake indicators for the opposite direction, respectively. The location of A and C should ensure, that each train is safely stopped before touching one of the switches. On the other side the longest train should completely fit into the block when being stopped. For this reason A or C, respectively, where trains are stopped, must be located close enough to the boundaries of the complete block.

The configuration displayed in Diagram 92 is very simple and inexpensive but has also some disadvantages. Block occupancy is not indicated. As long as the block is reserved for a train located inside this block this causes no major problem, because the Dispatcher will not allow another train to enter this block. But certain measures are to be taken to avoid premature reservation of this block for another train when a train leaves the block. There is also a disadvantage for passing trains. Let us assume that a train is passing the block from the left to the right and that a route is to be activated before the block ahead, to the right of this block. As soon as the passing train enters the block at A the route is activated. In the same moment the train begins to slow down, because A is also a brake indicator and the train has to wait, until the route is reported to be activated which needs a certain time. This can be avoided by adding an additional contact according to the following diagram:

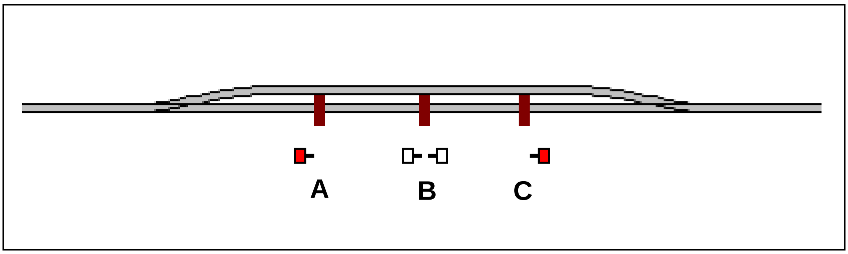


Diagram 93: Block with three momentary sensors

In Diagram 93 the indicator A is used as stop indicator for trains running to the left, C acts as stop indicator for trains running to the right. Indicator B acts as a brake indicator for trains running in both directions. In this configuration block occupancy is not indicated, too, and as in Diagram 92 certain measures are to be taken to avoid premature reservation of this block for another train when a train leaves this block. But trains can pass this block without any speed changes, even if there is a route to be activated before the block ahead – provided the distance between A and B or C and B, respectively, is large enough that the route can be activated after passing A or C, respectively, and before reaching B.

All examples discussed so far can be applied for blocks passed by trains in both directions. The configuration can be made simpler if trains pass a block only in one direction. This is shown in the following:

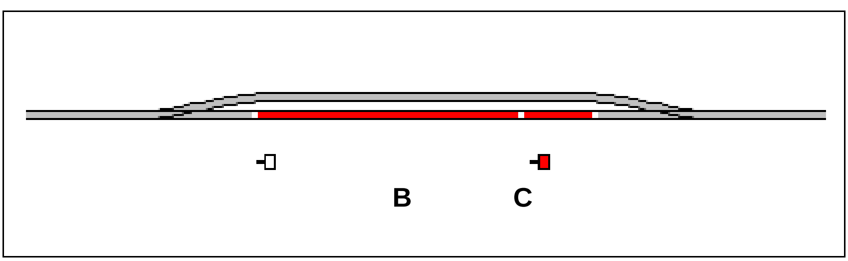


Diagram 94: Block with two occupancy sensors

Diagram 94 has been derived from Diagram 90 by eliminating sensor A. It is assumed that the block is only passed from the left to the right. B acts as brake indicator and C as stop indicator for trains running to the right.

The different configurations discussed in this section are only examples. Configurations similar to Diagram 94 or Diagram 139 can also be made with momentary contacts instead of occupancy sensors or with a mixture of both types similar to Diagram 91. One can think also of other configurations. There is no best way to setup a block. The optimal solution does not only depend on technical requirements but also which equipment you already got and how much you want to spend for new equipment.

In most of the examples discussed in the following sections blocks are represented only by one single indicator. This is done for reasons of simplicity. On an actual layout a block will almost always contain more than one indicator in one of the configurations discussed above.

One Sensor per Block: Combined Brake/Stop Indicators

For your convenience it is possible to replace certain contacts by combined brake/stop indicators. Assume the following configuration:

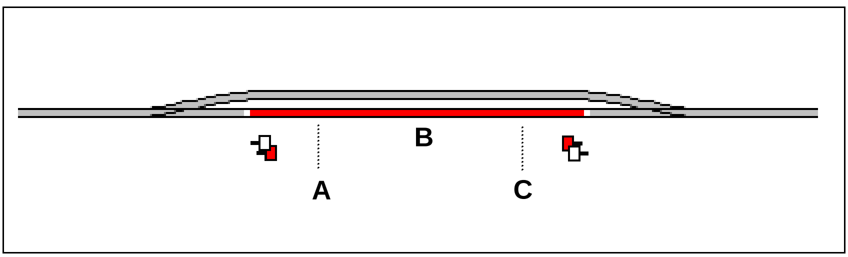


Diagram 95: Block with a combined brake/stop indicator

If your trains are running very precisely, then it is not necessary to mark the stop points A and C with separate indicators. Instead it is possible to specify the occupancy sensor B as combined brake/stop indicator for both directions. A combined brake/stop indicator accumulates the effect of a brake indicator and a stop indicator, that is located in a certain distance behind this brake indicator.

Assume that in the above example the desired stop point C is located in a distance of 50 inches to contact B. If it is desired, that trains decelerate and stop within 50 inches after entrance into B, then contact B can be specified as a combined brake/stop indicator with braking ramp 50 inches. If a train, that has to stop in this block, enters the occupancy section B from the left, it will be decelerated to threshold speed within 50 inches from the left border of B. When it arrives at the point C, which is 50 inches away from the entrance to B, the train will be automatically stopped.

In other words: the combined brake/stop indicator B with ramp 50 inches works exactly like a combination of a brake indicator with a braking ramp of 50 inches plus a stop contact located 50 inches behind this brake indicator.

!

Combined brake/stop indicators allow operation of a complete block with one single indicator symbol.

5.9 Block Signals

General

Traffic Blocking is used on real railroads to prevent two trains from running into each other by dividing the track into sections protected by signals. These signals (here called *block signals*) indicate to an oncoming train whether it can enter the block which begins beyond the signal. If the block ahead is occupied the driver of a train approaching the signal protecting that block sees a red stop light. If the section in front is unoccupied and the train has permission to enter it the driver sees a green signal light. In addition to the signal for the next block the driver is usually also presented with an advance signal which indicates the status of the block beyond that which is being entered. The advance signal indicates green, meaning that the following block is free to be entered; otherwise the block ahead is occupied and the train should proceed into the next block with caution and be prepared to stop at a red light.

When a train is running under control of the *Dispatcher*, **TrainController**™ automatically calculates signal aspects taking into account the availability of *blocks* and *routes* in front of the train. These signal aspects are displayed in the block diagrams and as main and advance signal in the Train Window (see chapter 3, “Train ”). The signals indicate, whether the corresponding block may be left and how the following block must be entered. The *brake* and *stop indicators* assigned to a block take care that a train is stopped accordingly in front of the signal assigned to the <u>same</u> block. Since **TrainController**™ assumes that the brake and stop indicators belonging to a block are located near the exit of the block, this is also assumed for the location of block signals.

TrainController™ already displays the signal aspect currently valid for a certain block, when the first indicator assigned to this block is reached. It is possible to say:

"The engineer is able to see the block signal at the end of a block already when the train enters the block."

Signal Aspects

TrainController™ uses five different signal aspects - each is associated with a specific colour:

Colour	Meaning
Red	Stop
Green	Proceed
Yellow	Proceed Restricted
White	Shunting
Grey	Signal not available

Table 1: Signal Aspects

For each train under its control the *Dispatcher* calculates the aspect of the next block signal and the advance signal. The signal aspect is calculated depending how the train is operated.

During shunting (see section 5.10, "Schedules") white is indicated for all blocks reserved for this train.

When a train is executing a schedule the availability of the next two blocks in front of the *current block* of the train is calculated as main and as advance signal. If the train must not enter the block, then the concerning signal is set to "red". If the train may enter the block, then the signal is set to "green". If the block is available and reserved for traveling with restricted speed, then the signal is set to "yellow".

Similarly the same signal aspects are valid for the *advance signal*, which indicates in advance, whether it is allowed to leave the block behind the *current block* and how the next block behind this advance block must be entered.

"Grey" is used, if the other colors do not apply. This is also the case, if the train is not running under control of the *Dispatcher*.

The calculated state of the main signal of each block is displayed on the according side in the symbol of the block.

Diagram 96: Block Signals

In the example displayed above a train may leave “Southtown 1” and proceed to “Main Line East”. The signal symbol on the right side of the block shows green. The signal on the other side displays red, because it is assumed that the train must not enter “Main Line West”.

The states of the main and advance signal (if available) are additionally indicated in the cab signal controls of the Train Window (see chapter 3, “Train ”), when a train is running under control of the *Dispatcher*.

How to use Signals on the Model Railroad Layout

TrainController™ does <u>not</u> need any signals on your model railroad to control trains. But for realistic operation it should be possible to indicate the calculated signal aspects with appropriate signal models on your model railroad. For this purpose **TrainController™** provides the feature to operate signal models on your layout according to the calculated aspects.

! **These signals are only used for indication. They do not need any facilities to control trains, because the trains are controlled by the Dispatcher.**

It does also not matter, if the used signal models represent main or advance signals, because the models are only used for display. Selecting the appropriate signal model and location you are free to decide, where main and where advance signals will be visible. These signal models are of course operated depending to the *direction of travel*. For this reason you are able to perform the assignment of signals to blocks depending to the direction of travel.

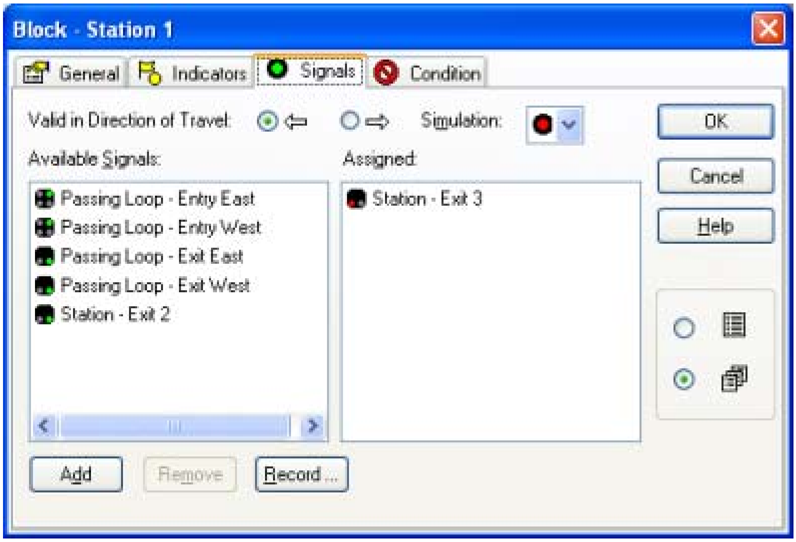


Diagram 97: Assigning Block Signals

How Block Signals Work

The following example shows for blocks A to D, which are subsequently passed by two trains.

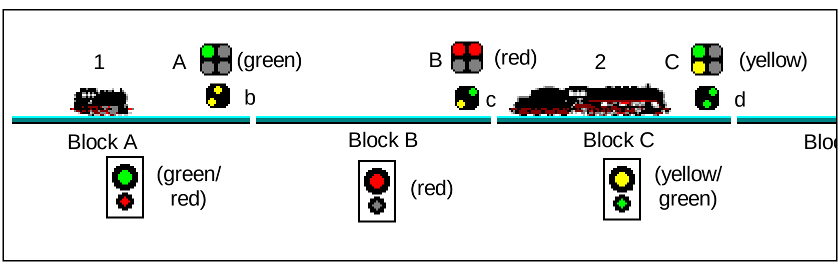


Diagram 98: Block Signals

The internally calculated signal aspects are indicated inside the black rectangles below the block name. These signals are also indicated as cab signals in the Train Window, while the train is inside the concerning block. Above the track there are signal elements controlled according to these aspects. For example the block signal labelled “B” and the advance signal labelled “b” are assumed to be assigned to block B.

Train 1 may enter block B but not block C, because block C is still reserved and occupied by train 2.

The calculated block signal for train 1 in block A is green, because train 1 may leave block A and enter block B without any conditions. This is also indicated by signal A, which is assumed to be assigned to block A.

Since train 1 must not enter block C, the block signal of block B is calculated as red (in this case an advance signal is not indicated in the Train Window). This state is indicated by signal B and as advance signal by signal b. Both signals are assumed to be assigned to block B.

Train 2 may leave block C and enter block D only with restricted speed. For this reason the calculated block signal for train 2 in block C is yellow. This state is indicated by signal C and as advance signal by signal c. Both signals are assumed to be assigned to block C.

The advance signal d is set to green, because it is assumed, that train 2 may leave block D without any conditions.

Additional Notes

!

The internal signaling system of **TrainController™ does not claim to simulate realistic signaling systems of the prototype**. For each block the software only calculates, whether a train may leave this block to the related direction and whether any speed restrictions apply. This calculation is only done for those blocks, that are currently in the focus of an active schedule.

By assigning signal models to blocks the internally calculated aspects can be made visible on the layout if desired.

If a signaling system according to the rules of the prototype is desired, then this can be realized by using the calculated block signals and reservation and occupancy states of related blocks as well as logical associations based on *conditions* and *operations* as outlined in chapter 2, “The Switchboard”.

5.10 Schedules

B

Schedule Diagrams

After drawing your block diagram you will specify the desired train movements. This is done with the help of *schedules*.

Schedules describe how trains travel from certain starting blocks to certain destination blocks.

The base of each schedule is a *schedule diagram*. This diagram contains all blocks, routes and links of the main block diagram, that the train shall use on its journey. This diagram can be displayed on the computer screen, too. This is done by displaying those parts of the main block diagram, that do not belong to the schedule, transparently in the background of the computer screen as shown below:

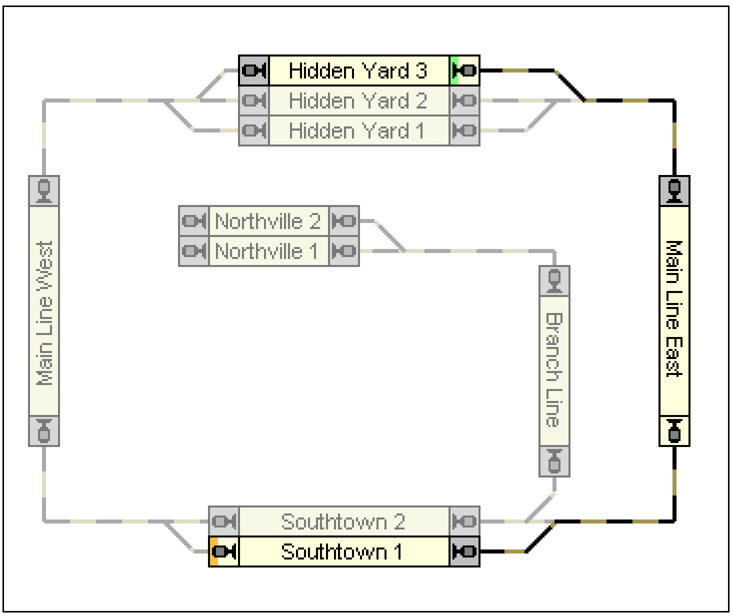


Diagram 99: Schedule Diagram

Diagram 99 shows the diagram of a schedule, that starts in “Hidden Yard 3”, passes “Mainline East” and ends in “Southtown 1”. The blocks, routes and links, that belong to this schedule, are drawn with normal intensity, while the objects, that do not belong to the schedule are drawn transparently in the background. In a specific mode of the software you can easily pick and add them to the current schedule with a click of the mouse to these objects.

Additionally one or more starting blocks and optionally one or more destination blocks are to be specified. Starting blocks are marked in the schedule diagram with a small green marking, destination blocks with an orange or red marking. In the diagram above "Hidden Yard 3" is marked as a starting block and "Southtown 1" is marked as a destination block.

In order to start this schedule, assign an arbitrary train to block "Hidden Yard 3", select the schedule on the computer screen and call the appropriate start command of **TrainController™**. The *Visual Dispatcher* will automatically allocate the blocks and activate the routes, that belong to this schedule and will automatically start the train. When the train reaches the stop indicator in "Southtown 1" then the schedule is terminated.

A schedule can only contain elements, that are also contained in the main block diagram. The location of each element in the display is determined by the location of the referenced element in the main block diagram. If an element in the main block diagram is changed, moved or deleted then this change is reflected in all schedule diagrams. In this way multiple schedules can be conveniently maintained by changes of the main block diagram.

Start and Destination of a Schedule

Each schedule contains one or more starting blocks and one or more destination blocks. Starting blocks are marked in the schedule diagram with a small green marking, destination blocks with an orange or red marking.

! **It is required that you mark the desired starting blocks otherwise the schedule cannot be started.**

After marking one or more blocks as starting blocks, the software automatically calculates one or more destination blocks. This is done according to the following rules:

- Each block which has no subsequent block in this schedule ("dead end") is calculated as destination block. In Diagram 100, for example, "Northville2" is calculated as destination block to the left. If a train executes this schedule, it cannot leave "Northville2" to the left, because there is no other block to the left of "Northville2", that is contained in this schedule.

- Each block which closes a loop is calculated as destination block. On circular layouts usually each starting block is calculated as destination block to the same direction.
- If a block cannot be reached from the starting block, i.e. if there is no incessant chain of blocks, routes and connecting links between the starting block and this block, then this block is excluded from the schedule. Especially it is no destination block.

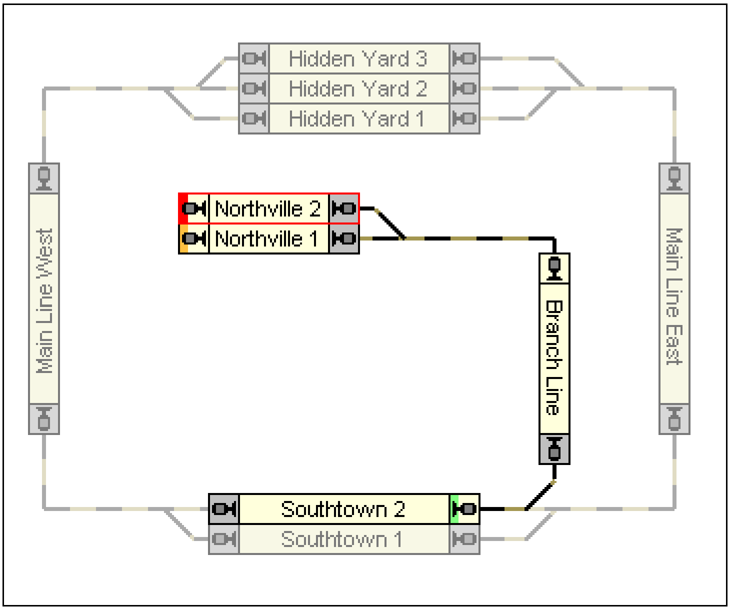


Diagram 100: Calculated and user defined destination Blocks

You can specify additional destination blocks yourself. But you cannot specify calculated destination blocks to <u>not</u> to be such destination blocks. Calculated destination blocks are marked with a small red marking. Destination blocks, that you specified yourself, are always marked with a small orange marking. In Diagram 100, for example, “Northville2” is calculated as destination block to the left. “Northville 1” is explicitly specified as a destination block and therefore marked with a small orange marking.

!

During operation it is only important, whether a block is a destination block or not. It does not matter, whether it is a calculated destination block (red marking) or a user defined destination block (orange marking). The red marking is only used to indicate the calculated destination blocks, that you did not mark as destination blocks yourself.

Start, destination and other schedule specific section settings are entered in the dialog box displayed below.

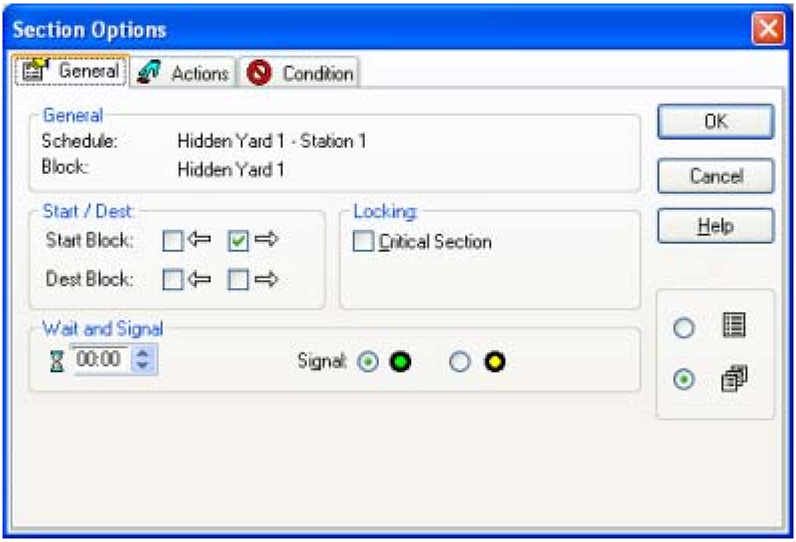


Diagram 101: Schedule Specific Block Settings

Alternative Paths

One of the most outstanding features of the *Visual Dispatcher* is the ease to specify alternatives for the path a train has to take when executing a certain schedule.

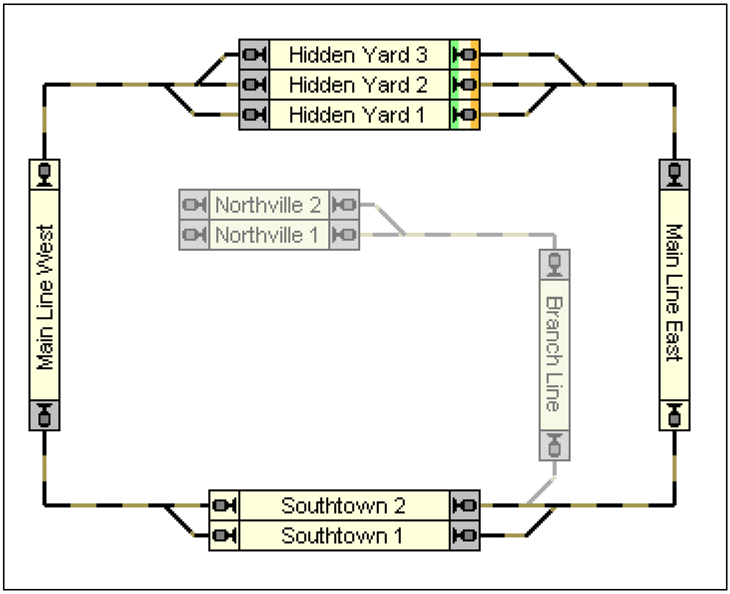


Diagram 102: Schedule Diagram with alternative Paths

Diagram 102 shows a schedule for train movements, that begins in of the three blocks in “Hidden Yard”, proceed on the mainline to clockwise direction, pass “Southtown” through one of the two blocks and end again in “HiddenYard”.

To start the schedule, assign a train to one of the blocks in “Hidden Yard”, select the schedule on the computer screen and call the appropriate start command of **TrainController™**. The *Dispatcher* will automatically allocate the blocks and activate the routes that belong to this schedule and will automatically start the train. If there is more than one train located in “Hidden Yard” and both can be used with this schedule, then one of these trains is automatically selected. It is also possible, though, to pre-select the train yourself prior to starting the schedule.

The *Dispatcher* will also look for an appropriate path through “Southtown” and will select a block in “Southtown” as well as appropriate routes to this block, that are available. If both blocks of “Southtown” are currently available, then the Dispatcher will perform a random selection. In the same way an appropriate block in “Hidden Yard” is selected, when the train approaches the destination.

Even more: each schedule can be started in either possible direction. If the schedule is started in the opposite direction, then the specified destination blocks of the schedule are used as starting blocks and the starting blocks become destination blocks. The schedule of Diagram 102 can be therefore started in counter-clockwise direction, too.

Since the start and destination blocks are identical in this example the trains will start and end in “Hidden Yard”. In Diagram 99, though, a train will start in “Hidden Yard 3” and end in “Southtown 1”, if the schedule is started in the regular direction. Starting the same schedule in opposite direction will cause these two blocks to swap their meaning. “Southtown 1” will become the starting block and the train will end in “Hidden Yard 3”.

The terms *start* and *destination* are mainly used to describe, from where to where the trains travel on this schedule and where trains end. The actual starting block of a train can also be located in the inside of the schedule. In Diagram 102 the *Dispatcher* will try to find an available train in “Hidden Yard” first. But if there is no appropriate train in “Hidden Yard” then the Dispatcher can be instructed to start a waiting train in “Southtown”, if desired. If you select a waiting train in “Southtown” and start a schedule with that train, then the Dispatcher will use this train, even though it is not located in the starting block of the schedule.

The destination blocks are always used as the end point of each schedule. In other words: a train can be started in any block of the schedule and it will always make its way to an appropriate destination block, that can be reached from where it is started.

Looking at Diagram 102 we realize also, that with one single schedule diagram and by picking a few blocks, routes and links from the main block diagram, we can describe all possible train movements in both directions on the main line of this layout.

- The starting blocks of each schedule are to be specified manually.
- Based on the specified starting blocks the *Dispatcher* automatically calculates the destination blocks.
- A block without a link to a ‘next’ block with regard to the direction of travel of this schedule will automatically become a destination block (“Southown 1” in Diagram 99 is an example).
- In order to avoid endless loops on circular schedules each starting block will automatically become a destination block (the blocks in “Hidden Yard” of Diagram 102 are examples).
- It is possible to specify additional destination blocks manually. It would be for example possible to explicitly specify “Southtown 1” as an additional destination block in Diagram 102 , too. If Southtown 1 is available, then each train coming from “Main Line East” will select “Southtown 1” as destination. If “Southtown 1” is not

available, then the train will automatically proceed via “Southtown 2” to “Hidden Yard”.

- It is not possible to reverse a train within a schedule. If, for example, a train enters “Southtown 1” from “Main Line West” then it is not possible to leave “Southtown 1 to “Main Line West” without terminating the current schedule and starting another schedule first. This other schedule can be another run of the same schedule diagram, though.
- It is not possible to change a train within a schedule.

Schedules describe train movements of one train from blocks to other blocks without changes of trains and without changes of direction.

You can create as many schedules as desired.

Schedules are not bound to specific trains, though. In principle each schedule can be executed with each train. In this way by specifying only a few schedules it is possible to achieve varied operation for many different trains. To start a schedule with a certain train, the train must be currently located in a block of this schedule, though.

To run your trains with realistic speed it is very important to adjust the speed profile of each affected engine accordingly (see section 3.5, “The Speed Profile”).

5.11 Execution of Schedules

For varied operation or special situations you can specify among others the following attributes for each schedule:

- If the schedule shall be executed manually or automatically controlled by the computer.
- A time period, in which the Dispatcher repeatedly tries to start the schedule, if the first attempt to start the schedule fails.
- Whether certain blocks or routes of the schedule shall be passed with restricted speed.
- Operations, which are executed at the beginning, at the end or during the schedule.
- Whether and how often the schedule shall be repeated as a cycle or by a shuttle train.
- A selection of other schedules, which are started after finishing the schedule with regard to availability or by random selection

Starting a Schedule

Each schedule can be started during operation of the layout in either of the two possible directions, i.e. from the starting to the destination blocks or vice versa.

When a schedule is being started, then the *Dispatcher* searches the starting (destination) blocks of the schedule until it finds a *current block* of a train, which is not already running on another schedule.

If there is no such block then the *Dispatcher* can optionally continue the search in other blocks, that are located on the path from a starting to a destination block (or back) to find a train that can be started from there. The attributes of each schedule contain an option with which you can specify, whether the *Dispatcher* may start a train from other blocks than the explicitly marked starting (destination) blocks or not.

If no train is found on a block of the schedule or all trains are already running other schedules then the start of the schedule fails. It is possible to specify a time period, in which the Dispatcher repeatedly tries to start the schedule, if the first attempt to start the schedule fails.

A schedule is always started with <u>one</u> train. If the same schedule shall be started with several trains, then the start of the schedule must be executed several times according to the number of trains to be started. This repeated start can be automated by Operations of Buttons and Macros (see section 11.3, “Operations”).

Reservation of Blocks and Routes

When a train is started on a schedule, then the *Dispatcher* tries to reserve at least the *current block* and the next block in front of the train. Also, when a train enters a block, then the block ahead is reserved.

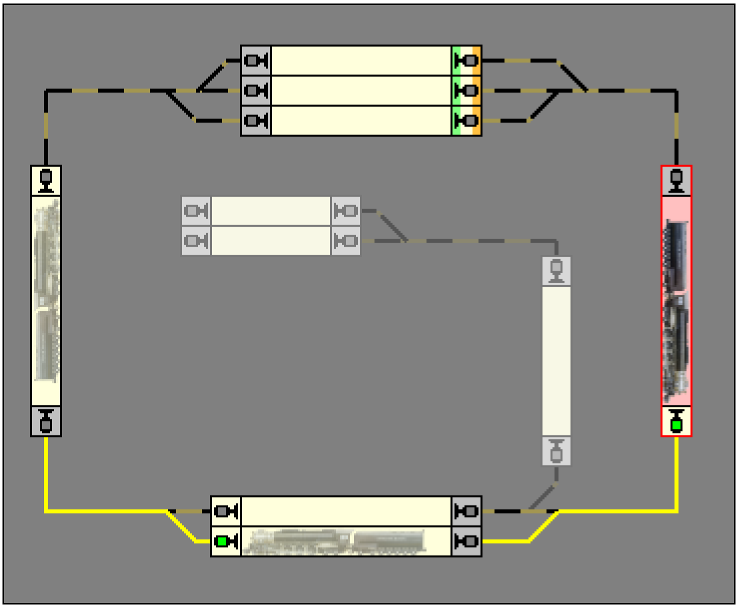


Diagram 103: Reservation of the Block ahead

In the situation displayed above the train has just entered block “Main Line East” (displayed in red). The block ahead is reserved for the train.

If there is a route located between the block and the block ahead, then this route is activated, too. A route is assumed to be located between two blocks, if it connects these blocks in the schedule diagram.

If it is not possible to reserve at least one block ahead of the train or if the route to this block cannot be activated, then the signal at the related block exit is set to red and the train must not proceed.

The *Dispatcher* follows different strategies to reserve the next blocks and routes.

By default, the *Dispatcher* applies a *smart* mode. This means: when a certain block directly ahead of the train is about to be reserved, then the *Dispatcher* checks, whether there is a route behind the block ahead. If this is the case, then this route and the block behind of this route are reserved, too. This is done to reserve and activate the route in time before it comes to unintentional train stops caused by long lasting route activation.

In the diagram displayed above the smart mode is shown. On entry into block “Main Line East” the Dispatcher does not only reserve block “Southtown 2” at the bottom. The Dispatcher also checks, whether there is a route directly behind “Southtown 2”. Since this is the case, this route and the block behind of this route are reserved, too. This is done to avoid unintentional train stops in “Southtown 2” due to the fact, that the train must not leave “Southtown 2” before the route to “Main Line West” is activated.

If “Southtown 2” and “Main Line West” were connected by a link that does not contain a route, then only “Southtown 2” would be reserved in this moment.

! **Smart reservation helps to avoid unintentional train stops caused by long lasting route activation.**

What happens, if “Main Line West” is currently not available in this situation? This is no problem. The *Dispatcher* only tries to reserve the additional route and the block behind of “Southtown 2”. If this is currently not possible, then the train is allowed to proceed at least to “Southtown 2”, though.

It is also possible to instruct the *Dispatcher* not to apply the smart mode to a certain schedule. In this case it is possible to specify a fixed number of blocks, that the Dispatcher shall try to reserve during execution of the related schedule. If, for example, the number of blocks to be reserved ahead is set to 2, then the *Dispatcher* will always try to maintain the next 2 blocks in front of the train reserved for this train. If it is not possible to reserve the specified number, then the *Dispatcher* will allow the train to proceed, though, if at least one block in front of the train is available.

Using a fixed number of 2 as look ahead for the block reservation ensures that the distant signal, which is assigned to the block ahead, always shows a correct state. If it is desired to install a signaling system, that is based on the internally calculated signal aspects of the *Dispatcher*, then this option might be useful, especially if distant signals should be taken into account.

By increasing the look ahead any further for certain schedules it is additionally possible to give certain trains a higher priority. When a train is able to reserve the complete path to the destination during start of the schedule, then it is sure, that it cannot be blocked by concurrent trains during its journey. It has received a high priority to reach its destination.

Path Selection

The *Dispatcher* follows a smart strategy, when it has to select one of several possible paths. In Diagram 102, for example, the *Dispatcher* has to select one of three possible paths, when a train is approaching the “Hidden Yard” from the west or from the east.

In the following the criteria which influence the selection of a path are listed. The following aspects lower the chance of a certain path to be used or might prevent a certain path from being selected at all:

- Other trains, that reserve one or more blocks and routes ahead of the train.
- Locks applied to the entry or exit of certain blocks (see page 106).
- Blocks or routes, that are reported as occupied by unknown objects; more severe, if the rules specified for the schedule do not allow to enter occupied blocks or routes.
- Conditions, that prevent a block from being reserved or a route from being activated (see the following section).
- The distance to an appropriate destination block.
- Superfluous loops.

There are also criteria, that raise the chance of a certain path to be selected:

- Blocks ahead of the train that have been already reserved for this train.
- Activated routes ahead of the train , that are not already reserved by other trains.
- The distance to the nearest obstacle listed in the previous list.

At first the *Dispatcher* evaluates each possible path according to the criteria listed above. Two paths are equivalent with regard to these criteria, if exactly the same aspects apply. If two paths are equivalent, then the Dispatcher performs a random selection.

The criteria listed above do not prevent a path from being selected. They lower the chance of a path to be selected, though, but the Dispatcher might select a path, which is affected by a negative criterion, if there is no “better” alternative.

Special attention should be paid to the distance to an appropriate destination block. If the distances to appropriate destination blocks of two alternative paths are different, then the Dispatcher will probably select the shorter path. If the shorter path is currently locked by an obstacle, then it depends on the difference of these distances, whether the Dispatcher uses the longer path or decides to try to pass through the shorter path in the hope, that the obstacle disappears soon. In other words: the Dispatcher does not select a free path under all circumstances, especially not, if the free path is much longer than other alternatives, that are currently not available.

Release of Blocks and Routes

In general a block or route reserved by a schedule is released when the train has reached a block ahead of this block/route and when this block/route is not indicated as occupied anymore. In Diagram 102, for example, block “Main Line East” is not released before a train coming from “Hidden Yard” has reached “Southtown”. If “Main Line East” is still indicated as occupied when the train reaches “Southtown”, though, then release of “Main Line East” is further delayed until the occupancy indication of “Main Line East” is turned off.

In detail the following rules apply:

- A block is assumed to be reached, when the train reaches a stop indicator assigned to this block.
- An occupied block or route is not released. (An exception of this rule is outlined below.)
- A block or route is not released until the train has reached a block behind of this block/route.
- When a train reaches a certain block all not occupied blocks/routes located before this block, but not located behind another occupied and reserved block/route, are released. If, for example, “Main Line East” in Diagram 102 is still reserved and occupied when the train reaches “Main Line West”, then the used block of “Southtown” is not released, regardless whether it is occupied or not. If both, “Main Line East” and the related block in “Southtown”, are not occupied when the train reaches “Main Line West”, then both blocks are released.
- When the train reaches the destination position of the current schedule, i.e. the stop indicator in a destination block of this schedule, then all blocks and routes apart from this last block are released, regardless whether they are occupied in this moment or not.

Simulation of Train Movements without Connection to a Model Railroad

If no digital system is connected, then it is possible to simulate the stimulation of track sensors by clicking to the related contact indicators on the computer screen with the mouse. In this way you can perform a trial run of a train on a schedule before the model railroad is connected.

A very convenient way to simulate train movements is the use of the *Traffic Control*. The indicators currently visible in the *Traffic Control* can be turned on and off by clicking to them with the mouse, too, if the layout is not connected (see also chapter 6, “The Traffic Control”).

Restricted Speed, Wait Time and additional Operations

For each *block* in a *schedule* you can specify, if the affected section is passed with *restricted speed* or not.

Additionally you can specify a *wait time* for each block contained in a schedule.

Finally it is possible, to assign a set of Operations to each block of a schedule. Possible operations are turning on or off an *engine function* (see section 3.6, “Headlights, Steam and Whistle”) or execution of a *macro* (see section 11.5, “Macros”) in order to perform more complex actions.

These operations can optionally be performed when

- the train enters the block
- the train reaching a brake indicator has to reduce its speed
- the train has to stop
- the train starts again after a stop
- the block is released after the train has left the section

Additionally it is possible to perform operations before starting or after finishing the schedule.

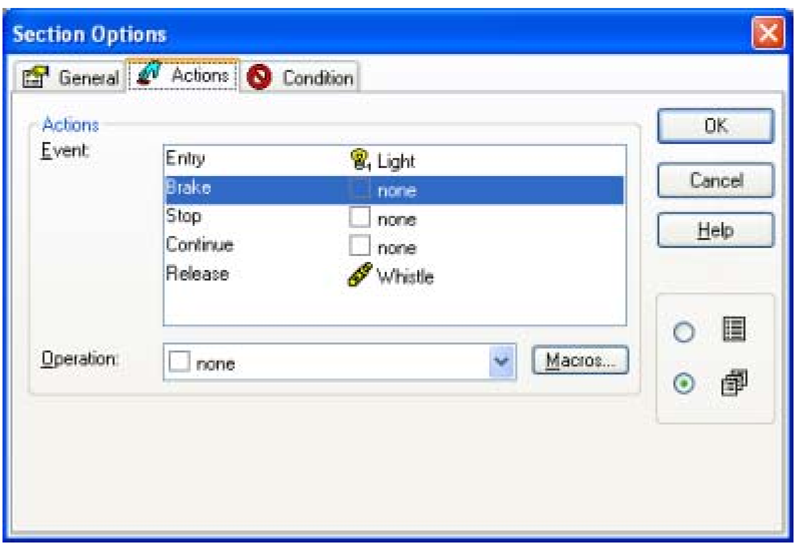


Diagram 104: Specifications of the Section of a Schedule

In the example displayed above each train entering the related block shall turn on the light. Additionally it shall blow its whistle when the block is released later.

If a function symbol specified here is not configured for an engine, then this engine will do nothing, when it executes this schedule. If, for example, the function symbol *Whistle* is only assigned to steam engines in the example displayed above, then diesel engines will remain quiet when executing this schedule.

Speed restrictions and wait time are specified in the dialog displayed in Diagram 101.

These attributes are specified on a per-schedule base. It is possible to specify different settings for different schedules.

Type of a Schedule - Shuttle and Cycle Trains

There are different types of schedules.

Normally – when no special type is selected – the journey of the train ends in a destination block of the schedule.

If a train shall repeat the schedule as *shuttle train*, it will be started again after arriving in a destination block and will run back in the opposite direction to an appropriate start block. It is possible to specify a repeat count to control, how often the schedule shall be repeated.

It is also possible to repeat the schedule as a *cycle* based on a circular diagram. In this case the train is started again on the same schedule after arriving at the destination block of the schedule. The train repeats traveling on the same schedule in the same direction as before. Like for shuttle trains it is possible, to specify how the cyclic schedule shall be repeated.

! **When repeating schedules as a cycle it is necessary that these schedules are circular, i.e. destination blocks must be start blocks, too.**

Shunting

An additional type of schedules is *shunt*.

If a schedule is provided for shunting then all *blocks* and *routes* of the schedule are reserved, when the *Dispatcher* starts the schedule. The blocks can be passed in an arbitrary direction. Shunting trains are operated manually and it is also allowed to reverse a train while shunting and to leave a block in the opposite direction. The *Dispatcher* does not intervene; it takes care only, that other engines or trains under its control do not enter the blocks reserved for the shunting train.

If a schedule is provided for shunting, then <u>all</u> blocks contained in schedule are reserved, when a train is started. Since each block can be reserved for only one train, at most one train can run simultaneously on this schedule.

Running Trains manually under Control of a Schedule

For each schedule you can specify its *mode*. If desired you can control engine and trains on the schedule completely manually. In this case the *Dispatcher* reserves the blocks, activates the routes and calculates the block signals. You are – like a real engineer – responsible for obeying the indicated signals and following the speed conditions. But it is also possible to transfer the control over the schedule completely to the *Dispatcher*. In this case all engines and trains on this schedule are operated automatically. Finally it is also possible to share the engineer's job with the *Dispatcher*. In this way it is for example possible, that the train is running under your manual control, but that the *Dispatcher* is able to intervene to stop a train in front of a red signal.

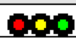
Mode	Explanation
	Trains are completely controlled by the Dispatcher
	Dispatcher intervenes when restricted speed is prescribed or in order to stop a train
	Dispatcher intervenes in order to stop a train
	Trains are completely controlled manually
	Schedule is used for shunting

Table 2: Mode of a Schedule

It is possible to use different modes for different schedules, regardless whether these schedules share the same blocks and routes or not. This enables full automatic operation of one part of your layout, running trains in a second part manually under control of the Dispatcher, operation of a third part as shunting area under control of the *Dispatcher*, and finally a fourth part outside the control of the *Dispatcher*.

Different schedules with different modes can be arranged for the same part of your layout. It is for example possible to create two schedules for the main track of your layout. The first schedule is used for automatically running trains, while the second schedule uses the same track for trains operated manually under control of the *Dispatcher*. In this way you can operate your favorite train manually while other trains in front of or behind this train are controlled automatically.

5.12 AutoTrain – Start of Schedules made Easy

B

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 before.

AutoTrain™ is especially useful in the following cases:

- If a train shall automatically run somewhere during operation and you did not specify an appropriate schedule before to perform this task.
- If you want to define a new schedule quickly from scratch.

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

- Press and hold the key ‘A’ on your computer keyboard (A = *AutoTrain*™).
- Press the left mouse button near the exit of the block in the block diagram or in the switchboard, where the train shall 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 shall stop.
- Release the left mouse button and the ‘A’ key.
- The train will now start and run automatically to the destination block.

With the *AutoTrain*™ toolbar you have more options for individual customization before the train is actually started. To run a train with the *AutoTrain*™ toolbar the following steps are performed:

- Open the *AutoTrain*™ toolbar .
- Select the locations (blocks) on the layout, where the train shall start.
- Select the locations (blocks) on the layout, where the train shall stop.
- Optionally specify additional options that influence the execution of the *AutoTrain*™, such as wait time, operations, cycle, shuttle, etc).
- Start *AutoTrain*™.

Diagram 105: AutoTrain Tool Bar

After start *AutoTrain*™ automatically tries to find a path from the specified start block to the specified destination blocks. If a train is located in the start block, then it is automatically started to run to the selected direction.

A started *AutoTrain*™ is very similar to a schedule which is currently executed. It has one starting block and one ore more destination blocks, that are selected before *AutoTrain*™ is started.

There are some additional options:

- After selection of the start and destination blocks you can let *AutoTrain*™ try to find a path from the start to the destination blocks without starting a train. This is useful in *edit mode*, especially if no train is located in the start block. This is also useful if you want to check the resulting path before actually starting the train. Together with another option, that allows to store the current *AutoTrain*™ as a permanent schedule for later use, this is a very fast method to create new schedules by letting the software calculate the appropriate paths for you.
- It is possible to select certain blocks to be included in the schedule anyway prior to start the search for an appropriate path. Each found path will then pass these blocks, if possible. This gives you more control over the resulting path.
- It is also possible to exclude certain blocks from *AutoTrain*™ prior to start the search for an appropriate path. This gives you additional control over the resulting path.
- You can also specify, whether only the shortest possible paths from the start to the destination blocks shall be taken into account or all possible paths.
- Additionally it is possible to limit the search to a maximum number of blocks. This option is useful in case of large or complex layouts and slow computers, where the search may take a certain while. Limiting the maximum number of blocks prior to starting the search can dramatically reduce the time needed to find the path.
- While an *AutoTrain*™ is active you can also store it as a schedule to execute it later, e.g. as part of a time table.

***AutoTrain*™ requires the prior creation of a full-functioning main block diagram.**

AutoTrain™ follows the same rules with regard to the inclusion of blocks or routes as regular schedules. That means: like it is possible to include blocks or routes, that are currently locked, reserved, or occupied etc. into a schedule during *edit mode*, it is also possible that *AutoTrain*™ includes blocks or routes, that are currently locked or reserved by another train. In this way it is possible to let *AutoTrain*™ create a schedule for later use, that contains blocks or routes, that are currently not available. The only way to prevent certain blocks or routes from being included by *AutoTrain*™ is to explicitly exclude them prior to start the search. Even though it is possible to include blocks or routes, that are currently not available, into *AutoTrain™* before the train is started, the train will not enter such blocks or routes after it has been started, just like in a regular schedule, too.

5.13 Successors of a Schedule

For each schedule it is possible to specify a set of other schedules, of which one is started, after the schedule is finished.

As explained in the following examples successors are used in many applications.

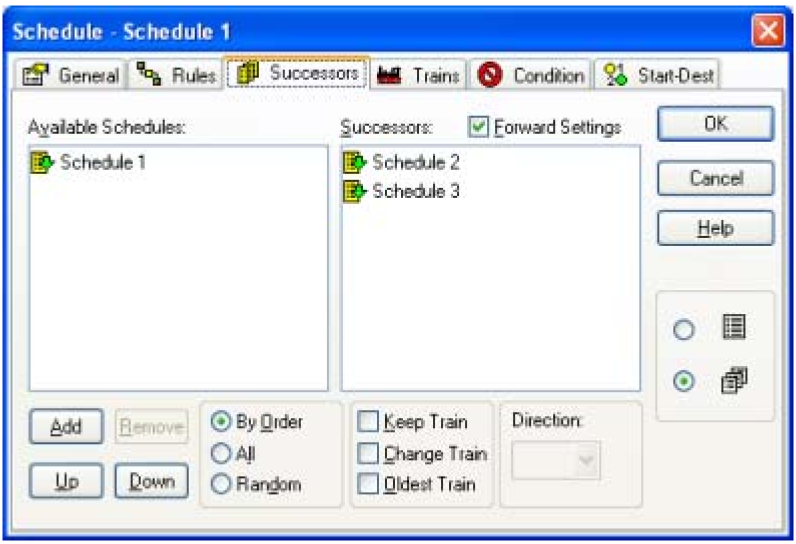


Diagram 106: Successors of a Schedule

Several options allow you to specify how control of the train is passed from a schedule to its successor.

It is possible to specify that the successor is selected **by order** or **randomly**. Additionally you can select to **keep the train**, i.e. to enforce that the successor continues with the same train as before, to enforce a **train change** or to continue with the **oldest train**, this is the train, which was not operated by another schedule for the longest time. If none of these options is selected then any available train is used. This can be the same train as before or another train.

In conjunction with the possibility to select successors randomly it is possible to control a hidden yard automatically. A train arriving in a hidden yard can be enabled to select another waiting train, which will leave the hidden yard.

If it is intended to start the successor with the same train, then it is recommended, that the successor starts with a destination block of the previous schedule. In this block the control of the train is transferred to the successor.

It is also possible to specify that **all** listed successors are started. These successors are started simultaneously, when the previous schedule is about to be terminated.

If several schedules shall be executed in a sequence, e.g. schedule 2 shall be executed after schedule 1 and schedule 3 shall be executed after schedule 2, then schedule 2 is to be specified as successor of schedule 1 and schedule 3 as successor of schedule 2.

Since it is not possible to reverse a train or to change trains during the execution of a schedule successors must be used if

- a train shall be reversed
- trains shall be changed

5.14 Schedule Selections

Sometimes it is desirable to select one of several schedules. This is supported by *schedules selections*. A *schedule selection* enables the selection of certain schedules out of a selection of several other schedules. Even though there is schedule diagram associated with a schedule selection such selection can be started like any other normal schedule. It can be used wherever a normal schedule can be used. When a schedule selection is started then one or more of the schedules contained in the selection are selected and started. This selection may also include other schedule selections.

6 The Traffic Control

B

During operation of a layout the *Traffic Control* shows the status of the currently selected train, block or route and the current status of the indicators, that have been assigned to the current object.

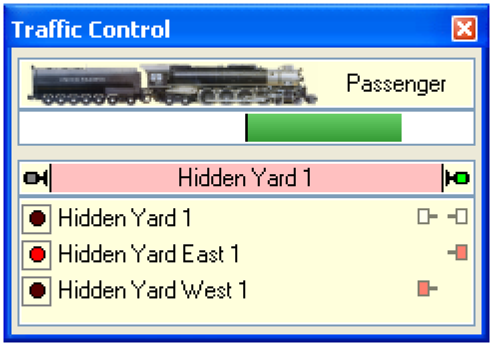


Diagram 107: Traffic Control

Here all important information about the currently selected train and its current location congregates. When you select a train on the computer screen, then this train and the block, where it is located, are displayed here. When you select a block or a route, then this block/route and the train, which is currently there, if any, are displayed.

The speed of the train is made visible with a colored rectangle. The status of the block, whether occupied or not, and the status of the block signals on both exits are displayed as well.

Additionally the indicators, that have been assigned to the block or to the route, are displayed. The status of each indicator, whether occupied or not, and the usage of each indicator as a brake or stop indicator for a certain direction are displayed here, too.

If the digital system, to which these indicators belong, is running in offline mode, then you can toggle the state of each indicator by clicking to it with the mouse. In this way the movements of trains can be simulated very conveniently: simply select the block that you want to look at on the computer screen and click to the occupancy, brake or stop indicator to simulate what happens if a train passes this indicator. Please refer also to page 138 for further details about simulation.

7 The Inspector

B

The *Inspector* helps to have an overview of the objects of your model railroad - this is especially very useful in case of large layouts with many *turnouts, signals, routes, engines, trains, blocks, schedules,* etc. The Inspector clearly displays the properties of the currently selected object. The references to other objects (for example turnouts in routes or blocks in schedules, etc.) are visible, too. With a click it is possible to skip to other referenced objects, to view their properties. Important attributes like the name or digital address of objects can be edited directly in the Inspector without the need to go through separate dialog boxes.

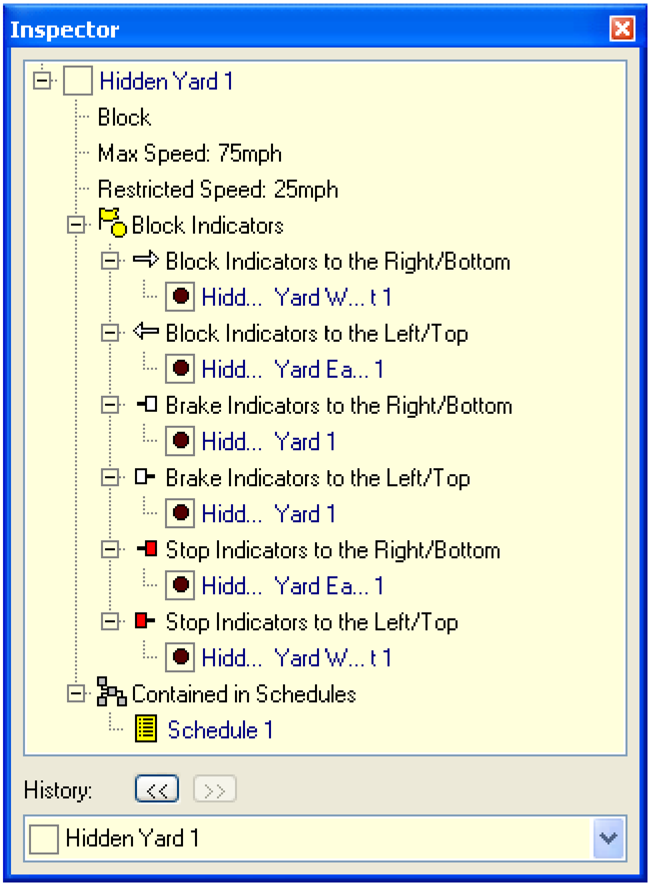


Diagram 108: Inspector

8 The Message Window

B

With the *Message Window* you can keep yourself up to date about the events occurring in **TrainController™** while operating your model railroad with the computer. In certain situations **TrainController™** displays informative, warning or error messages in the *Message Window*.

Most of these messages are generated by the *Dispatcher* (see chapter 5, “The Visual Dispatcher”). A special mode enables displaying of additional informative messages, which are useful to search errors during creation of your automatic control system with the *Dispatcher*.

Using *system operations* (see page 168) it is additionally possible to display user defined messages in the *Message Window*.

The different types of messages are marked with different symbols.

Symbol	Meaning
	Informational message. This type of message is often displayed, when a certain operation has been completed successfully.
	Warning. The related action is performed, but certain problems may occur.
	Error. The execution of the related action is aborted.
	Fatal error. This message is for example displayed, when an object needed to perform the current action, has been deleted by the user. Normally a user intervention is necessary, to correct the data.
	Planned wait.
	An engine or train is ready to be controlled manually.
	Custom message – generated by *system operation*.
	Diagnostic message. Messages of these type can be optionally displayed to ease the search for errors when the control system is created.

9 A Sample Layout

B

General

The layout displayed below shall be operated with **TrainController™**:

Diagram 109: Sample Layout

The layout has two stations: "Southtown" located on the left side of the layout and "Northville" located at the end of the branch line. There is an additional hidden yard that is covered by the mountain.

This can be seen better in the track plan displayed below:

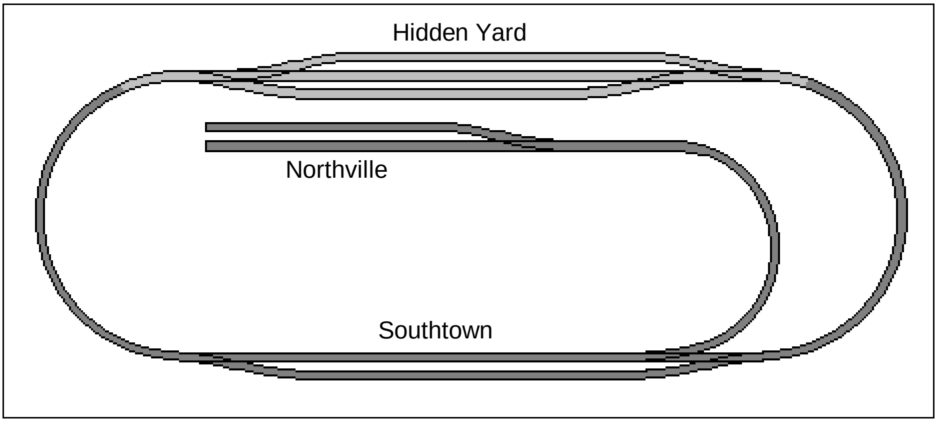


Diagram 110: Track Plan of the Sample Layout

The main line, i.e. the loop that connects "Hidden Yard" and "Southtown", will be operated automatically under full control of the *Dispatcher*. The branch line from "Southtown" to "Northville" will be operated manually.

In the following the necessary steps to control this layout are explained. **TrainController™** is installed with a set of sample files called STEP1.YRR to STEP6.YRR. Each of these file corresponds to the content of one of the following sections. By loading these files into **TrainController™** you can reconstruct yourself, how the particular steps are performed.

Step 1: Creating the Switchboard

The first steps are creation and drawing of the *switchboard*.

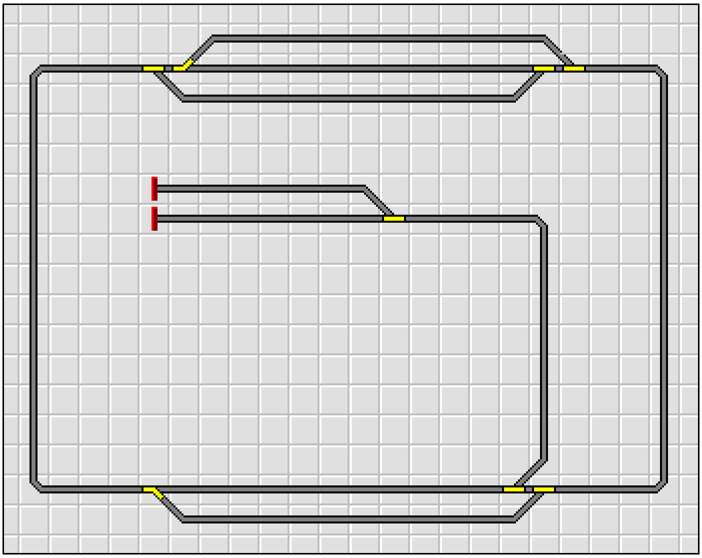


Diagram 111: Switchboard Southtown

Diagram 111 shows the switchboard of the sample layout. All switches get appropriate names. The related digital addresses are assigned, too.

At this stage we are able to control all switches on our sample layout.

Step 2: Defining the Engines

Our switchboard is now completed and we are going to create the entries for the engines that we want to run on the layout. We want to run three trains, a passenger and a freight train that can run on the main line only, and an additional train that can go to Northville, too. The trains are entered into the Train Window as displayed below:

Diagram 112: Engine list

By editing the properties of each engine we assign a digital address to each engine and can additionally specify engine functions, measure the threshold speed and the speed profile and edit other properties. This is not outlined in detail here, because it is not important for understanding of this sample layout. Further details can be found in chapter 3, “Train Control”.

The images have been prepared with **TrainAnimator**™.

Through the **Window** menu of the software you can open additional Train Windows, if you like to control each train through a separate Train Window.

At this stage of the sample we are able to control our trains manually with the computer on all parts of the sample layout.

Step 3: Creating Blocks

At first we divide our layout into logical blocks. We follow the guidelines on page 96. The resulting block structure looks as follows:

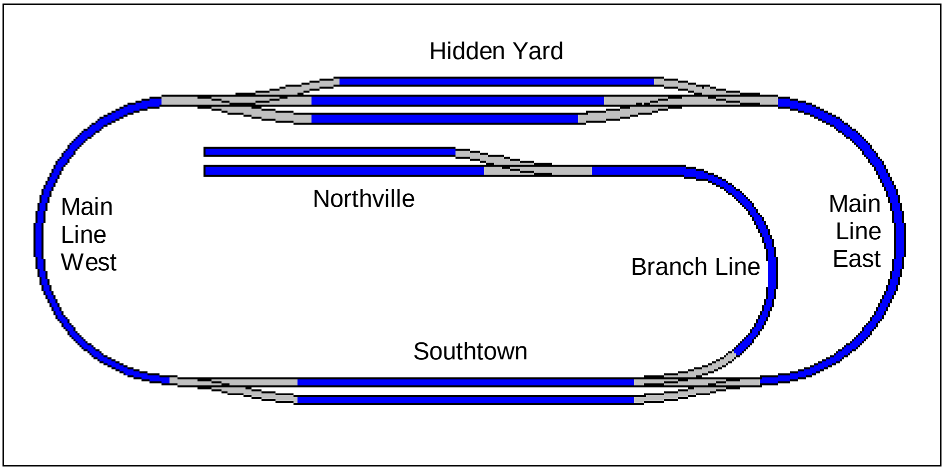


Diagram 113: Block structure of the sample layout

Each blue track section represents a separate block.

Based on this diagram we insert a traffic box for each block into the switchboard. The resulting switchboard is displayed in the next diagram:

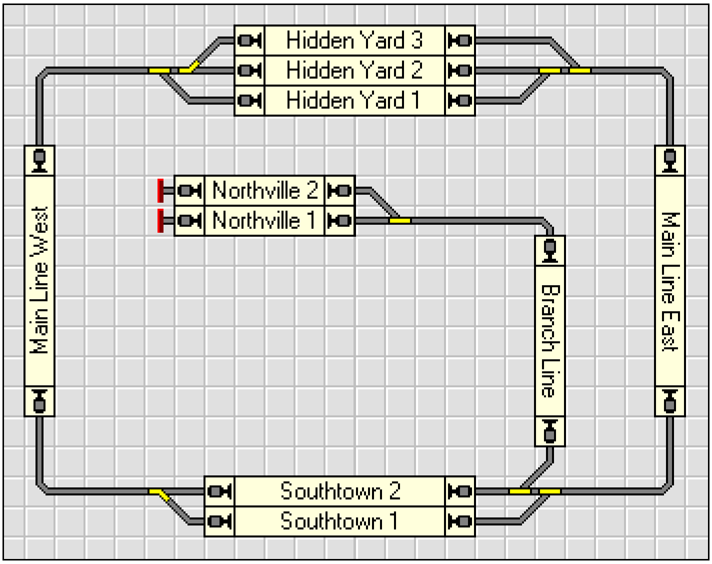


Diagram 114: Switchboard with Traffic Boxes

Based on this switchboard the *Visual Dispatcher* automatically calculates the following main block diagram:

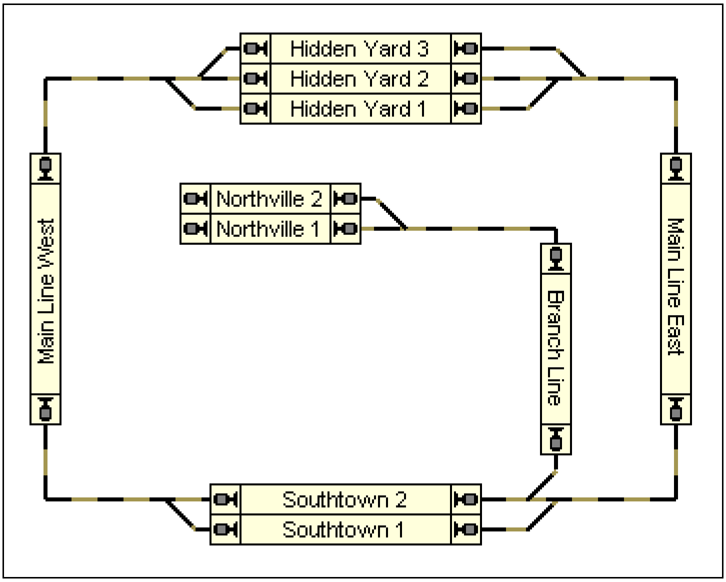


Diagram 115: Main Block Diagram in the Visual Dispatcher

Please note that the block diagram represents the track layout in rough outline. The actual track connection between "Main Line West" and "Hidden Yard 3", for example, contains two switches. These switches are not drawn in the block diagram in detail. Instead a route between both blocks is created.

All necessary routes between all blocks are created and recorded automatically.

Step 4: Contact Indicators

We want to equip each block on the main loop with three occupancy sensors. The arrangement of indicators of each block follows Diagram 90 (please refer to page 118). The occupancy sensor in the center of each block (dark red zones in Diagram 116) will be used as brake indicator for both directions; the sensors on both sides of each block will be used as stop indicator for the related direction (light red zones in Diagram 116).

The branch line to "Northville" contains 3 blocks. Since we do not want to run automatic trains there it is sufficient to install one occupancy sensor in each of these blocks for train tracking of manual trains.

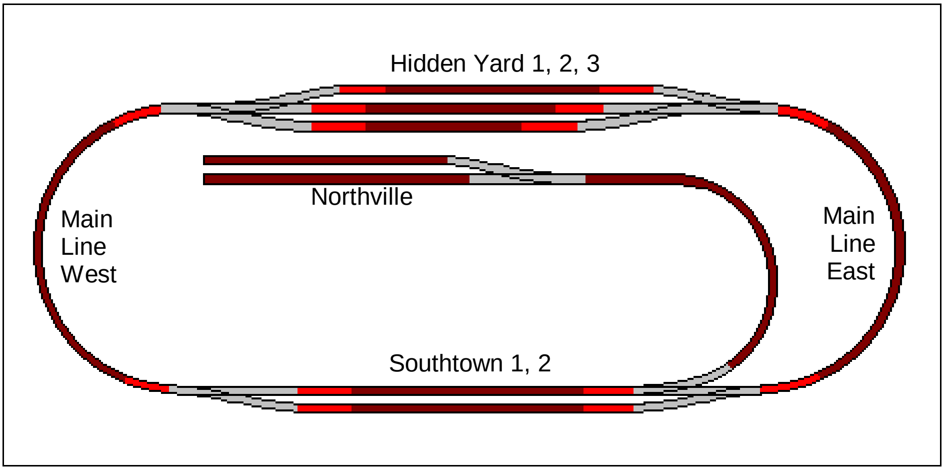


Diagram 116: Indicator arrangement of the sample layout

The grey tracks in Diagram 116 are not contained in any block. They are part of routes, which are assumed here to be located between the blocks.

Indicators are created for each block according to the following table:

Block	Indicator	Usage
Hidden Yard 1	Hidden Yard 1	
	Hidden Yard East 1	
	Hidden Yard West 1	
Hidden Yard 2	Hidden Yard 2	
	Hidden Yard East 2	
	Hidden Yard West 2	
Hidden Yard 3	Hidden Yard 3	
	Hidden Yard East 3	
	Hidden Yard West 3	
Main Line East	Main Line East	
	Hidden Yard East Entry	
	Southtown East Entry	
Main Line West	Main Line West	
	Hidden Yard West Entry	
	Southtown West Entry	
Southtown 1	Southtown 1	
	Southtown East 1	
	Southtown West 1	
Southtown 2	Southtown 2	
	Southtown East 2	
	Southtown West 2	
Northville 1	Northville 1	
Northville 2	Northville 2	
Branch Line	Branch Line	

Table 3: Indicator Configuration

The small icons indicate in which direction of travel a certain indicator is effective as brake or stop indicator. The indicator “Hidden Yard 1”, for instance, marked by and is used as brake indicator of block “Hidden Yard 1” for both directions of travel.

The indicator “Southtown East Entry”, marked by  is used as stop indicator of block “Main Line East” for trains that pass this block from the top to the bottom of the layout, i.e. from the Hidden Yard to Southtown. For trains running to the opposite direction this indicator reports that the train enters the block.

Step 5: Creating Schedules

One single schedule is sufficient to describe all train movements on the main line of the sample layout:

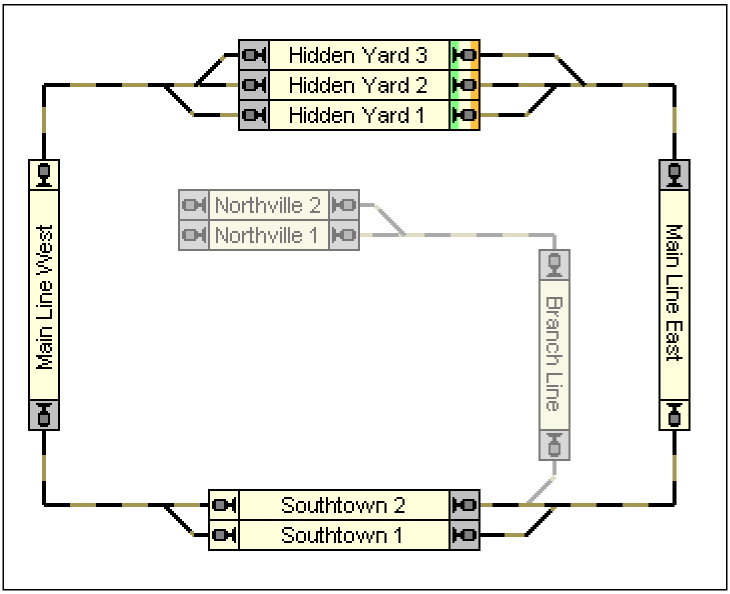


Diagram 117: Schedule Diagram of the Sample Layout

The blocks in “Hidden Yard” are marked as start blocks of the schedule. Since the schedule forms a closed loop these blocks are automatically calculated as destination blocks, too. The schedule can be started to both directions, i.e. trains can run clockwise or counter-clockwise under control of this schedule. Depending on a specific setting of this schedule, it allows either for train movements that start in Hidden Yard or for train movements that start in any other block of the main loop. All train movements will end in “Hidden Yard”, though.

Manual Operation

The branch line from “Southtown” to “Northville” and back shall be operated manually.

All precautions for train tracking have been already done by integrating the blocks of the branch line into the main block diagram accordingly.

Trains waiting in “Southtown” and bound to “Northville” will release block “Southtown 2” as soon as they left “Southtown”. They will be automatically tracked to “Northville” and back. All is done by proper drawing of the main block diagram, no further measures are to be taken. A train that comes from “Northtown” and arrives in “Southtown” will automatically reserve block “Southtown 2” again.

From there it can be started by the schedule shown in the previous section and automatically travel to the “Hidden Yard”. This can be done even automatically on arrival in “Southtown” without further intervention by the human operator.

Further Steps

Now the framework is completed to add varied operation.

It is for example possible to assign the schedule to the operations of a push button symbol in a switchboard in order to start automatic operation manually when desired.

Another schedule can be added with “Southtown 2” as destination block. This schedule will lead a train to “Southtown 2” where it can be taken over to perform a manually operated travel to “Northville”.

It is also possible to start the schedules created here according to a time table.

A hint in case you have configured endless automatic operation: by adding a global on-off switch symbol located somewhere in a Switchboard to the conditions of some or all of your schedules you can implement a global power off mechanism. Lets assume that the schedules can only be started, if this global on-off switch is on. If this switch is turned off during automatic operation then all trains will finish the run of the current schedule and will not start another run of any schedule that is restricted by this on-off switch. In this way you can smoothly shut-down your automatic operation in a very clean way.

Such and more advanced techniques will be explained in the second part of this Users Guide.

Part III

Extensions

This Part III of the Users Guide explains the extended features of **TrainController**™. These features enable advanced users to make professional use of all possibilities of the software.

Novice users should focus to the previous Part II first and should not read any further, before they put the content of Part II into practice. With the possibilities outlined in Part II you can control your complete layout manually and perform basic automatic operation of your trains.

10 The Clock

TrainController™ can display a fast clock on your computer screen. Using a fast clock time spaces are stretched artificially. This simulates more realistic timing.

Diagram 118: The Clock Window

The clock is used to perform a timetable based operation with the *Dispatcher* (see chapter 12, “ The Visual Dispatcher II”). It is also used for realistic simulation of inertia when a train is accelerated or decelerated. *Simulated distances* are calculated using the clock, too.

Additionally the clock provides a perpetual calendar, with which an arbitrary date between 1830 and 2030 can be selected. In this way it is possible to play in your favorite epoch and to run different timetables, for example varying between working days or holidays.

The clock is permanently active and runs always in the background of the program. If desired you can display the clock on the computer screen. If the clock is visible it can be stopped, if desired, or its settings - such as *scale factor*, *current time* or *date* - can be changed.

An additional useful feature is skipping time intervals without operation. If you run a timetable in which no trains are running at night then you can skip such a period. In this way you can shorten those intervals as desired.

11 Indicators and Semi-Automatic Control

With the mechanisms outlined in this chapter you can extend manual control to semi-automatic control of your layout with switchboards. Additionally some of the mechanisms explained here can be applied to the *Visual Dispatcher* or can be used to influence automatic control individually.

This is the reason why they are discussed in a separate chapter.

11.1 The Memory of Indicators

In the simplest case, an *indicator* is automatically turned on and off by the associated track contact or occupancy sensor (see chapter 4, "Contact Indicators"). Additionally, *indicators* provide a *memory* in which the event, that has occurred, can be "stored" for a longer period.

For this purpose you can select one of the following methods to turn off the *indicator*:

- **Automatic:** this is the default method. In this case the *indicator* is automatically turned on and off by the associated track contact or occupancy sensor.
- **Manual:** in this case, the *indicator* remains turned on until you turn it off – by clicking to it with the mouse.
- **Timer:** in this case, the *indicator* remains turned on for a certain period. This can be used to reset a signal a few seconds after a train has passed it.
- **By Train:** if this option is selected, then the *indicator* remains turned on until a train has passed the associated track contact or occupancy sensor or another point on your layout. With this option it is for example possible to use a momentary track contact for *Virtual Occupancy Indication*.
- **By Indicator:** if this option is selected, then the indicator remains turned on until *another indicator* is turned on.
- **Toggle:** if this option is selected, then the indicator is alternately turned on and off. With this option, it is possible to create a track occupancy detector with two momentary track contacts. This is explained in more detail in the example on page 176 "Simple Track Occupancy Detection".

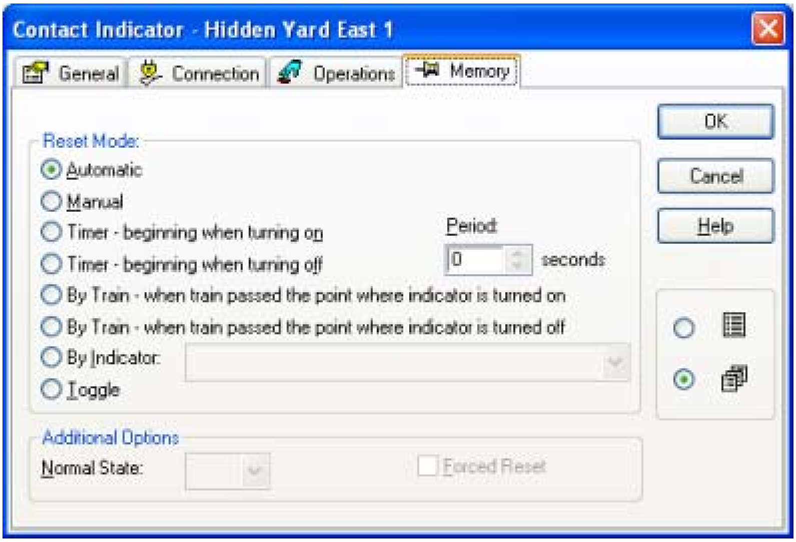


Diagram 119: Memory of a an Indicator

Normally, the *indicator* is turned off when the condition that causes the activation of the indicator no longer applies, e.g. if **Timer** is selected, and the specified time period has passed, then the *indicator* is turned off only if the condition does no longer apply. If the *condition* is still valid, then the *indicator* remains turned on even if the specified time period has passed. Sometimes it is useful to turn off the *indicator* regardless of the current state of the *condition*. For this purpose the additional option **Forced Reset** can be selected.

Example: Preventing an Indicator from Flickering

In the following example it is assumed that a certain momentary track contact is triggered by each axle of a passing train. It is shown how the indicator symbol can be prevented from flickering. Finally the indicator will be turned only once by a passing train.

- Create a *contact indicator* and link it to the momentary track contact.
- Set the *memory* of the indicator to **Timer 2 Seconds**.

	Memory
Indicator	Reset: after 2 seconds

Table 4: Preventing an Indicator from Flickering

When the first axle of a passing train touches the track contact, then the indicator is turned on. When this axle leaves the track contact, then the indicator remains turned on until the 2 seconds have passed. If the next axle of the train touches the track contact before the timer expires then the indicator will remain turned on for another 2 seconds and so on. The indicator is turned off when no further axle of the train touches the track contact, i.e. when the train has passed the contact completely. In the software the indicator is turned on only once regardless how much cars and axles the train contains.

11.2 Protection and Locking with Conditions

In addition to the locking mechanisms provided by *routes*, there are even more possibilities for locking and protection. It is possible to restrict the operation of *switches*, *signals*, *accessories* and *routes* to certain conditions called *conditions*. For instance, it is possible to specify that a certain switch may be operated only if a certain dependent signal is red. Even more complex conditions, which depend on the combination of several objects, can be specified. For instance, it is possible to specify that a certain signal may be turned to green only if the switch behind the signal is closed <u>and</u> the track section behind the switch is not occupied.

Such conditions are specified by assigning a *condition* to the respective element. This is done by selecting the symbol of the element and using the **Properties** command of the **Edit** menu. In the following dialog select the tab labelled **Conditions**. Now select the state that will be affected by the condition – in the second example mentioned above, the state *green* of the signal **-** as well as the elements that will be checked to verify whether the condition applies or not. Also in the example, you would have to select the *switch* that will be closed and an appropriate *contact indicator* that indicates, whether the track section behind the switch is occupied.

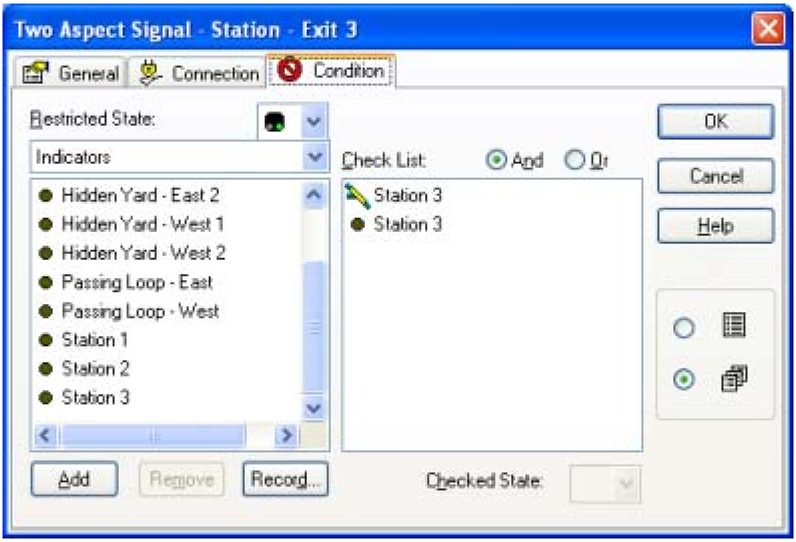


Diagram 120: Conditions of a signal

With the options **And** and **Or** it is possible to adjust the condition to your special needs. If **And** is selected, then <u>all</u> listed elements must have the required state to meet the condition. If **Or** is selected, then the condition already applies if at <u>least one</u> of the listed elements has the required state.

In the example displayed above, it is possible to turn the signal to green only if the switch "Station 3" is closed <u>and</u> the contact indicator "Station 3 " is turned off.

The elements that are part of the *condition*, and the element that is to be restricted, may be located at arbitrary locations of your model railroad. It should be noted that it is not necessary that the elements are placed in the same switchboard window.

11.3 Operations

X

It is possible to assign several *operations* to a *push button* or *on-off switch* instead of a digital address. Doing this, you are able to operate several elements with one single push button or on-off switch. It is for example possible to change the state of several signals, simultaneously, with one single on-off switch.

Each push button or on-off switch provides two sets of *operations* – one set for each state (on/off) of the push button or on-off switch. In this way, you can turn a group of related signals to green by turning on a certain on-off switch. The signals can be turned to red again by turning off this switch.

Operations are specified by selecting the symbol of the push button, or on-off switch, in the switchboard and using the **Properties** command of the **Edit** menu. In the following dialog select the tab labelled **Operations**. Now select the state that will trigger the operation – e.g. the state “on” of an on-off switch - as well as the elements that will be operated.

Diagram 121: Operations of a push button

In the example displayed above, the switch “Station 3” and the related signal “Station - Exit 3” are operated by pressing a push button.

An additional feature is *system operations*. Among others the following system operations are available:

- Playing of sound files
- Execution of an external program
- Output of a warning tone with the speaker of the computer
- Turning off the power of the digital system
- Emergency stop of all trains

With these system operations, you are able to create, for example, an emergency stop button in your switchboard.

Operations not only can be executed by push buttons and on-off switches, but also by other elements such as *indicators* or *routes*.

If *operations* are assigned to *contact indicators*, then passing trains are able to trigger other operations automatically. For example, a passing train can open or close certain crossing gates automatically. Another possibility is playing sound files triggered by passing trains. Since operations can also contain features of the *Dispatcher* – e.g. starting of a schedule (see chapter 5, “The Visual Dispatcher”) – virtually unlimited possibilities for automatic operation are provided.

A special application of *operations* is performed by *routes*. *Switches*, *signals* and other elements that are operated by routes can be locked until the corresponding route is released again.

Example: Automatic Reset of Signals

The following example explains how a signal can be reset to red after a train has left an occupancy section.

- Place or select a signal in the control panel.
- Create a *contact indicator* and link it to the occupancy section.
- Specify the signal in the “red” state as an *operation* of the indicator. This is performed when the indicator is toggled off.

	Operations	
Indicator	**On**	-
	Off	**Signal**

Table 5: Automatic Reset of Signals

When the train reaches the occupancy section, then the indicator is turned on. When the train leaves the section, the indicator is turned off. This again resets the signal, also. This is performed by the *operations* of the indicator.

Example: Emergency Stop Button

The following example explains how a push button symbol can be used to perform an emergency stop of the model railroad layout. It is also shown how the emergency stop can be triggered by pushing a certain key (here 'S') on the computer keyboard.

- Place or select a push button symbol in the control panel.
- Assign 'S' as a hot key to the push button (see section 2.7).
- Specify the system operation "Power Off" as an *operation* of the push button. This is performed when the push button is pressed.

	Hot Key	Operations	
Push Button	**'S'**	**On**	**Power Off (System Operation)**
		Off	-

Table 6: Emergency Stop Button

When the push button is pressed, either by clicking to it with the mouse or by pressing the 'S' on the computer keyboard, then the complete model railroad is stopped.

11.4 Semi-Automatic Control Mechanisms using Flagman Elements

The Flagman

X

With the possibilities described in the previous sections, it is already possible to create many and diverse semi-automatic control mechanisms. Even much more powerful functions are provided by the *flagman indicators* introduced in this section. This will be made clearer by the examples provided in this section. *Flagman indicators* work like intelligent relays that are turned on under certain conditions. *Flagmen* are able to indicate certain events and to execute *operations* automatically.

Flagman indicators are somewhat similar to *contact indicators*. While a contact indicator indicates if a certain feedback sensor is activated or not, a *flagman* indicates that a certain more complex event has occurred. A *flagman* is able, for example, to indicate that a train is waiting in front of a red signal. The event to monitor this is assigned to each *flagman* is a *trigger*. A *trigger* contains a set of elements whose states are to be monitored. In the example mentioned above, the *trigger* would contain the red signal and a *contact indicator* that monitors the track section in front of the signal. When the signal is red and a train touches the contact indicator, then the flagman is turned on by its *trigger*.

A *trigger* is specified by selecting the symbol of the flagman and using the **Properties** command of the **Edit** menu. In the following dialog select the tab labeled **Trigger**. Now select the elements that will be monitored.

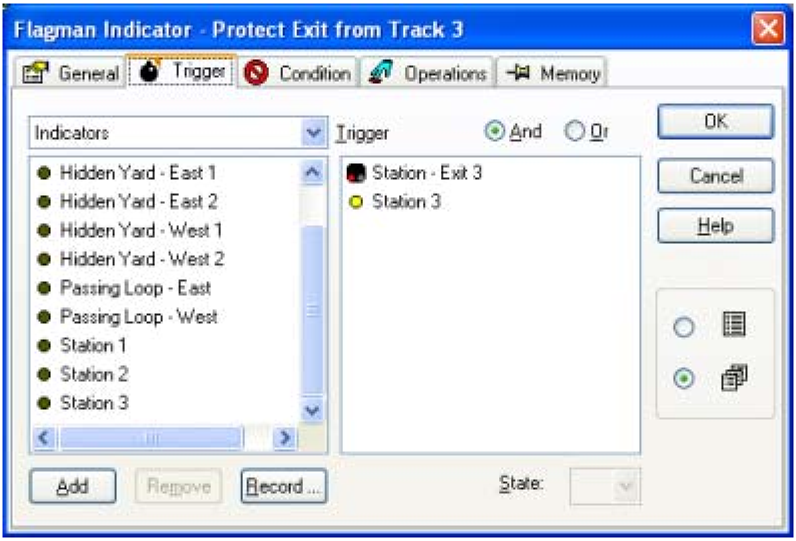


Diagram 122: Trigger of a Flagman

By selecting the option **And** or **Or**, additional possibilities are available. If the option **And** is selected, then all elements listed in the *trigger* must be in the required state to turn on the *flagman*. If the option **Or** is selected, then the *flagman* is already turned on, if at least one of the elements has the required state.

In the example displayed above, the *flagman* is turned on if the signal “Station – Exit 3” is red and the track section “Station 3” is occupied.

It is possible to include a *flagman* in the trigger of other *flagmen*. This function provides the ability to specify *trigger* conditions with virtually unlimited complexity.

Flagmen and Operations

It is possible to assign a set of *operations* to each state (on/off) of a *flagman* (see section 11.3, “Operations”). In this way it is possible to operate a set of elements automatically when a certain event occurs. This feature enables flexible semi-automation of your switchboards.

Flagmen and Conditions

It is also possible to assign a *condition* to a *flagman* (see section 11.2, “Protection and Locking with Conditions”). The *condition* is additionally checked each time, after the *trigger* of the flagman has been activated and before the *flagman* is turned on. If the condition does not apply the *flagman* remains turned off.

The following example demonstrates an application of this feature.

Example: Detecting Train Direction

The *condition* of a *flagman* can be used to detect the direction of a passing train.

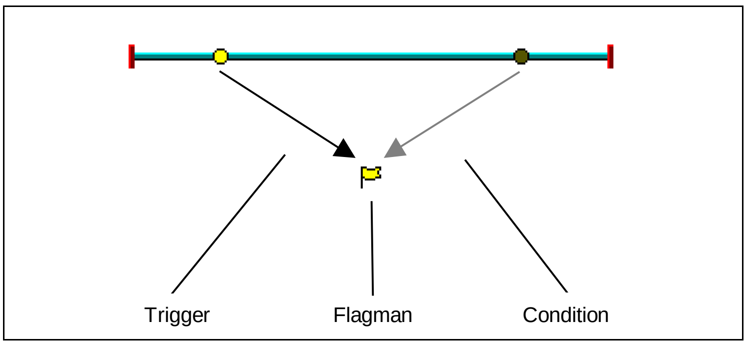


Diagram 123: Detecting Train Direction

On the track section displayed above, an operation should be performed by trains running from the left to the right. If a train passes from the right to the left, then nothing should happen. For this purpose, a detection mechanism that is activated by trains running from the left to the right only is needed.

In order to create this detection mechanism, two track sensors are placed on the track section. The distance between these sensors should be smaller than the length of the shortest train passing this section.

The following steps should be executed:

- Create a switchboard and draw the track diagram displayed above.
- Place two *contact indicators* (see section 4, “Contact Indicators”) in the track diagram and specify the digital addresses of the respective track sensors.
- Create a *flagman*.

- Specify the left contact in the “on” state as the *trigger* of the flagman.
- Specify the right contact in the “off” state as the *condition* of the flagman.

	Trigger	Condition
Flagman	Left Contact	Right Contact

Table 7: Detecting Train Direction

If the left sensor is passed by a train coming from the left, then this event is reported to the *flagman* by the *trigger*. The flagman then checks its *condition* and detects that the right contact is turned off. Since the *condition* applies, the *flagman* is turned on as required.

If the right sensor is passed by a train coming from the right, then nothing happens because the right contact is not part of the *trigger*. If the train passes the left contact a few moments later, then this event is again reported to the *flagman* by the *trigger*. The flagman again checks its *condition* and detects that the right contact is still turned on. Since the *condition* does not apply, the *flagman* is not turned on.

By assigning *operations* to the *flagman*, it is possible to operate other elements depending on the direction of passing trains.

Example: Detecting uncoupled Cars

The following example demonstrates how inadvertently uncoupled cars can be detected. This mechanism is useful at the entry to hidden yards.

For this mechanism, a track occupancy detector and two additional *flagman indicators* are needed. In the following, these flagmen are called “timer” and “alarm”.

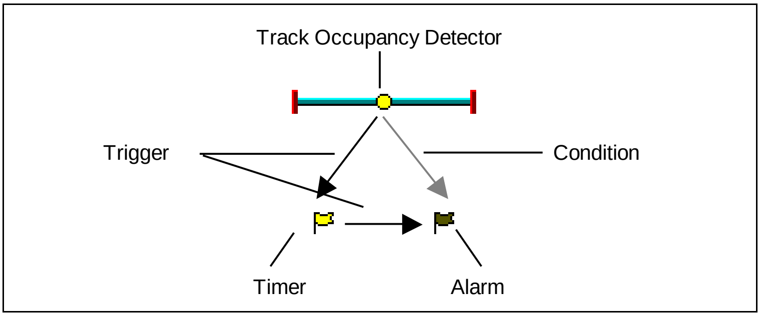


Diagram 124: Detecting uncoupled Cars

- Place or select a *contact indicator* in the track diagram and specify the digital address of the respective track occupancy detector.
- Create the two *flagmen* “Timer” and “Alarm”.
- Specify the occupancy detector in the “on” state as *trigger* of flagman “Timer”.
- Set the *memory* of flagman “Timer” to turning off **After 30 Seconds** and select the **Forced Reset** option.
- Specify “Timer” in the “off” state as *trigger* of flagman “Alarm”.
- Specify the track occupancy detector in the “on” state as *condition* of flagman “Alarm”.
- Specify the appropriate operations to be performed when flagman “Alarm” is turned on (e.g. emergency stop of all trains).

	Trigger	Conditions	Operationen	Memory
Timer	Occupancy Detector	-	-	Forced Reset: After 30 Seconds
Alarm	Timer	Occupancy Detector	appropriate emergency operations	-

Table 8: Detecting uncoupled Cars

When the track occupancy detector is passed by a train, the flagman “timer” is turned on by its *trigger*. It remains on for 30 seconds. After 30 seconds, the “timer” is turned off again, even if the track occupancy detector is still turned on – this is done because the **Forced Reset** option is selected. Turning off flagman “timer” is reported to the flagman “alarm” by the *trigger* of “alarm”. The flagman “alarm” now checks its *condition*, i.e. if the track occupancy detector is still turned on by some inadvertently uncoupled cars. If this is the case then “alarm” is turned on and performs the emergency operations.

The time period specified as memory of flagman “timer” must be large enough to enable the longest/slowest train to leave the occupancy detector. Otherwise a false alarm would be triggered. On the other hand the time period must be shorter than the interval between two successive trains. Otherwise it could happen that the next train has already turned on the occupancy sensor when the “timer” is turned off.

It is obvious that this mechanism only works if uncoupled cars can be detected by the track occupancy detector. If necessary, the axles at the end of each train can be made conductive using an appropriate resistor.

Example: Simple Track Occupancy Detection

The following example demonstrates how a track occupancy detection is made possible using temporary track sensors.

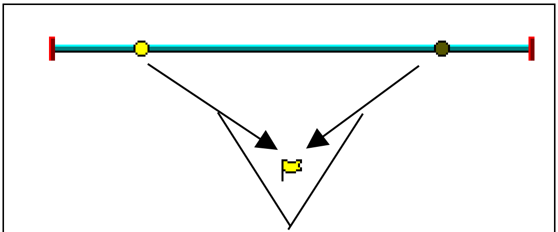


Diagram 125: Simple Track Occupancy Detection

In addition to the *contact indicators,* an additional *flagman indicator* is needed for indication of occupancy.

- Place or select the *contact indicators* into the track diagram and specify the digital addresses of the respective track sensors.
- Create a *flagman* for indication of occupancy.
- Specify both contact indicators as *trigger* of the flagman using the option **Or**.
- Set the *memory* of the flagman to **Toggle**.

	Trigger	Memory
Flagman	Left **OR** Right	**Toggle**

Table 9: Simple Track Occupancy Detection

When a train enters the track section between the track sensors the *flagman* is turned on by its *trigger*. When the train leaves the track section, the corresponding *contact indicator* is turned on. This event is again reported to the flagman through its *trigger*. The option **Toggle** now makes sure that the *flagman* is turned off.

This mechanism also works if the train enters and leaves the track section on the same side.

11.5 Macros

Macros are used to operate other elements.

They are very similar to *push buttons* in the *switchboard* (see section 2.5, “Signals and Accessories”). Like push buttons they are also able to perform operations (see section 11.3, “Operations”). Unlike push buttons they are not placed in a *switchboard*. Instead macros are for example used in the operations called by other elements (see section 11.3, “Operations”), executed in *schedules* (see section 5.10, “Schedules”), executed in *timetables* (see section 12.6, “Timetables”) or assigned to *engines* as part of their functions (see section 3.6, “Headlights, Steam and Whistle”).

In this way macros are invisible and work in the background of the program.

Engine functions assigned to macros can only be executed, if the macro is executed in the context of an engine. This is the case, if the macro is assigned to another engine function (in this way engine functions can be triggered indirectly by other engine functions), or if the macro is assigned to a schedule. If the macro is not executed in the context of an engine (e.g. by a timetable), then all engine functions contained in the macro are ignored.

Example: Automatic Engine Whistle

Engine or Trains running a schedule shall blow their whistle for exactly two seconds when passing a certain section.

This is done in the following way:

- Open the macro list and create a new *macro* “Whistle”.
- Specify “Whistle on”, “Delay 2000 milliseconds” and “Whistle off” as *operations* of the macro according to Diagram 126.
- Assign macro “Whistle” as engine function or as an operation to be performed by a schedule.

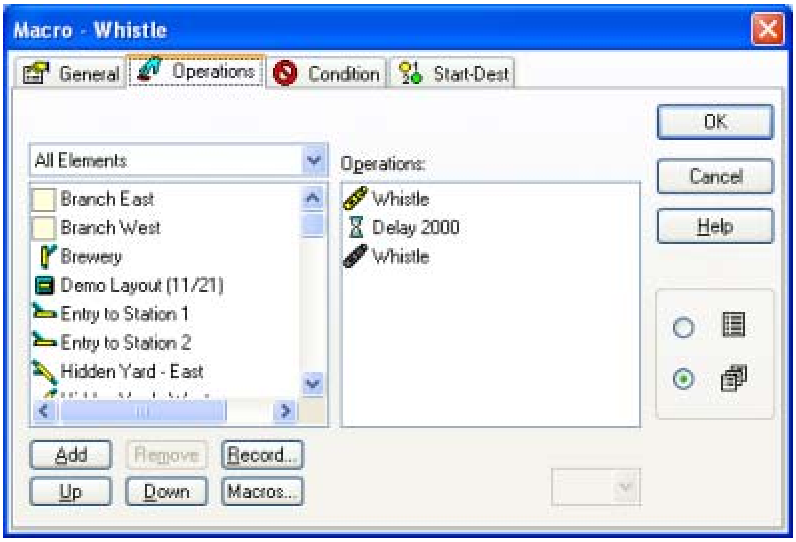


Diagram 126: Setting up macro Whistle

For a detailed discussion about engine functions refer to section 3.6, “Headlights, Steam and Whistle”, please.

11.6 External Control Panels

Running an external control panel simultaneously with your computer is made possible by **TrainController™**. One solution is to not connect the buttons of your external control panel directly to your model railroad, but indirectly through the feedback decoders of your digital system. If a button on your external control panel is pressed, then this event is reported to the computer as a feedback signal by the digital system. With **TrainController™**, you are able to create appropriate *contact indicators* to monitor these feedback signals. By assigning appropriate *operations* to these contact indicators, the accessories on your model railroad can be operated as desired.

Please note, that a push button or switch on your external control panel is associated with a contact indicator in TrainController™.

To operate a switch with two buttons in an external control panel perform the following steps:

- Create a switch symbol in your switchboard.
- Connect the real push buttons in your external control panel with two input contacts of a feedback decoder of your digital system.
- Create two contact indicators and assign the addresses of the two input contacts to them.
- Assign the first state of the switch symbol to the *operations* of the first contact indicator and the second state of the switch symbol to the *operations* of the second contact indicator.

A very useful application is the operation of *routes*. Without using a computer, the installation of expensive equipment would be necessary to operate *routes* with an external control panel. The option to assign contact indicators as start and destination keys to routes (see section 2.6, “Routes”) is very useful as well.

To operate a route with start and destination keys in an external control panel perform the following steps:

- Create a route symbol in your switchboard.
- Connect the real push buttons in your external control panel with two input contacts of a feedback decoder of your digital system.
- Create two contact indicators and assign the addresses of the two input contacts to them.
- Assign the contact indicators as start and destination keys to the route symbol.

12 The Visual Dispatcher II

12.1 The manually created Main Block Diagram

In section 5.2, “Blocks”, you have been made familiar with the *main block diagram*. This diagram contains *blocks* as well as the routes and track connections (*links*) between blocks. and describes the track layout of your entire model railroad in rough outline.

The block diagram can be drawn by yourself or automatically created by the software. In Part I of this Users Guide it was always assumed, that the block diagram is automatically calculated by the software. This is very simple, convenient and useful for small and medium size layouts, where the track plan of the entire layout fits into one single switchboard, which is used as the base to calculate the block diagram.

Advanced users, however, may desire not to be limited by the conditions, that apply to the automatic calculation of the block diagram. These are:

- The complete track diagram of the layout with all turnouts and crossings must be drawn without any gaps in one single switchboard.
- Traffic boxes must be created in this switchboard for all blocks of the layout.
- The blocks must be connected by track symbols without any gaps.

In cases were it is desired to split the track diagram into several switchboards in order to control several yards with individual panels or in cases, were users do not like traffic boxes to be displayed in the switchboard it is useful to create or edit the block diagram manually.

TrainController™ offers this possibility.

But even if your layout is large and you intend from the beginning, to create your block diagram yourself, it is useful to start from scratch with a calculated block diagram for an important part of your layout, such as the main station. Draw this station in the switchboard displayed in the main window of **TrainController**™ and let the software automatically calculate the block diagram for this part of the layout. Whenever you want you can turn off the automatic calculation of the block diagram and edit and extend the existing block diagram yourself.

! It is also no problem, if you started to draw the “wrong” part of your layout in the main window of the switchboard and want to exchange this track diagram with another track diagram, that has been drawn in an additional switchboard window. It is possible at any time, to exchange the content of the main window and the content of an additional switchboard window.

! **It is also possible, to turn on automatic calculation of the block diagram again, after a certain time of working with a manually created block diagram. This function is to be used with extreme care, though! Existing routes and blocks may be deleted and replaced by calculated blocks and routes. Existing schedules may be affected, too, and may become invalid, too!**

You can safely turn on calculation of the block diagram and check, if you like the result. If you do not like it, you should not safe the result to your original data file, because this will overwrite your original data stored on the disk of your computer.

Editing the Block Diagram

For manual editing of the block diagram the automatic calculation of the block diagram must be turned off. **TrainController™** provides powerful and intuitive tools for editing of the block diagram and for insertion of blocks, routes and links to this diagram.

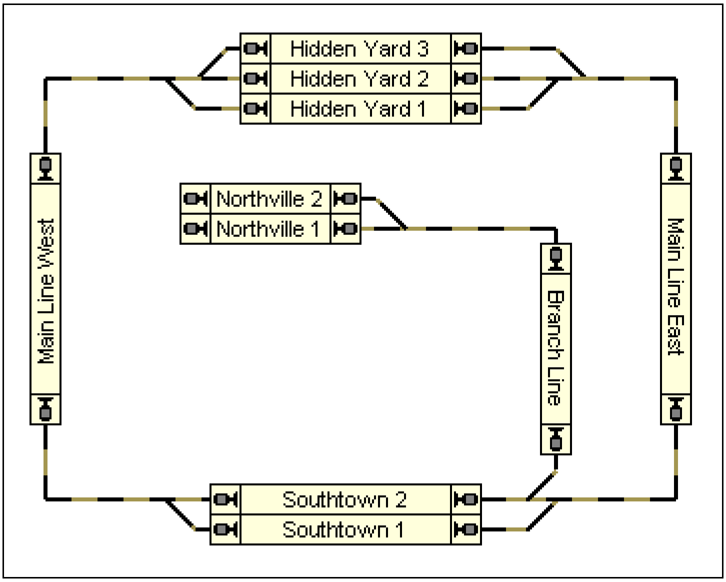


Diagram 127: Main Block Diagram in the Visual Dispatcher

Blocks are displayed on the computer screen by rectangular boxes. The blocks are connected by routes or links. These routes or links are drawn as lines. Since there are turnouts to be operated in the example displayed above in order to travel from one block to the next each two blocks must be connected by a route instead of a plain link. These routes must be recorded as outlined in section 2.6, “Routes”.

Please note that the block diagram represents the track layout in rough outline. The actual track connection between “Main Line West” and “Hidden Yard 3”, for example, contains two switches. These switches are not drawn in the block diagram in detail or as separate objects. Instead a link between both blocks is created, that indicates, that there is a track connection between both blocks.

In order to create the block diagram perform the following steps:

- Create all blocks of the layout, place them according to their location on the actual layout and turn them vertically, if desired.
- Create the routes and links between the blocks. Ensure that the routes and links attach the blocks at the correct exits(see below).

Routes and Links

Routes and links are used to connect blocks with each other. If there is a track connection on your layout, on which trains can travel from one block to another, then a route or link must be drawn between both blocks. Routes and links are displayed by lines and look quite similar. The only difference is the following: if there are turnouts or crossings contained in the track path between both blocks, then a route is to be inserted. A route represents a track connection, that contains turnouts. If there are no turnouts or crossings in the track connection, then it is sufficient to draw a plain link between both blocks.

Links can be conveniently turned into routes and vice versa at any time.

The following image explains the terms once more:

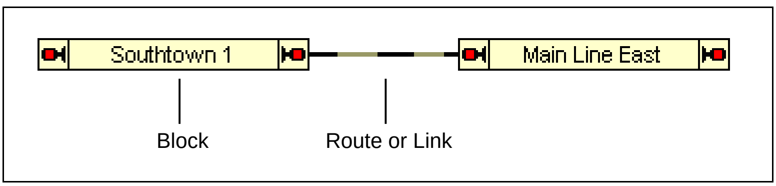


Diagram 128: Blocks, Route and Link

In the diagram displayed above the blocks “Southtown 1” and “Main Line East” are connected with a link or a route.

If you want to use a *manual route*, i.e. a route symbol, that has been created previously in the switchboard, for automatic control with the *Visual Dispatcher*, then you can move the original route symbol from the switchboard to an appropriate link in the main block diagram. This converts the link into an automatic route (see also page 60). Additionally you can have the software create an on-off switch at the previous switchboard position of the route symbol, after a route symbol has been dropped to the main block diagram. This on-off switch contains appropriate *operations* for manual operation of the route in the switchboard, if desired (see also 11.3, “Operations”).

! **The main block diagram of the complete layout is drawn within one single diagram. The space provided by the software is not limited. In case of large layouts it**

is possible to zoom and scroll the window accordingly, in which the block diagram is displayed.

Please pay attention that the blocks, routes and links are arranged accordingly. A certain block can only be passed without stop when the train can enter the block through one entry and leave the block through the opposite exit.

Diagram 129: Passing Blocks and Links without Stop

In the diagram displayed above a train can pass block “Southtown 1” without stopping and reversing its direction. Coming from block “Main Line West”, for example, a train will enter “Southtown 1” through the left entry and can leave this block through the opposite exit to proceed to block “Main Line East”.

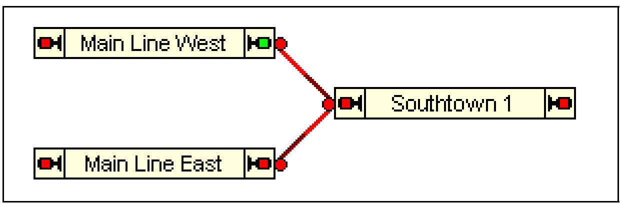


Diagram 130: Blocks with enforced change of Direction

This arrangement allows also for train movements from “Main Line West” via “Southtown 1” to “Main Line East”. But after arriving in “Southtown 1” a train has to stop and to reverse its direction in order to proceed to “Main Line East”, because it enters and leaves the block on the same side.

Even though you are not forced to do so it is recommended to draw the block diagram according to the actual conditions of your layout. If you follow the layout of your model railroad you will probably not encounter any problems.

! **Nevertheless you should always check whether the routes and links between your blocks touch the blocks at the correct side.**

! **Please note also that a train cannot run from one block to another under control of the Dispatcher if there is no route or link between both blocks.**

It is not necessary to draw routes or links always as straight lines. If desired for display purposes these lines can also contain corners. Such corners are only used for clearer visualization of the diagram, they do not have any impact to the operation of trains.

Guidelines for routes and links:

- Routes and links must touch the blocks at the correct entries/exits, because this affects the direction of travel through the related blocks assumed by the *Dispatcher*.
- Routes are to be used, if the track connection between both blocks contains turnouts or crossing.
- Each pair of blocks can be connected by at most one link, but by an arbitrary number of routes.

12.2 Nodes

Imagine the following block diagram, that shows two yards with 4 blocks each. Each block in the left yard is directly connected with each block in the right yard by a route. There are no blocks between both yards.

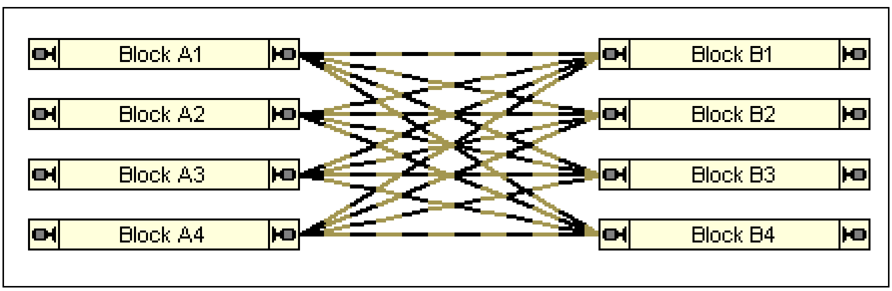


Diagram 131: Multiple Routes without Nodes

The structure of this small diagram does not look very clear at all. The routes can be formed and overlaid, though, to show a more clear structure as shown in the following image.

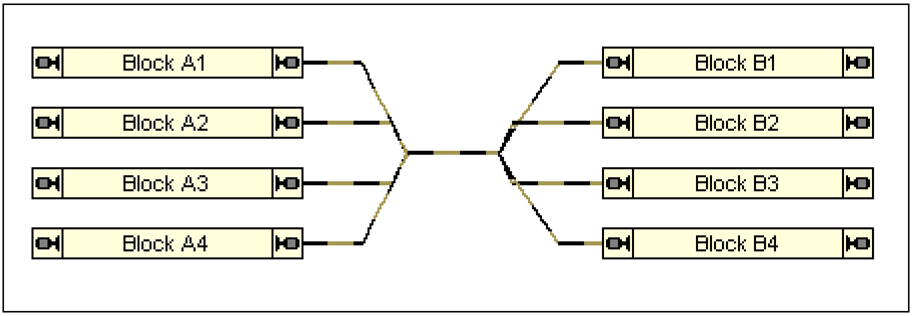


Diagram 132: Multiple Routes without Nodes

Diagram 132 shows a much more clear structure. It has still some drawbacks, though. In total 16 routes are to be created and at each point of the lines there are at least 4 routes, that overlay each other, which makes it difficult, to select the right route, if necessary.

Graphical problems like these are solved with the help of *nodes*. These are additional elements in the block diagram, that can help to reduce the complexity of a block diagram and that can be used to create a clear structure. Nodes can be used in all cases, where multiple blocks are to be connected with each other.

Nodes are used like blocks with regard to drawing in the block diagram. They are represented by small rectangles and look like “small blocks”. Like blocks they can be linked to multiple elements on both sides. This is shown in the following diagram:

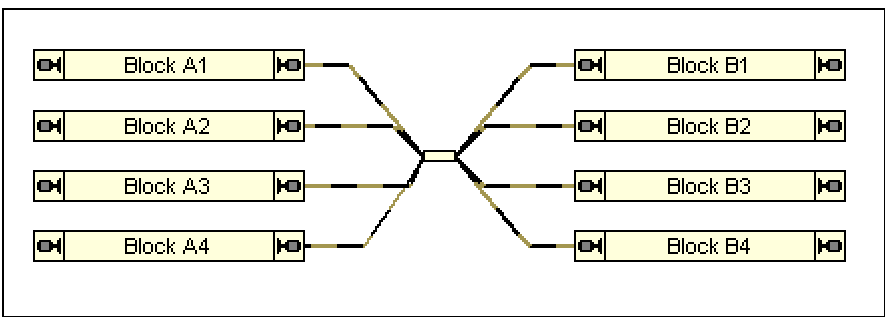


Diagram 133: Multiple Routes with Nodes

The node shown in Diagram 133 connects to 4 routes on both sides. The resulting structure is very clear now and it can be clearly seen, which routes connect which two blocks. The node additionally helped here to reduce the number of necessary routes from 16 to 8. This was possible by splitting each route into two parts.

There is an important difference between blocks and nodes with regard to operation, though: nodes are ignored during operation, i.e. they are just used to reduce the number of necessary routes or links in the block diagram, but they have no counterpart on the real layout and cannot be reserved by trains. Concurrent trains may “share” or “pass” the same node simultaneously. Interlocking of concurrent trains must be done on the base of the reservation of blocks and routes. In the example above this means that concurrent routes on the same side of the node must not be activated at the same time. This can be achieved for instance by including common track elements into concurrent routes.

12.3 Virtual Contacts and Virtual Occupancy Indication

General

Virtual Contacts are similar to normal *contact indicators* (see section 4, “Contact Indicators”). But unlike contact indicators there is no related track contact or real sensor on the model railroad. Instead virtual contacts are assumed to be located in a certain distance from another indicator, which is called reference indicator.

Virtual contacts can be used to reduce the number of necessary track sensors on your model railroad. Typical applications are stopping trains or triggering operations by passing trains in a certain distance from an existing sensor (see also section 11.3, “Operations”). Another application is stopping of trains depending on the length of the trains - e.g. in order to stop trains in the middle of a platform.

After creation of a virtual contact the following properties are specified:

- a maximum of two reference indicators, one for each direction of travel (see section 5.3, “Direction of Travel vs. Engine Orientation”). **These indicators must be already assigned to a block.**
- the distance from the specified reference indicator
- whether the virtual contact shall be turned on, when the head, the middle or the tail of a train passes the point, where the virtual contact is assumed to be located

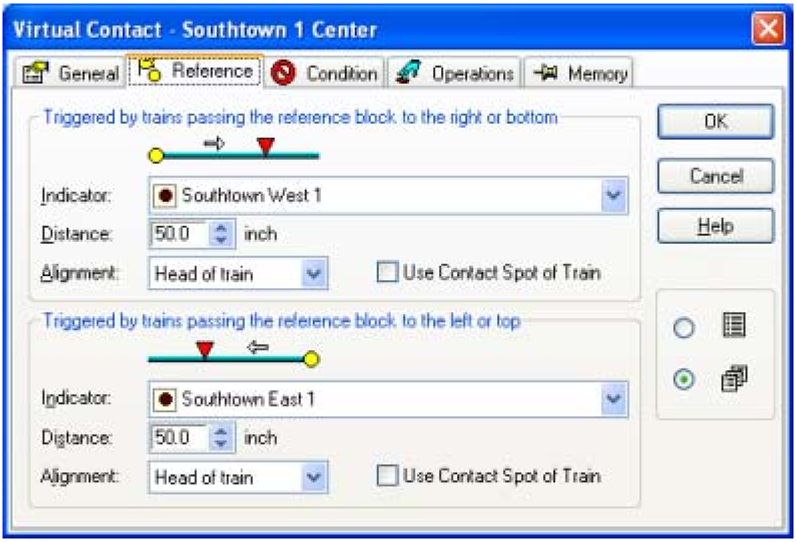


Diagram 134: Arranging a Virtual Contact

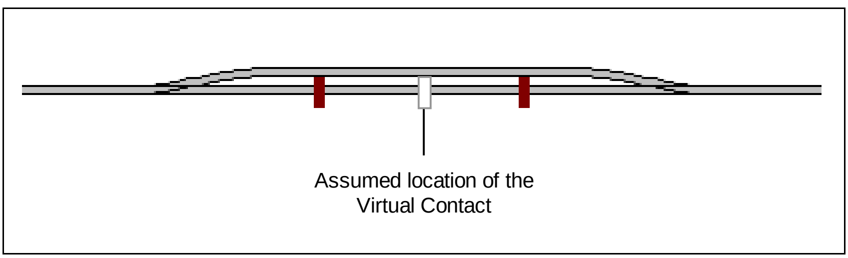


Diagram 135: Virtual Contact with two reference indicators

The image above shows a virtual contact (white rectangle) with two reference indicators (dark red markers). When a train passes the left indicator from the left to the right, then the current scale speed of the train and the distance of the virtual contact to this indicator is taken into account to calculate the time, at which the train passes the assumed location of the virtual contact. Even if the train changes its speed after passing the left indicator, this is taken into account, too, and the resulting time is adjusted accordingly. When a train passes the left indicator from the right to the left, the virtual contact is not turned on.

Virtual contacts work only under the following conditions:

- if a train is stopped or changes its direction after passing the reference indicator and before arriving at the assumed location of the virtual contact, then the virtual contact is not turned on, even if the train continues traveling in the original direction and passed this location.
- it is very important, that the scale speed of the passing engines and trains can be calculated correctly. For this reason it is recommended to adjust the speed profile of each affected engine accordingly (see section 3.5, "The Speed Profile").
- it is very important, that the direction of travel of each passing engine or train is known. Otherwise virtual contacts could be turned on by trains traveling in the wrong direction. For this reason it is also essential to determine, which engine or train passes the reference indicators assigned to the virtual contact. This is only possible, if the affected trains are running under control of the Dispatcher, and if the reference indicators are assigned to blocks.

! **Virtual contacts can only be turned on by engine and trains running under control of the *Visual Dispatcher*. The reference indicator must be assigned to a block.**

In conjunction with Virtual Contacts the difference between momentary contacts and occupancy sensors has to be taken into account. If an indicator representing a momentary contact is used as a reference indicator of a Virtual Contact then the one and only sensing point represented by the momentary contact is used as the base of the distance from the reference indicator to the Virtual Contact.

If an indicator representing an occupancy sensor is used as a reference indicator of a Virtual Contact then the sensing point reached first by trains running into the particular direction is used as the base of the distance from the reference indicator to the Virtual Contact. In Diagram 72, for instance, the left boundary of the sensed track section is used as the base of the distance for trains running from the left to the right.

Using Virtual Contacts as Indicators in a Block

Virtual contacts can also be used to reduce the necessary number of track sensors used as indicators in the Dispatcher, for example as brake and stop indicators. Nevertheless if it is desired, that trains stop very exactly - e.g. in front of a red signal, then it is recommended to use a real sensor and a contact indicator as stop indicator to stop the train.

But it is actually possible to use a virtual contact to report block entry or as a brake indicator. The reference contact of the virtual contact could be the stop indicator of the previous block, in which case the entry into the related block is indicated a certain time after a train passed the stop indicator of the previous block. Since the brake indicator does not need a very exact location, it is possible to use a virtual contact as brake indicator, too.

Some of the configuration explained in section 5.8, “Arranging Indicators in a Block” can be created with Virtual Contacts, too, by replacing momentary sensors by Virtual Contacts:

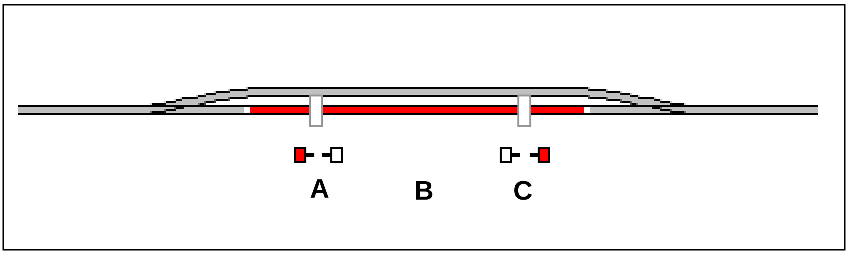


Diagram 136: Block with occupancy sensor and Virtual Contacts

Diagram 136 is a variant of Diagram 91. The momentary sensors A and C are replaced by Virtual Contacts. B is the reference contact of A and C for both directions. For each Virtual Contact A and C and each direction different distances to the reference contact B must be specified. The indicators A, B and C are assigned to the block and act as stop and brake indicators similar to Diagram 91.

This method is less expensive than the method displayed in Diagram 91, because Virtual Contacts are free of charge. But this method requires that the speed profile of the affected locomotives is adjusted thoroughly and that the locomotives run very exactly according to their profile (see also section 3.5, “The Speed Profile”).

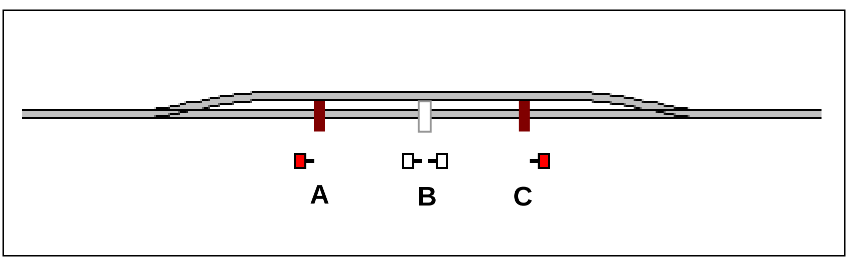


Diagram 137: Block with momentary sensors and Virtual Contact

It is also possible to add a Virtual Contact to Diagram 92 in order to avoid undesired speed changes of passing trains caused by activation of routes ahead. This is shown in

Diagram 137. This configuration works in the same way as Diagram 93. The Virtual Contact B acts as brake indicator for both directions. A is the reference contact of B for trains running to the right and C, respectively, the reference contact for the opposite direction. Even though a Virtual Contact is used the locomotives do not have to match their speed profile as exactly as in Diagram 136, because B acts as a brake indicator only and it does not matter, if trains begin to slow down a little bit too early or too late.

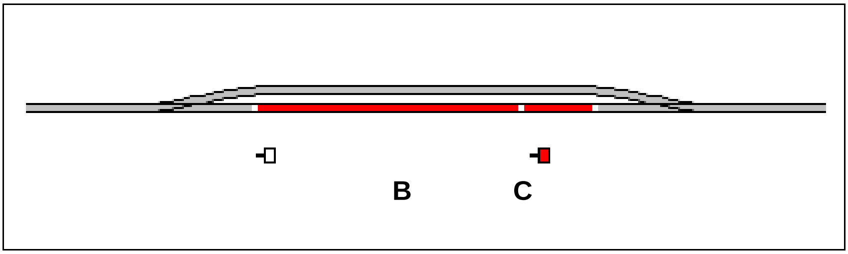


Diagram 138: Block with two occupancy sensors

In the diagram displayed above it is assumed that the block is only passed from the left to the right. B acts as brake indicator and C as stop indicator for trains running to the right. But there is a disadvantage for passing trains. Let us assume that a train is passing the block from the left to the right and that a route is to be activated before the block ahead, to the right of this block. As soon as the passing train enters section B the route is activated. In the same moment the train begins to slow down, because B is also a brake indicator and the train has to wait, until the route is reported to be activated which needs a certain time. This can be avoided by adding an additional Virtual Contact according to the following diagram:

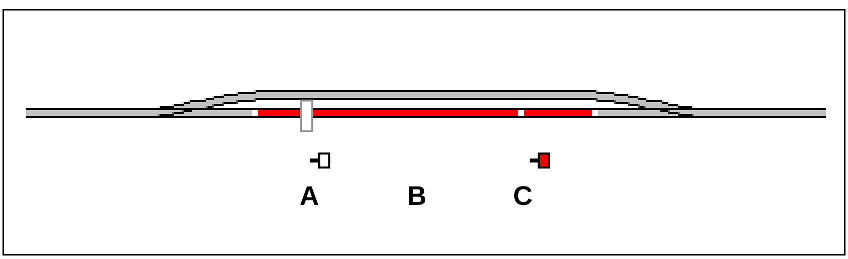


Diagram 139: Block with occupancy sensors and Virtual Contact

In Diagram 139 it is also assumed that the block is only passed from the left to the right. The occupancy indicator B is the reference contact of the Virtual Contact A for trains running to this direction. A acts as brake indicator and C as stop indicator for this direction. By adding the Virtual Contact A undesired speed changes of passing trains caused by activation of routes ahead can be avoided provided the virtual distance between B and A is large enough. Even though a Virtual Contact is used the locomotives do not have to match their speed profile as exactly as in Diagram 136, because A acts as a brake indicator only and it does not matter, if trains begin to slow down a little bit too early or too late.

Stopping in the Middle of a Platform

Virtual Contacts can also be used to stop trains automatically in the middle of a platform.

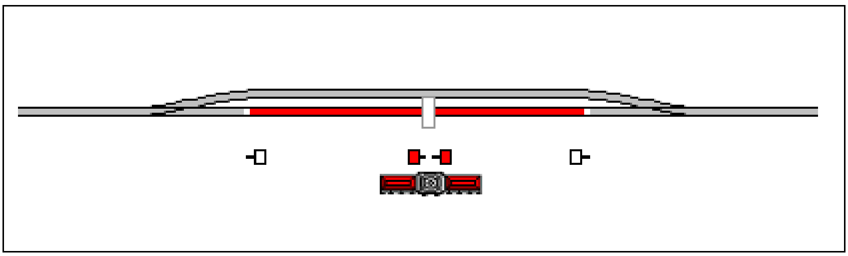


Diagram 140: Stopping in the middle of a platform

If the distances to the two reference points in the example displayed above is determined accordingly, then the virtual contact is assumed to be located exactly in the middle of the platform. In order to stop trains in the middle of the platform the option **Middle of train** must be selected for both directions of travel according to Diagram 134.

A special situation is the case, where a real momentary sensor is already located in the middle of a platform. In this case locate the virtual contact exactly on the position of the contact indicator assigned to the sensor. This is done by specifying the contact indicator as reference indicator for both directions of travel and entering 0 as distance.

Please note that using the option Middle of train requires, that the correct length of the affected engines or trains is specified accordingly.

Virtual Occupancy Indication

If an indicator is associated with a momentary contact then this contact can be upgraded with the *memory* of the indicator to a *virtual occupancy* sensor (see section 11.1, “The Memory of Indicators”). If this is done the indicator stays on after activation of the contact until the complete train has passed the point where the contact is located. It is possible to take into account the point where the contact is activated or the point where the contact is deactivated. In this way it is for example possible to avoid premature release of routes in cases when long trains pass a route and only momentary contacts are used. This option works only for trains under control of the Dispatcher and it relies on correct specification of the length of each train.

Virtual Contacts can be combined with Virtual Occupancy Indication, too. The Memory is namely also available for Virtual Contacts. In this way a Virtual Contact will be turned on when a train reaches a certain point on the layout. And the Virtual Contact will stay turned on, until the last car of the train has passed this point.

Please note the difference between Virtual Contacts and Virtual Occupancy Indication. A Virtual Contact marks a certain point on your model railroad, i.e. a Virtual Contact is turned <u>on</u>, when a train is assumed to arrive at a certain point. Virtual Occupancy Indication is used to turn a certain real or Virtual Contact <u>off</u> when a train has passed a certain point completely.

12.4 Controlling the traffic flow in Schedules

Limiting the Reservation of Blocks and Routes in certain Schedules

For each *block* and each route in a schedule it is possible to specify a *condition.* This is a condition, which must be valid when a block or route is about to be reserved during a running schedule. As long as the condition does not apply it is not possible to reserve the block or route. How conditions work is outlined in section 11.2, “Protection and Locking with Conditions”.

This feature provides additional control. It is for example possible to specify, that a certain block may only be reserved, if a certain *on/off switch* is toggled off. Turning off or on this switch you can intervene into the traffic flow at any time and lock or release the affected block.

Such conditions can be specified on a global or a per-schedule base. Global conditions are specified as part of the properties of blocks or routes as outlined in section 11.2, “Protection and Locking with Conditions”. They are valid for <u>all</u> schedules, that use these blocks or routes.

Conditions can also be specified on a per-schedule base, while editing the diagram of a schedule. Conditions specified in this way apply only when <u>this</u> schedule is being executed. Such local per-schedule conditions are always only valid for the schedule they have been specified for, other schedules are never affected by these local conditions.

Preferring Blocks

Each block can be temporarily marked as *preferred block* during operation. If a train is running on a schedule and there are several equivalent alternatives to continue the travel, then the train will select an alternative that contains a *preferred block*, if any.

By marking blocks as *preferred* you can influence at any time the selection of paths for running trains (see also page 137).

Please note that preferring a block affects all trains that have a chance to go to this block.

A further option to control the reservation of blocks and routes are *critical sections*. These are explained in the following section.

Critical Sections

In the diagram displayed below "Main Line East" and "Main Line West" are specified as critical sections. Critical sections are displayed on the computer screen with a blue marking.

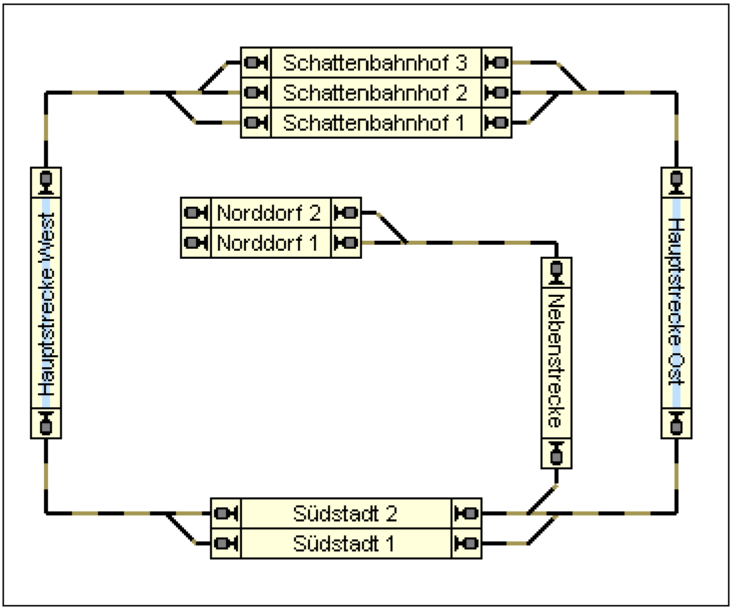


Diagram 141: Critical Sections

The most usual application of critical sections is to protect opposing trains from deadlocking each other. If the *Dispatcher* encounters during reservation of the next blocks ahead, that a certain block is marked as critical section, it will continue reserving further blocks, until a block is reached, that is not marked as a critical section.

If in the diagram displayed above block “Main Line East” is reserved for a train that is about to leave a block in “Hidden Yard”, then the *Dispatcher* continues with reservation of a block in “Southtown”, too. If it is not possible in this moment, to reserve a block in “Southtown”, because both blocks in “Southtown” are already reserved by other trains, then the *Dispatcher* will not even reserve “Main Line East” and the train does not get the permission to leave “Hidden Yard”.

A train may enter a critical section only if it is sure that it can leave the critical section on the other side.

If a critical section contains more than one block, then either all blocks of this section plus the first block behind this section are reserved in one step or no block is reserved at all and the train must not proceed.

A typical example of a critical section is a single track line between two stations, which can be traveled in both directions. If there is one or more block between these two stations, then these blocks should be marked as critical sections. A train, which is about to leave one of the two stations towards the other station will not leave this station, if it is not sure, that a block inside the other station is available, i.e. if it is not sure, that it can leave the critical section on the other side. This prevents trains from deadlock each other on the single line track between both stations.

There is a specific option, though, that allows trains, that are executing the same schedule at the same time, to share a critical section. In this way it is possible to queue several trains heading to the same direction in the same critical section while opposing trains must wait, until the complete critical section is clear. This allows several trains to follow each other on a single track line while locking opposing trains.

Critical sections can be assigned to blocks on a per-schedule base or alternatively in the main block diagram, too. A block that is marked as a critical section in the main block diagram will be treated as such in all schedules, that contain this block. A block, that is marked as critical section in certain schedules only will only affect those trains, that are controlled by this schedule.

In Versions prior to Version 5.0 of **TrainController™** the *Dispatcher* used a method based on so called *sub-lines* and *stages* to avoid such deadlocks. These methods are superseded by the more flexible concept of critical sections. Data files imported from an earlier version of the software are converted accordingly, i.e. stages and sub-lines contained in such data files are converted to critical sections.

Long trains to Long Tracks – Short Trains to Short Tracks

X

Usage of a schedule can be restricted to certain engines, trains or train groups. In this way it is possible, that certain schedules are only started with passenger trains or to avoid, that electric engines enter tracks without overhead cable. This feature can also be used to make sure, that trains automatically entering a hidden yard are directed to tracks, which are long enough to hold the train.

A schedule can only be started when a train can be found on a block of this schedule and this train has the permission to use this schedule.

If no engine or train is explicitly specified as permitted train, then the schedule can be used by all trains.

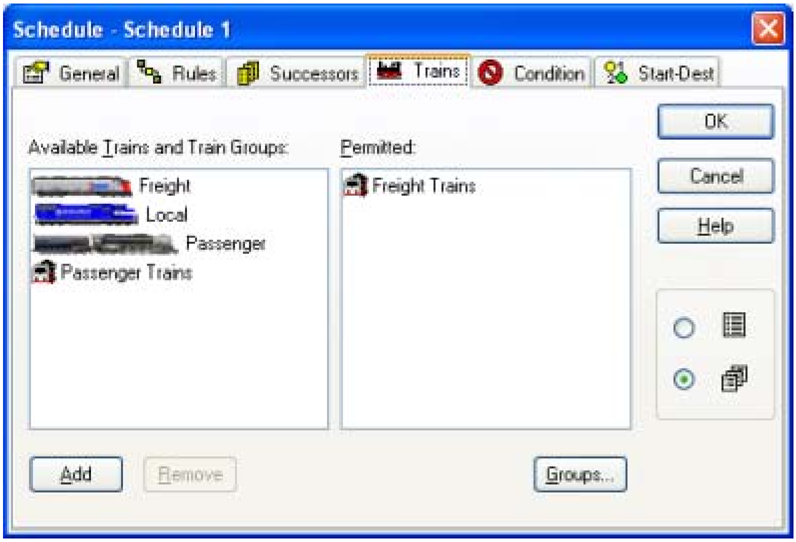


Diagram 142: Specifying the Trains permitted to use a Schedule

Train Groups are useful in conjunction with schedules in order to put together related engines or trains in groups. For example you can create the group of all *passenger trains*, or all *freight trains* or all *electric engines*. If you want to create several schedules only for freight trains, then you do not have to specify the particular trains as permitted trains for each affected schedule. It is much more convenient to create a train group for your freight trains and assign only this group to the affected schedules.

Train groups can contain other train groups, too. In this way the train group of all passenger trains can contain the group of all local trains and the group of all express trains.

For each *schedule* it is additionally possible to specify a *condition.* This is a condition, which must be valid when the schedule is started. As long as the condition does not apply it is not possible to start the schedule. How conditions work is outlined in section 11.2, “Protection and Locking with Conditions”.

This feature provides additional control. It is for example possible to specify, that a certain schedule may only be used, if a certain *on/off switch* is toggled off. Turning off or on this switch you can intervene into the traffic flow at any time and lock or release the affected schedule. It is not possible to start a locked schedule.

Routes with separate occupancy indication

Release of routes can be controlled individually and independently from the occupancy state of adjacent blocks. It is possible to assign a set of indicators to each route that are used to determine whether a route is occupied or not. If at least one of these indicators is turned on then the route is assumed to be occupied. It is possible to assign the same indicator to several routes. The software provides different options that control, which indicators are assigned to the route. There is an option called **Auto-Detect**. When this option is set, then all indicators located on the recorded track path of the route are automatically used to determine the occupancy state of the route. It is also possible to turn this option off and/or to assign an individual set of indicators that does not depend on the track path of the route. With these indicators it is possible to control independently from the occupancy state of adjacent blocks, when a route is released.

As outlined on page 138 blocks or routes are not released, until the train reaches a stop indicator in a subsequent block. If your routes are equipped with an own occupancy indication, it is possible to turn of this rule. In this case routes can be already released, when the train reaches the first indicator at the entrance of a subsequent block, assumed, that the route is not reported as occupied anymore. In this case the track area covered by such routes is earlier available for other trains.

! The rule to release preceding blocks and routes of a certain schedule at the stop indicator of subsequent blocks should be turned off only, if the routes contained in this schedule are equipped with an own occupancy indication.

12.5 Examples

Example: Manual Control of Station Entry

Trains shall run automatically on the small layout displayed below. Before entering the station trains shall wait until the human operator selects a destination track with start and destination key.

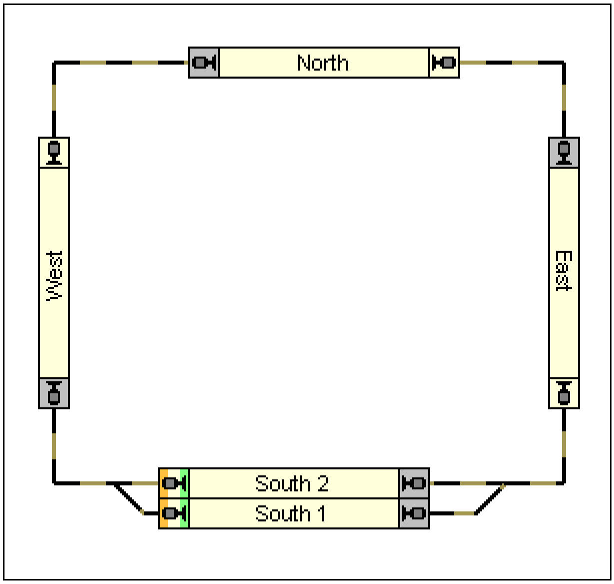


Diagram 143: Manual Control of Station Entry

This situation can be controlled with one single schedule, that is displayed in Diagram 143, too. The start and destination blocks of this schedule are located in block “South 1” and “South 2”. Since each schedule can be started in two directions, this schedule is able to control trains that travel clockwise, as well as trains that travel counter-clockwise.

When the schedule is being started in either direction, then we take care, that the entry to both, “South 1” and “South 2” is locked. The started train will proceed to “East” or “West”, respectively, and stop, if the lock has not been released until then.

By activating a route you can pre-select a path to “South 1” or “South 2”, respectively. This route can be associated with a pair of start and destination keys. If additionally the release of both blocks in “South” is performed as *operation* by each route, too, you are not only able to pre-select a path with the start and destination key, but also to release the lock which causes the train to continue its journey.

There are several possible variants. Instead of locking the entries to “South 1” and “South 2” you can also lock the bottom exits of “East” and “West”, respectively. You can also terminate the schedule in “East” or “West” and start another schedule with start and destination key, that directs the train from “East“/“West” to “South 1” or “South 2”.

Example: Manual Control of Station Exit

The exit of the hidden yard shall be controlled manually in the following way: it shall be possible to select the train, which is to be started by a schedule, by selecting the track, from which the train shall start.

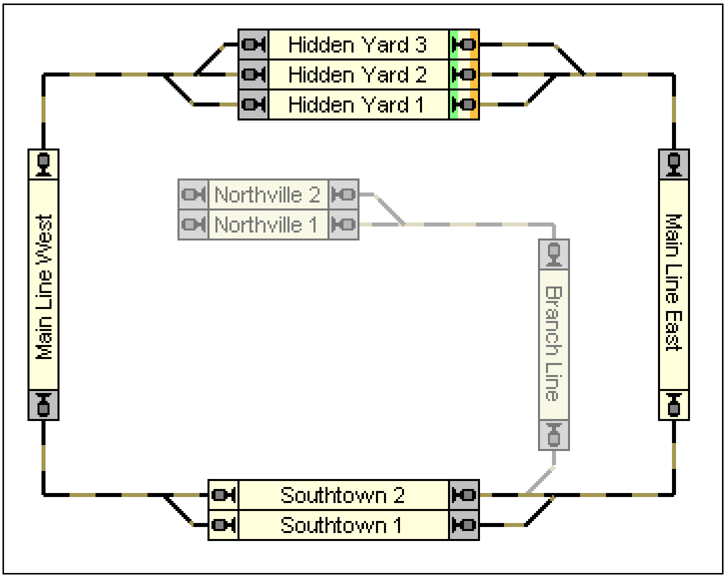


Diagram 144: Manual Control of Station Exit

The most simple solution is to assign an *operation*, that starts the schedule, to the operations of each route, that connect the blocks in “Hidden Yard” with the blocks “Main Line East” or “Main Line West”, respectively. Details about operations can be found in section 11.3, “Operations”. Instead of starting the schedule directly we activate the route. Through the operations of the route the schedule is started after activation of the route. This configuration additionally uses a trick, that has been explained on page 137. When selecting one or several alternatives to start or continue a schedule **TrainController™** prefers to select an alternative that prefers an already activated route. Since we activate the desired route first before the schedule is started, the activated route will be selected as the route to be used by the schedule and the train waiting in the block next to this route will be started.

An alternative solution uses macros (see section 11.5, “Macros”) and the possibility to lock the exit of each block (see page 106). For each track and each possible path a separate macro is defined. Additionally, the exits of each block in “Hidden Yard” are held locked by default. This can be done by manually locking all exits manually at the beginning. Appropriate operations are assigned to the macro, that first remove the lock from the exit of the related block and then start the schedule. For example, the macro, that controls the exit from “Hidden Yard 2” to the right, executes removing the exit lock of “Hidden Yard 2” to the right plus start of the schedule. In the same way the other macros are configured, one macro for each track and each side of the “Hidden Yard”. With an appropriate operation, that restores the lock to the exits and executed by the schedule, when the block is released (see page 139), we can ensure that the exit lock of the start block is restored to the default.

In the first case a route is started instead of the schedule and in the second case a macro. In either case the schedule is indirectly started by the route or the macro, respectively, by their operations. The actions performed before the schedule is actually started ensure, that the right train leaves the “Hidden Yard”.

In both cases we can additionally trigger the route or the macro, respectively, by appropriate start- and destination keys (see page 63) from a switchboard or an external control panel. This provides the possibility to select the train to be started from such panel, too.

Example: Hidden Yard with Train Length Control and Automatic Bypass

The hidden yard displayed below shall be operated automatically in the following way:

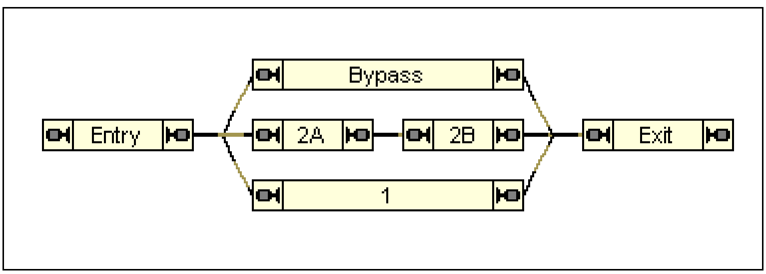


Diagram 145: Hidden Yard with Train Length Control and Bypass

- Trains enter the hidden yard through block “Entry” on the left and leave the yard through block “Exit” on the right.
- Long trains should go to block 1, if this track is available. If track 1 is already occupied, then long trains shall bypass the hidden yard via block “Bypass” and leave the hidden yard at once. Long trains must not enter track 2.
- It is assumed that two short trains fit into track 2. Short trains shall stop in block “2 B”, if track 2 is available. If there is already a train that is occupying “B 2”, then the next short train shall enter track 2, too, and stop in block “2 A”. If there are already two trains waiting in track 2 then the next short train shall go to track 1. If both tracks 1 and 2 are completely filled then a short train shall bypass the hidden yard via block “Bypass” and leave the hidden yard without stopping.
- When a short train that has been waiting in “2 B” leaves the yard then another short train waiting in “2 A”, if any, shall automatically move up to “2 B”.

The following schedules are created:

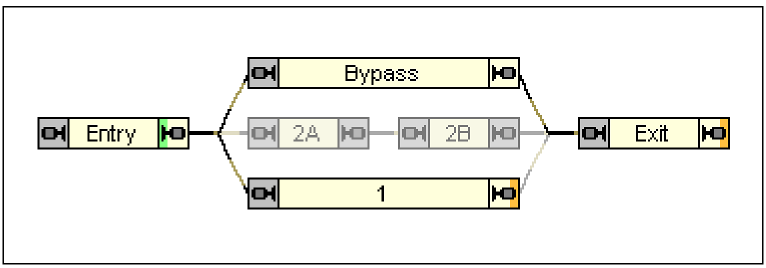


Diagram 146: Schedule for long Trains “Entry Long Trains“

The schedule to describe the entry of long trains is displayed in Diagram 146. The start block of this schedule is block “Entry”. The destination blocks are alternatively block “1” or block “Exit” via “Bypass”.

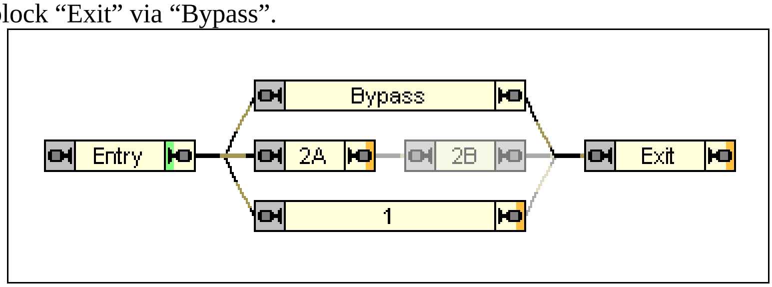


Diagram 147: Schedule for short Trains “Entry Short Trains“

The schedule for short trains is displayed in Diagram 147. The start block of this schedule is block “Entry” again. The destination blocks are alternatively block “2A”, block “1” or block “Exit” via “Bypass”.

The conditions listed in the following table ensure that the tracks are filled as required:

Schedule	Block	Condition	Remark
Entry Long Trains	Bybass	Block 1	May go to “Bypass”, only if block “1” is in use
Entry Short Trains	Block 1	Block 2A	May go to “Block 1”, only if block “2A” is in use
	Bypass	Block 1 **AND** Block 2A	May go to “Bypass”, only if block “1” <u>and</u> block “2A” are in use

Table 10: Conditions of Block Reservation

The schedule, that controls the exit of waiting trains, is shown below:

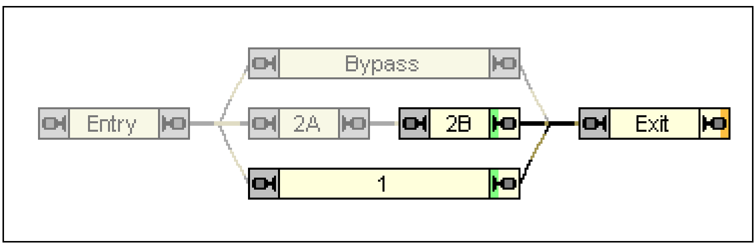


Diagram 148: Schedule “Exit“

For moving up trains from block “2A” to “2B” we need another schedule:

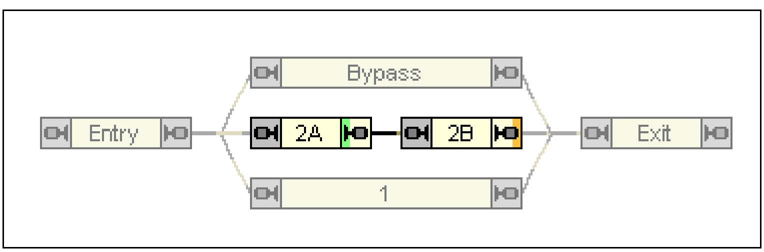


Diagram 149: Schedule “2A to 2B“

This schedule can be linked to schedule “Exit” as a successor. In this way each train that leaves the hidden yard controlled by schedule “Exit” will try to execute schedule “2A to 2B”, when it arrives in block “Exit”. If there is a train waiting in “2A” and if “2B” is free in this moment then this train will proceed to “2B”.

The two entry schedules can be linked as successors to other schedules that move a train from anywhere to block “Entry”. In the same way other schedules, that move a train from block “Exit” to anywhere can be linked as successors to schedule “Exit”. In this way the small configuration shown here can be integrated into the operation of a complete layout.

12.6 Timetables

It is possible to execute *schedules* or *macros* (see section 11.5, “Macros”) at specific times. Using a time table entry you can specify, at which days or at which times a schedule or macro shall be executed.

Schedules can be started daily, at specific days of the week or at a specific date, as desired. The latter feature enables creation of up to 365 different timetables. In this case the valid timetable is selected by setting the date of the *Clock*.

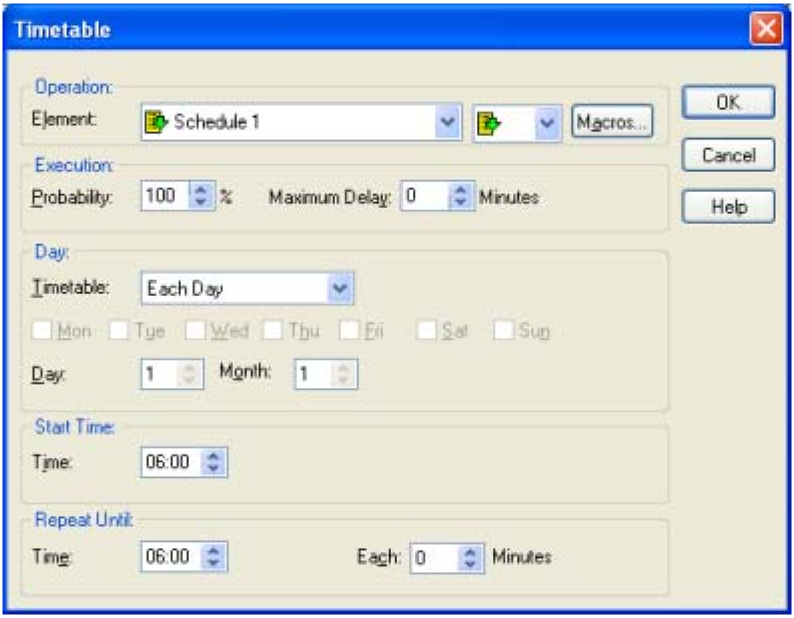


Diagram 150: Specifying the start time of a Schedule

Using these specifications the Dispatcher creates a *timetable* for the current day. The current day is determined by the date, which is currently displayed by the *Clock* (see chapter 10, " The Clock"). The *Dispatcher* starts the particular schedules or macros dependent to the time currently displayed by the *Clock*.

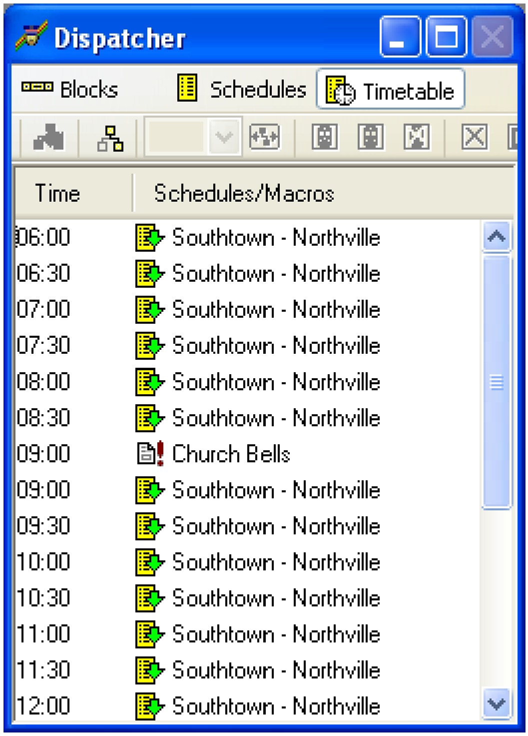


Diagram 151: Timetable in the Dispatcher

Using macros in timetables allow interesting effects. It is for example possible to turn on or off the lights on the model railroad or to play sound files at certain times.

The additional features to execute timetable entries only by chance or to insert random delays provide even more variety.

13 The Turntable Window

13.1 Introduction

The *Turntable Window* provided by **TrainController**™ is used to operate your turntables and transfer tables. Different *Turntable Windows* can be opened simultaneously to control several turntables/transfer tables at the same time - one for each turntable or transfer table on your model railroad. The number of Turntable Windows is only limited by the capacity of your computer.

The window can be configured to operate a turntable or to operate a transfer table as displayed below:

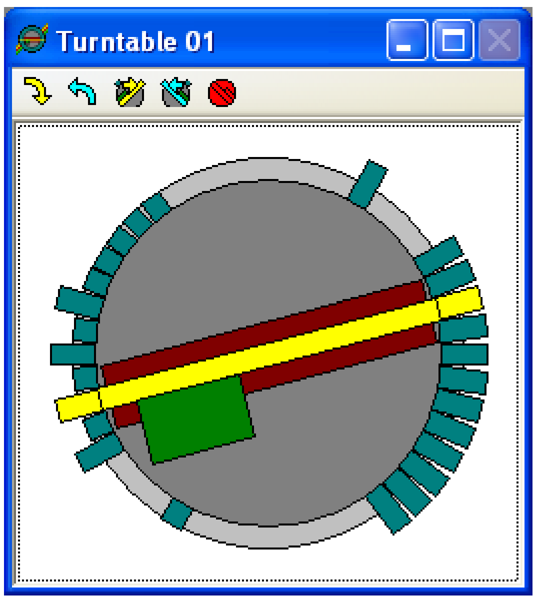


Diagram 152: Turntable

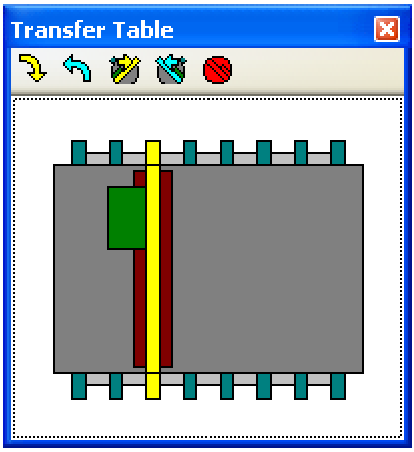


Diagram 153: Transfer Table

Special features are:

- up to 80 tracks on each turntable or transfer table
- each track can be individually configured as active or inactive as well as removed completely
- each turntable can be operated manually as well as automatically (e.g. by indicator elements or as part of routes and schedules of the Dispatcher
- predefined software drivers for all leading turntable types
- generic turntables and transfer tables allow adaptation to custom driven devices
- software supported hardware configuration for programming of decoder addresses or track configuration

Supported Turntable/Transfer Table Commands

TrainController™ supports the following turntable/transfer table commands:

- permanent move in either direction
- stop of permanent move with automatic alignment to the next active track
- step to the next or previous active track
- direct selection of specific tracks *(indexing)*
- 180° turn (turntables only)
- dedicated adjustment of locomotive orientation during automatic operation (turntables only, see page 216)

13.2 Configuring a Turntable or Transfer Table

To configure a turntable or transfer table use the **Properties** command of the **Edit** menu. Next, select if you want to control a turntable or transfer table and how many tracks can be connected to the turntable/transfer table.

Additionally you can specify a **name** for the turntable/transfer table This is useful in identifying the turntable/transfer table when it is referred to later. By specifying or measuring the **turn time** of the turntable/transfer table, you can assure that the movement of the bridge on the computer screen is synchronized with the movement of the actual bridge on your layout.

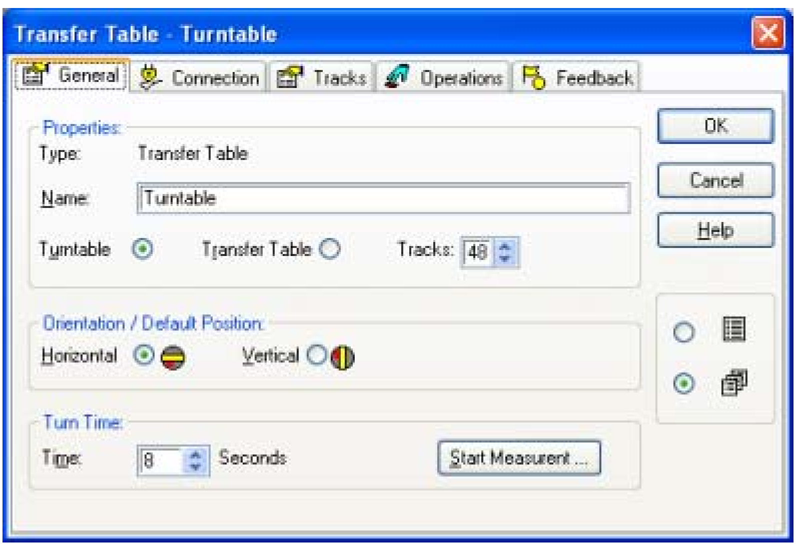


Diagram 154: Specifying general properties of a turntable

13.3 The Type of a Turntable/Transfer Table

TrainController™ supports the following types of turntables/transfer tables:

- Marklin Digital Turntable 7686
- Marklin Turntable 7286 with digital turntable decoder 7687
- Marklin Turntable 7186
- Fleischmann Turntable
- Digital Turntable Decoder Rautenhaus SLX815
- Marklin Transfer Table 7294
- Generic Turntables/Transfer Tables

Digital Turntable

A turntable is called a *digital turntable* if it is driven by a (built-in) digital turntable decoder. Examples of digital turntables are

- Marklin Digital Turntable 7686
- Marklin Turntable 7286 with digital turntable decoder 7687
- Turntable driven by the Digital Turntable Decoder Rautenhaus SLX815

! **Digital turntables support all commands listed on page 208. Specifically, they support the direct selection of specific tracks (*indexing*). Since indexing is vital for automatic operation digital turntables can be operated automatically without any limitations or special measures to be taken.**

Diagram 155: Specifying the type and digital address of a turntable

A digital address must be specified for each digital turntable. This is the digital address of the digital turntable decoder.

Analog Turntables/Transfer Tables

A turntable/transfer table is called an *analog turntable/transfer table* if it supports the following limited subset of turntable commands:

- permanent move in either direction
- stop of permanent move

Examples of analog turntables/transfer tables are:

- Marklin Turntable 7186
- Fleischmann Turntable
- Marklin Transfer Table 7294

The turntables/transfer tables listed above are not intended by the manufacturer to be operated by a digital system. Nevertheless, they can be made controllable by the computer. In this case, they must be wired to accessory decoders and, optionally, latching relays. When wired in this way, they are accessed via a digital turnout address. For details about digital addresses and wiring diagrams for the particular analog turntable/transfer table types, refer to the **Help** menu of **TrainController™**, please.

! **Analog turntables do usually not support indexing and cannot be used for automatic operation without further measures.**

It is also possible to configure analog turntables/transfer tables for support of indexing, too. In this way, it is possible to upgrade an analog turntable to a (pseudo-)digital turntable by means of the software. If this is done they can be used for automatic operation like digital turntables. For further details refer to section 13.7, "Turntable Operations", please.

Generic Turntables

Generic turntables/transfer tables are all turntables/transfer tables, that are not explicitly listed as devices supported by **TrainController™**. An example is a home- made turntable driven by custom hardware.

Generic turntables are not associated with a certain digital address. Instead they are only able to perform certain operations when one of the turntable/transfer table commands listed on page 208 is given. If no operation is specified for a certain command then a generic turntable/transfer table does nothing when this command is given.

Usually you will assign the operation of push buttons, on-off switches or toggle switches located anywhere in one of your Switchboards to a turntable/transfer table command. In this way the associated element is operated when the command is given. The associated element again can then operate the actual turntable on the model railroad layout accordingly.

Generic turntables can be setup to operate like analog turntables and – if operations for indexing are added as well – even like digital turntables.
For further details about the operations assigned to a generic turntable refer to section 13.7, "Turntable Operations", please.

13.4 The Track Layout of a Turntable/Transfer Table

After entering the maximum number of tracks that can be connected to the turntable/transfer table (see Diagram 154) you can specify the individual track layout of your turntable/transfer table.

The maximum number of tracks depends on the brand of the turntable/transfer table, typical numbers for turntables are 24 or 48, respectively.

The track layout describes which of the available tracks of your turntable/transfer table are connected to the tracks of your layout.

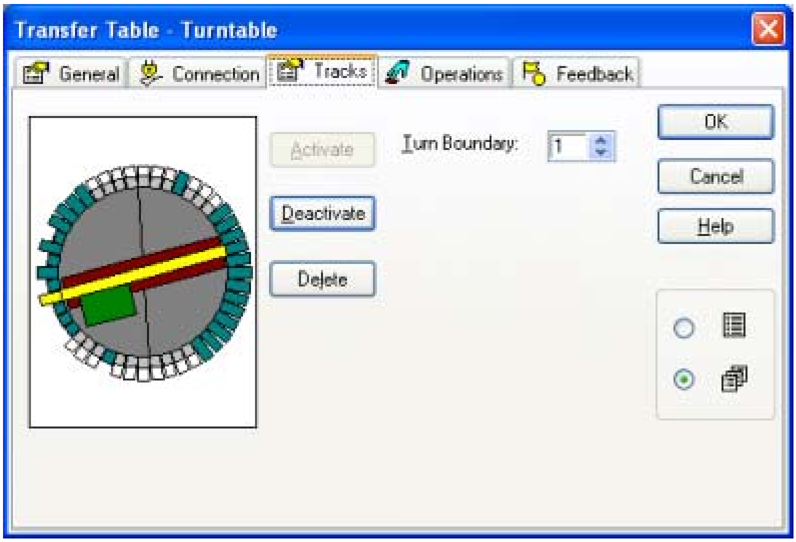


Diagram 156: Setting up the track layout of a turntable

By default all possible tracks of the turntable/transfer table are **active**, i.e. supposed to be connected to the rest of your layout. A certain track, track that is not used is called **not active**, if the opposite track is active. If neither a certain track nor the opposite track is active, then both tracks are **deleted**.

In Diagram 156 you can see active tracks on the right side of the turntable, deleted tracks on the upper and lower side and some inactive tracks on the left side.

13.5 Turntables and Feedback

It is possible to assign an indicator element to each active track, as well as to the bridge of a turntable/transfer table. Whenever such indicator is turned on, then the associated track or bridge his highlighted in the turntable window. In this way, it is possible to control via the turntable window if a specific track or bridge is occupied or not.

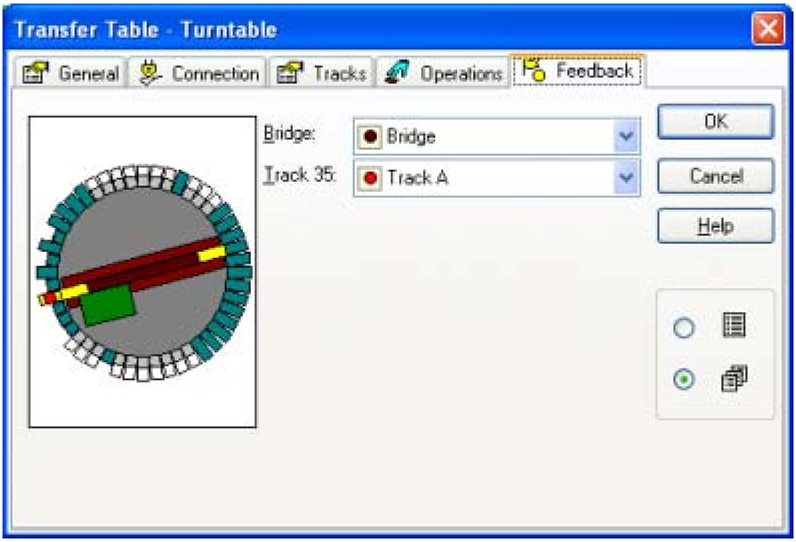


Diagram 157: Assigning feedback indicators to a turntable

If a Marklin Digital Turntable 7686 or 7286+7687 is operated through your turntable window, then three additional indicators are provided to display the state of the red, green and yellow indicator light connector provided by the turntable device.

13.6 Automatic Operation of Turntables

All turntable commands can be assigned as operations to other elements like *indicator elements*, *push buttons* or *routes* (see also section 11.3, “Operations”). In this way, turntables/transfer tables can be operated by passing trains, through your switchboard or automatically as part of *schedules* of the Dispatcher.

If you want to operate a turntable/transfer table automatically with the Dispatcher assign the desired turntable operations to an appropriate route first. If this route is then assigned to a schedule (see section 5.10, “Schedules”) each train running on that schedule will first trigger the turntable operation and wait until the turntable command has been executed completely. In this way it is assured that the turntable/transfer table has reached the desired position, before the train will enter or leave the bridge of the turntable/transfer table.

!

Please note, that the turntable/transfer table must support indexing in order to be operated automatically. If you are using an analog or generic turntable/transfer

table, then setup this turntable for indexing according to section 13.7, "Turntable Operations". Digital turntables support indexing and no further measures are necessary.

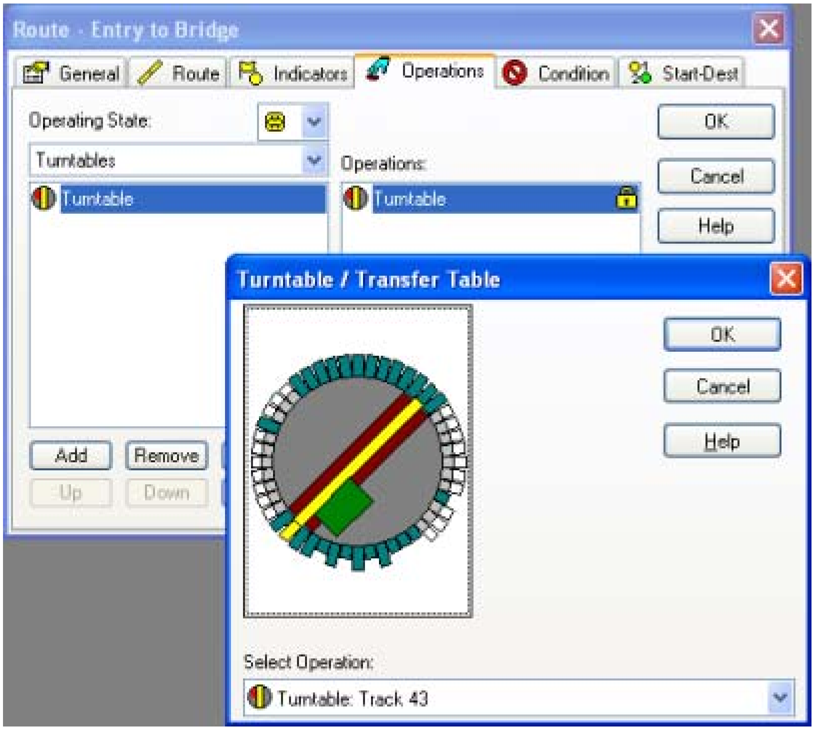


Diagram 158: Assigning a turntable to the operations of a route

Furthermore for automatic operation with the Dispatcher, additional special turntable commands are provided with which the orientation of each train can be controlled. For example, it is possible to implement the following scenario: assume there is a turntable which is located in front of a roundhouse. Each steam engine shall enter the roundhouse in backward direction to make sure that the funnel is located near the door. To ensure this, each steam engine entering the turntable must be turned appropriately depending on its current orientation. For this purpose, special turntable commands are provided that ensure that each engine can be turned, for instance, to the right when passing the turntable on its way to the roundhouse (for details see the following section).

Using different routes and schedules for steam engines and other engines it is even possible to turn steam engines to a dedicated orientation before they enter the roundhouse while other engines directly enter the roundhouse without any turn.

Turning Engines Automatically - The Turn Boundary of a Turntable

The turn boundary of a turntable, in conjunction with specific operations to move the bridge of a turntable automatically, are important features when a turntable is operated automatically under control of the Dispatcher.

With the turn boundary, it is possible to control the orientation of engines during automatic operation. It is assumed that an engine changes its orientation if the bridge passes the turn boundary while moving an engine. This is called a turn. For details about engine orientation refer to section 5.3, “Direction of Travel vs. Engine Orientation”, please.

The following does not apply for transfer tables and it applies only, and it only applies to a turntable if it is controlled automatically.

Turntables are able to perform different types of moves. This is illustrated in the diagrams below. These different types of moves can be assigned as turntable operations to a route, push button or indicator element, for example, in order to perform these moves automatically.

In the examples below, the turn boundary is marked as a red line. It divides the turntable into two parts. In all examples it is assumed that an engine enters the turntable from the left and leaves the turntable to the right.

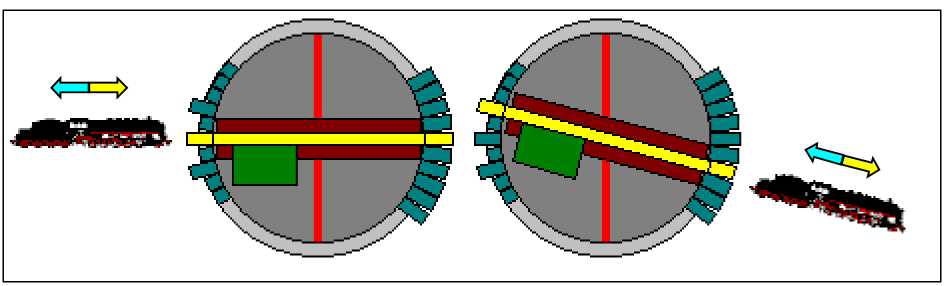


Diagram 159: Move without Turn

During a *move without turn*, the bridge of the turntable does not pass the turn boundary and the engine does not change its orientation. If you want to maintain the orientation of an engine during automatic operation, select this kind of move.

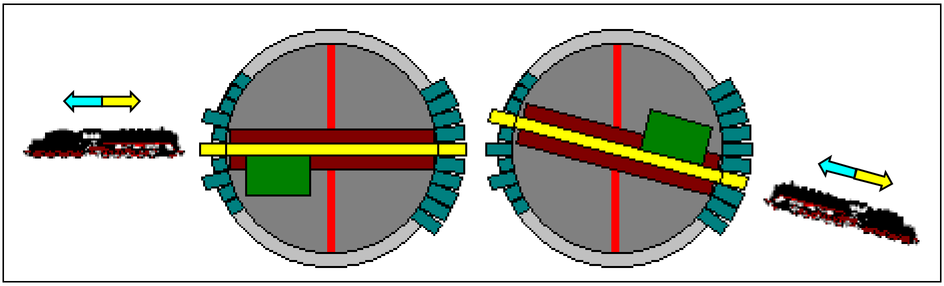


Diagram 160: Move with Turn

During a *move with turn*, the bridge of the turntable passes the turn boundary and the engine changes its orientation. If you want to enforce a change of the orientation of an engine during automatic operation, select this kind of move.

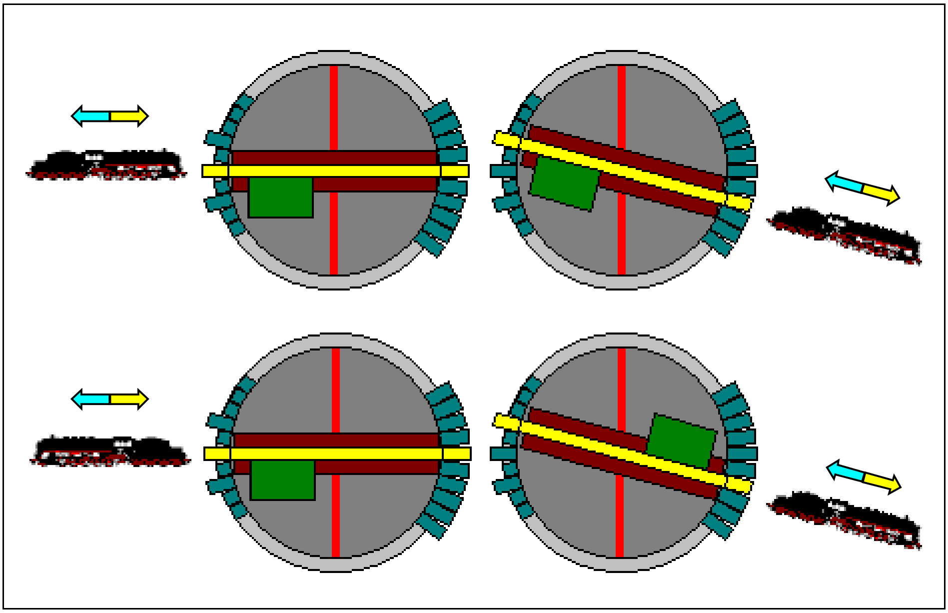


Diagram 161: Move with Turn to the Right

During a *move with turn to the right or bottom,* the bridge of the turntable may or may not pass the turn boundary. This depends on the orientation of the entering engine. If the orientation of the engine points already to the right or bottom when it enters the bridge, then the turntable does not perform a turn. The engine leaves the bridge with the same orientation as before. If the orientation of the engine points to the left or top when it enters the bridge, then the turntable performs a turn in order to change the orientation of the engine to the right or bottom. When the engine leaves the bridge its orientation points to the right or bottom in both cases. A move with turn to the left/top is performed accordingly. If you want to enforce an engine to leave the bridge with a certain orientation, select this kind of move.

If you assign a move with turn to the right/bottom (or left/top) direction as an automatic operation of the turntable to a route, for instance, and this route is activated during automatic operation by the *Dispatcher*, then **TrainController**™ evaluates the current orientation of the affected engine and automatically performs a move with or without turn depending on this current orientation. In this way it is possible to ensure that an engine leaves the bridge in a dedicated orientation regardless of its previous orientation.

When the turntable is directed to perform a specific move and this move is not performed under control of the Dispatcher or the bridge is not occupied by an engine, then a *direct move* is performed in all cases, i.e. the fastest (shortest) possible move to reach the destination track. The bridge is assumed to be occupied if the indicator element associated with the bridge is turned on. If no indicator element is associated with the bridge, then the specified move is performed anyway.

!

Since a direct move is more prototypical in cases where no engine is located on the bridge, it is recommended to assign a feedback indicator to the bridge, if possible. Otherwise TrainController™ is not able to determine if the bridge is free or not while deciding to perform a direct move or not. Without a feedback indicator assigned to the bridge TrainController™ will never perform a direct move when under control of the *Dispatcher*.

!

Usually you will also associate a block of the Dispatcher with the bridge of an automatically operated turntable. Stop indicators assigned to this block are needed to stop each engine automatically at the correct location on the bridge.

The only other thing you will have to do is to adjust the turn boundary carefully. The rest is done automatically by the software. In the usual case where a turntable is located in front of a roundhouse you will adjust the turn boundary in a way that the accessing tracks are located on one side of the turn boundary and the roundhouse is located on the other. In the diagrams displayed above you can assume that three accessing tracks are located on the left side of the turn boundary and the roundhouse is located on the right side.

More generally the following rule applies: if the tracks connected to a turntable are associated with blocks, then the blocks located on one side of the turn boundary face the turntable with their right exit, the blocks on the other side of the turn boundary face the turntable with their left exit. This is illustrated in the following diagram:

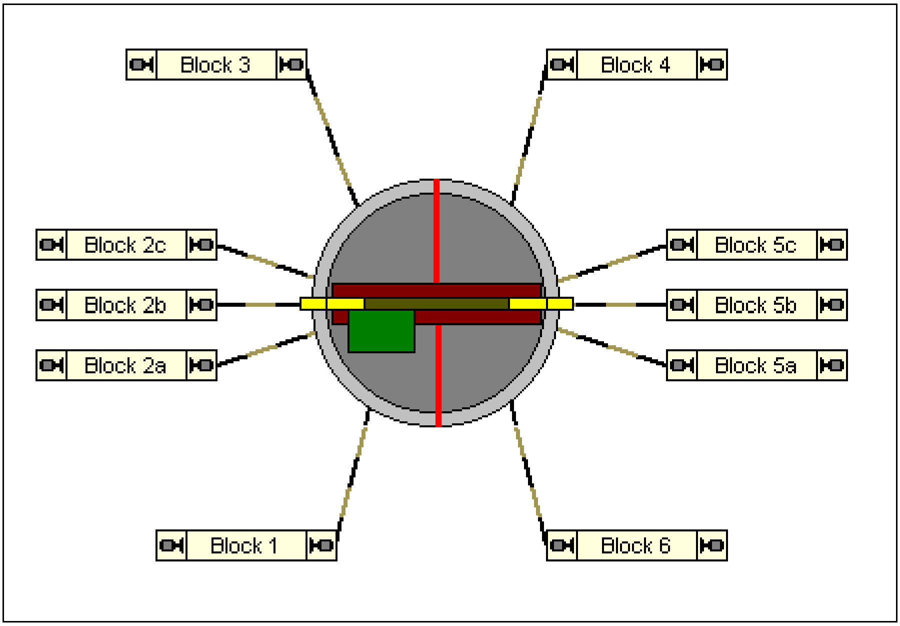


Diagram 162: Turn Boundary

The blocks on the left side of the turntable are facing the turntable with their right exit. The blocks on the right side of the turntable are facing the turntable with their left exit.

!

All blocks on the same side of the turn boundary must face the turntable with the same exit.

Even though it might be tempting in some cases to draw some blocks located around the turntable as horizontally aligned blocks and the others as vertically aligned blocks it is more clear to draw all either horizontally or vertically. This has only a visual impact, though, but if all blocks are drawn with the same alignment it is easier to see, whether the turn boundary has been adjusted correctly.

The turn boundary does not apply to transfer tables.

Example: Turntable and Roundhouse

In this example, it is assumed that a turntable is located in front of a roundhouse. Each steam engine shall enter the roundhouse in backward direction to locate the funnel near the door. To ensure this, each steam engine entering the turntable must be turned appropriately depending on its current orientation. This is done by operating the turntable automatically by routes and by using the special turntable commands explained above.

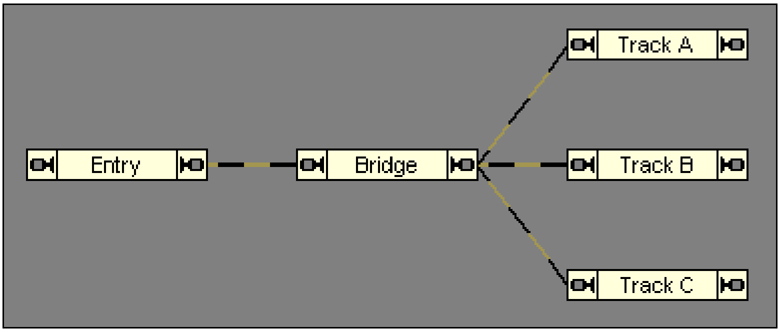


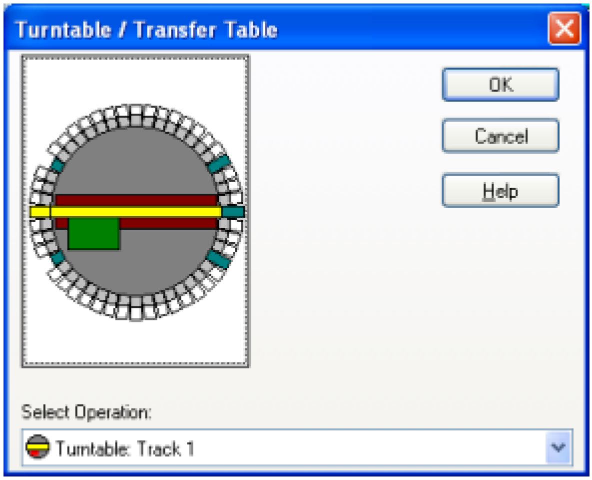
Diagram 163: The block diagram

The block diagram is displayed in Diagram 163.

- There is one block "Entry" on the left side of the turntable.
- On the right side of the turntable there are three blocks assumed to be located in the roundhouse. These blocks are called "Track A", "Track B", "Track C".
- There is an additional indicator wired to the bridge of the turntable, i.e. the indicator is turned on, when an engine is located on the bridge. This indicator is associated with block "Bridge". This indicator is also assigned to the bridge of the turntable according to Diagram 157.
- A Turntable Window is created and the track layout of the turntable is configured accordingly. The turn boundary is adjusted to separate the block "Entry" from the blocks in the roundhouse.
- There are four routes. The first route is called "Entry to Bridge" and the other are called "Bridge – Track A", "Bridge – Track B", "Bridge – Track C". The first route is used to turn the bridge to track "Entry" to allow engines to enter the bridge from the left. The route "Bridge – Track A" is used to turn the bridge to "Track A" to allow engines to leave the bridge and enter the roundhouse via this track. The parallel tracks are operated accordingly. This is configured as operations of the routes according to Table 11. The operation assigned to the route "Entry to Bridge" is a normal or direct move to track "Entry". When this route is selected no engine is located on the bridge and the bridge shall move to track "Entry" on the shortest possible (direct) way. The operation assigned to the other routes is specified as **turn to the left/top** because the engine shall enter the roundhouse in backward direction facing the bridge of the turntable, i.e. with orientation to the left.

Route	Operation
Entry to Bridge	Turntable Track 1
Bridge – Track A	Turntable Track 21 – Turn Train to the left/top
Bridge – Track B	Turntable Track 25 – Turn Train to the left/top
Bridge – Track C	Turntable Track 29 – Turn Train to the left/top

Table 11: Route configuration



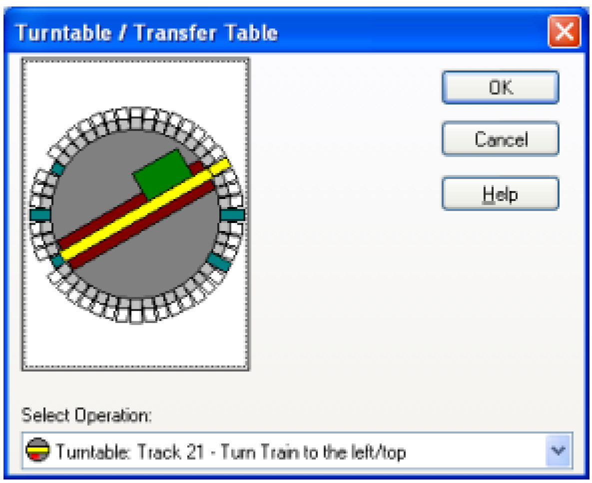


Diagram 164: Setting up the routes for automatic turntable operation

- Next step is configuration of an appropriate schedule.

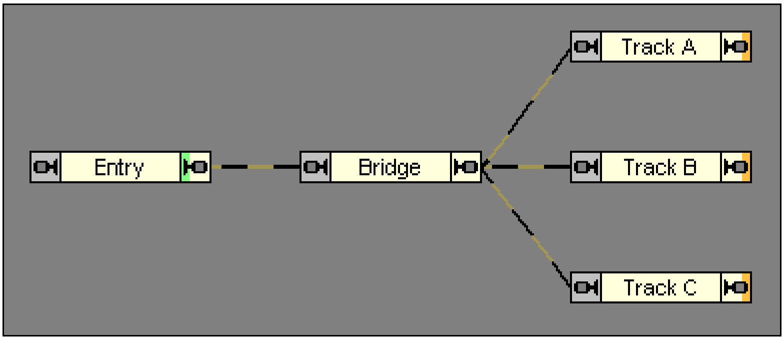


Diagram 165: Schedule for automatic turntable operation

We can operate the complete example with one schedule. Block “Entry” is specified as start block and the blocks, that belong to the roundhouse are specified as destination blocks.

- Finally an engine is created and assigned to block “Entry” facing to the right.

If this has been configured, then you can start the schedule and the following sequence is running automatically without further intervention:
If the schedule “Entry – Roundhouse” is started, then the route “Entry to Bridge” triggers the turntable to perform a direct move to track “Entry”. The route is activated after completion of this move and the engine is started forward to enter the bridge.

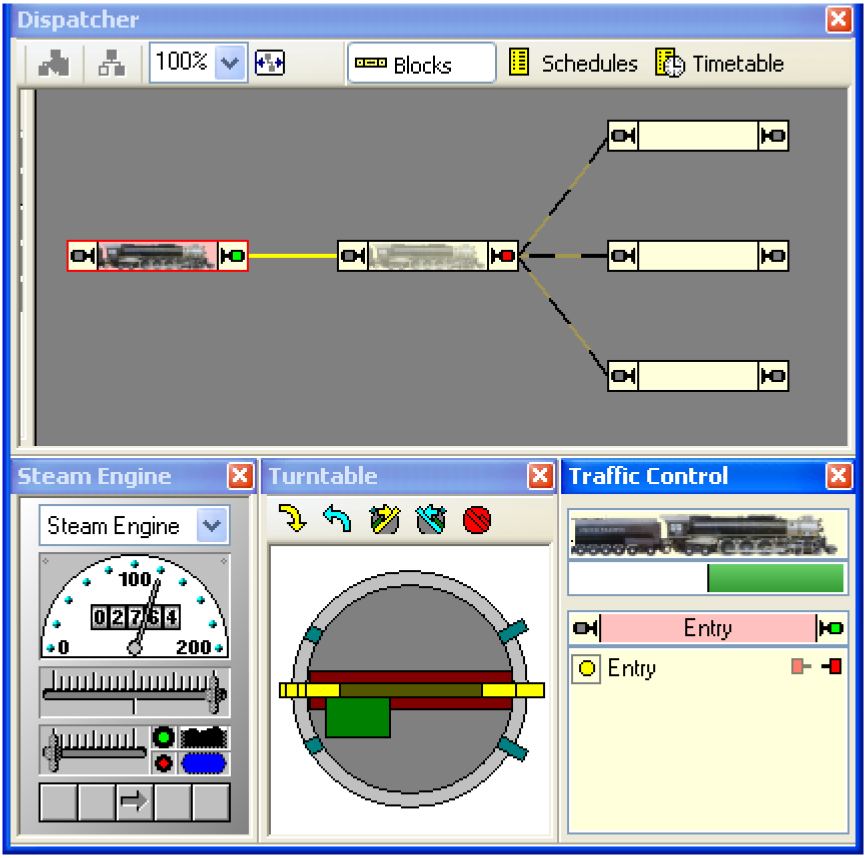


Diagram 166: Engine is moving forward to the bridge

After arriving at the indicator associated with the bridge, that is used as stop indicator of block "Bridge", the engine is stopped. The first route is released and a route to a roundhouse track is selected and activated. The turntable is started to move with a turn to the left. Since the engine is facing to the right, a move with passing of the turn boundary is performed automatically.

The following Diagram 167 shows the turntable during the turn just after passing the turn boundary. The engine is waiting for the completion of the turn, the block, that belongs to the selected roundhouse track, is displayed as reserved and the *traffic control* shows that the engine orientation has changed. The bridge of the turntable is indicated as occupied, because the indicator assigned to the bridge is turned on.

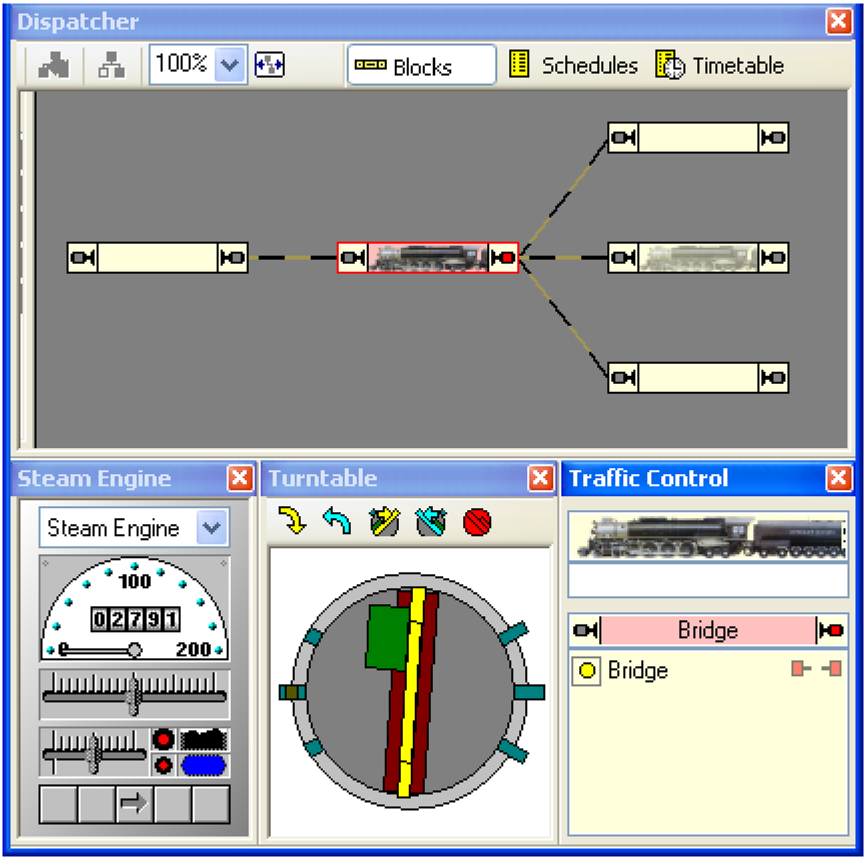


Diagram 167: During the turn

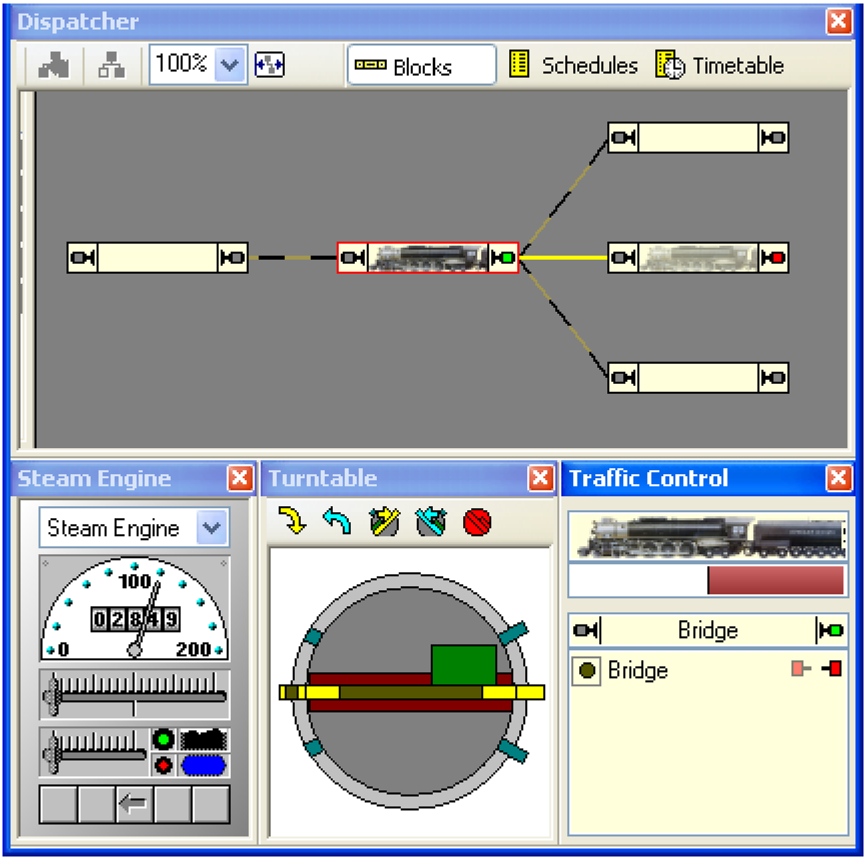


Diagram 168: Leaving the bridge

After completion of the turn the route is reported as activated and the engine is started backward to leave the bridge (Diagram 168).

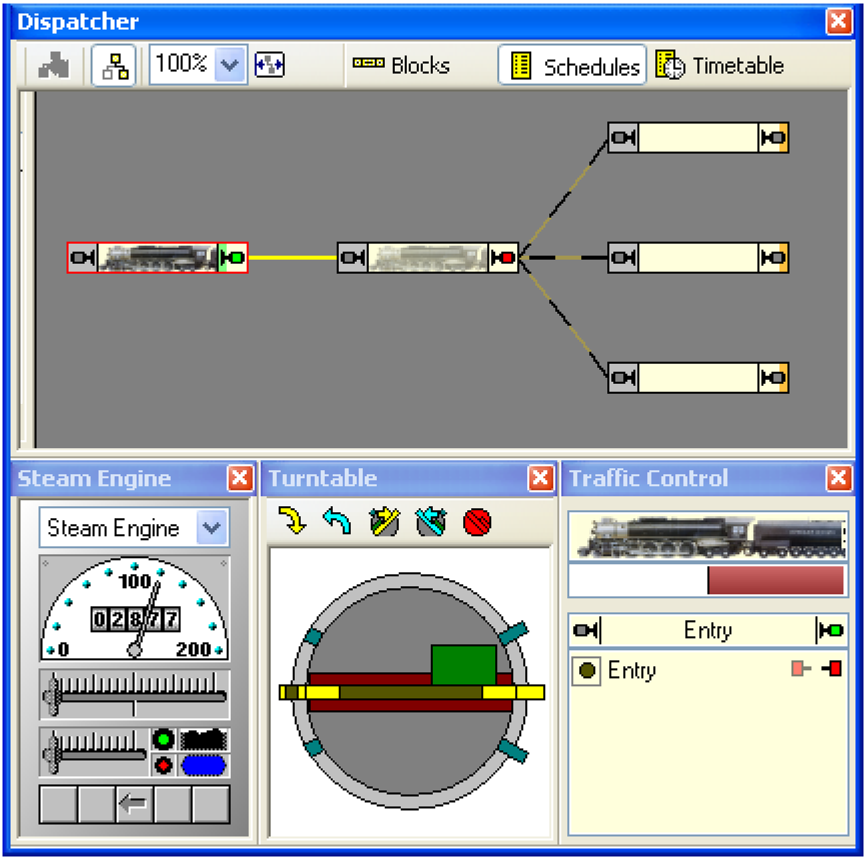


Diagram 169: Engine is moving backward to enter the bridge

If the engine is assigned to block "Entry" facing to the left then the engine moves backward to the bridge after start of the schedule. Compare Diagram 166 with Diagram 169: in the first case the engine is moving forward, in the second case the engine is moving backward.

Since the engine is already oriented to the left only a short or direct move, respectively, to the selected roundhouse track is performed. Compare Diagram 167 / Diagram 168 with Diagram 170. In the first case a long clockwise turn is performed in order to change the orientation of the engine from right to left. During this move the bridge passes the turn boundary. In the second case a short/direct move can be performed because the orientation of the engine is not to be changed.

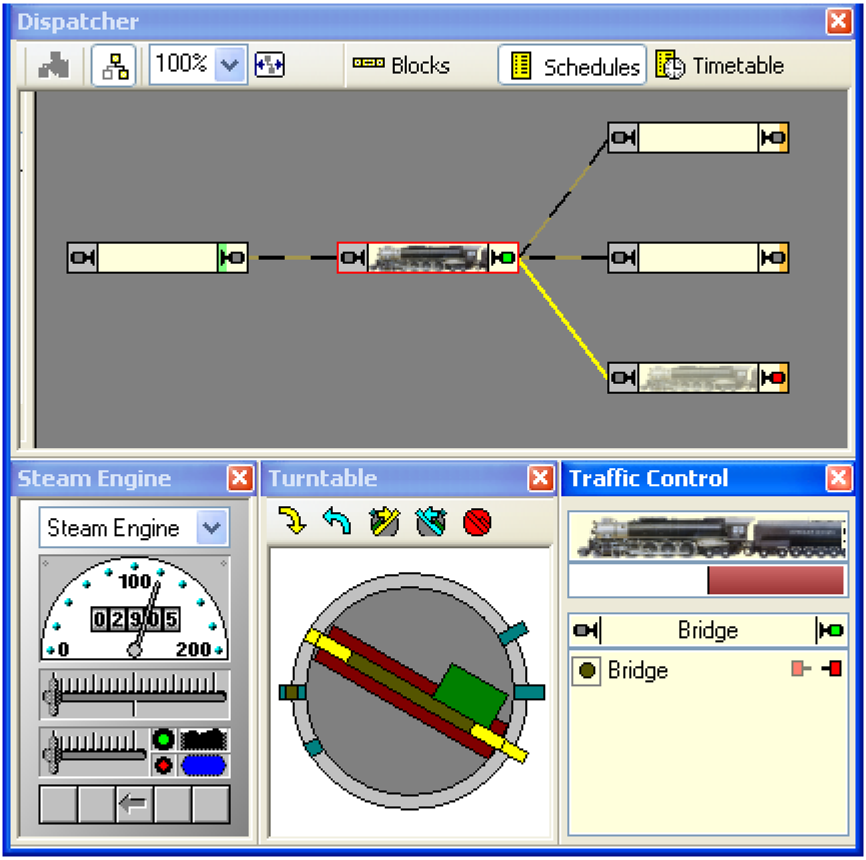


Diagram 170: After the (direct) turn

It sounds complicated, but all of the actions outlined in this example are done automatically by the software. The only things you have to care about is an appropriate adjustment of the turn boundary and an appropriate specification of routes with correct operations of the turntable.

13.7 Turntable Operations

For each command listed on page 208 it is possible to specify a certain operation, that is executed when this command is given (see also section **11.3**, “Operations”). These operations are mainly intended to be used for upgrading of analog turntables to operate like digital turntables. This is done by adding operations for *indexing*.

And these operations are used to setup a generic turntable/transfer table to operate like an analog or digital turntable.

They can be used additionally by digital turntables for special purposes as well, if desired.

Usually you will assign the operation of push buttons, on-off switches or toggle switches located anywhere in one of your Switchboards to a turntable/transfer table command. In this way, the associated element is operated when the command is given. The associated element can then operate the actual turntable on the model railroad layout accordingly, e.g. via relays wired to accessory decoders.

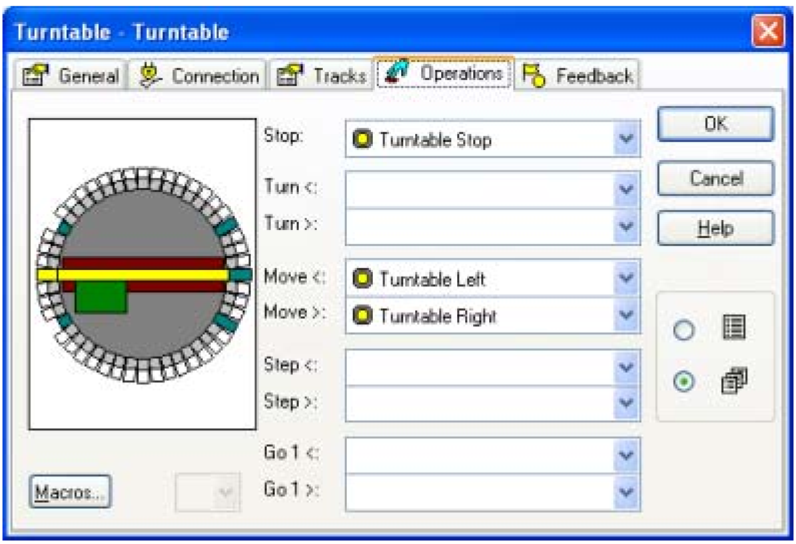


Diagram 171: Assigning operations of buttons to a turntable

If, for example, operations are assigned to a generic turntable according to Diagram 171, then this turntable can be operated like an analog turntable. With this setup of operations, a generic turntable can perform exactly the same commands like an analog turntable.

It is also possible to operate macros. In conjunction with evaluation of indicator elements and restricted execution of operations (see 11.2, “Protection and Locking with

Conditions"), it is even possible to setup *indexing* for analog or generic turntables/transfer tables. This is demonstrated in the following example.

Example: Indexing of an Analog Turntable

This example explains how an analog turntable, such as the Fleischmann Turntable or the Marklin Turntable 7186, can be setup for indexing in order to be controlled automatically. It is assumed that the track layout of the turntable is identical to Diagram 166. In the following, it is explained how indexing is setup for track "A" (see page 220ff). The setup for the other tracks is done accordingly.

- Wire the analog turntable according to the instructions in the **Help** menu of **TrainController**™.
- A Turntable Window is created and the track layout of the turntable is configured accordingly.
- Create a feedback indicator "Track A" that is turned on, when the physical bridge of the turntable reaches the position of track "A". Of course you need an appropriate physical sensing device on your model railroad layout that is able to detect and report when the bridge reaches this position. This indicator is used to trigger stopping of the bridge at the destination position.
- Create an on-off switch "Track A". This on-off switch is used to trigger turntable movement and to act as a memory in order to stop the bridge at the correct position.
- Assign the move of the turntable bridge (in any direction) as operations to the on-off switch "Track A".
- Assign the on-off switch "Track A" as turntable operation to the turntable according to the image below.

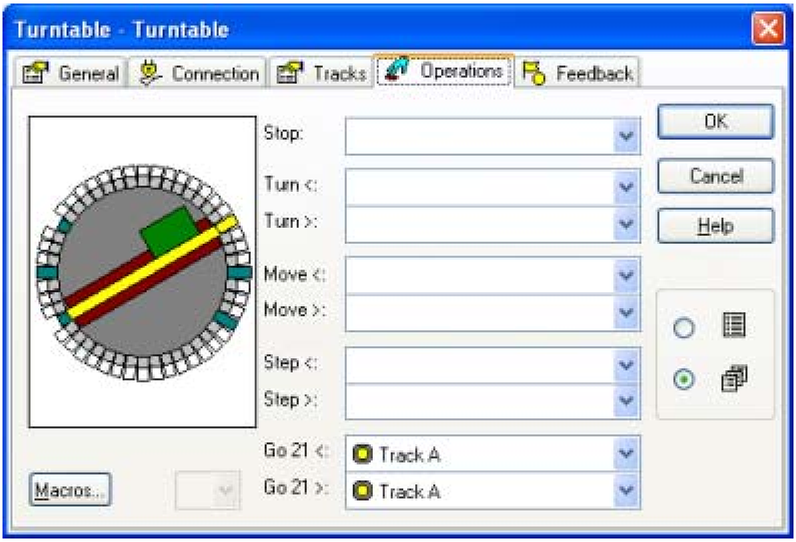


Diagram 172: Assigning operations of a on-off switch to a turntable

- Create a flagman "Track A" and assign the indicator "Track A" as trigger. This flagman is used to stop the turntable when the bridge reaches track "A".
- Assign the state "on" of the on-off switch "Track A" as condition to the flagman. In this way it is ensured that the bridge is stopped at track "A" only if it should do so.
- Assign the command to stop the turntable as operation to the flagman.
- Via the operations of the flagman the on-off switch "Track A" should be turned off again in order to return to the initial state.

How it works:

If the turntable is instructed to go to track "A", then the on-off switch "Track A" is turned on. This on-off switch starts the bridge to move. When the physical bridge of the turntable reaches track "A", the feedback indicator "Track A" is turned on. This again triggers the flagman, that is turned on, because the on-off switch "Track A" acting as a memory is still turned on. The flagman then stops the turntable.

Notes:

This is a very rough explanation of the setup. Detailed instructions would be out of proportions of this manual. This example should give you a first idea how the mechanism works in principle.

The key is the usage of the on-off switch as a memory. It is turned on at the beginning of the move to the destination track and it ensures that the turntable is correctly stopped at the right track.

Normally you will create two on-off switches for each track, one for each direction.

A problem might arise due to the fact that in many cases stopping of the bridge must be triggered just <u>before</u> the bridge reaches the destination track rather than just after arrival in order to stop the movement in time.

Transfer tables are setup for indexing accordingly.

Setup of a generic turntable for indexing is done in the same way. The only further measure to be taken is setup of additional operations for the commands normally supported by analog turntables as displayed in Diagram 171.

14 Special Applications

14.1 Mixing manual and automatic Operation

TrainController™ will not supersede you – the human operator. The software can make large scale railroad operations manageable by one person, matching operations found on the largest club layouts. In many cases, several trains will run automatically under control of the computer while certain other trains remain under manual control of the human operator.

Very often certain parts of the layout are controlled fully automatically by the computer (e.g. hidden yards) while other parts of the layout remain under complete control of the human operator (e.g. fiddle yards). In this section, it is outlined how trains can be passed from manual to automatic control or vice versa.

A typical example is displayed in the block diagram below:

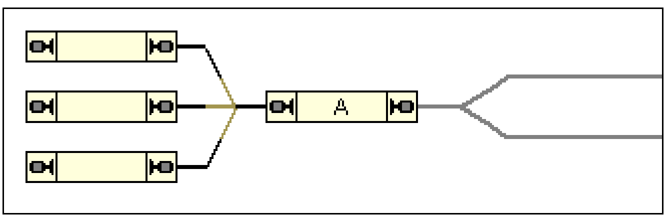


Diagram 173: Mixing manual and automatic operation

On the left side of the layout, a hidden yard is located. This hidden yard is operated fully automatically by the computer. On the right side of the layout, a small yard is located that is operated manually.

The left part – the automatic part of the layout – is equipped with indicators in each siding. A block diagram with blocks and routes and additional schedules have been created to control entry and exit of trains into and out of the left part of the layout automatically.

The right part – the manual part of the layout – is not included into the main block diagram. The track layout is indicated in Diagram 173 with gray lines.

Passing trains from manual to automatic control

The key is the block marked with an “A”. It marks the interface between the manual and the automatic part of the layout. If trains leaving the manual part of the layout shall be passed to automatic control without further interaction a *train detection device* is needed here (see section 5.5, “Train Detection and Train Tracking”). Such a device is able to detect which train is about to enter the automatic part of the layout. If a *contact indicator* is associated with the train detection device and this indicator is again assigned to block “A”, then **TrainController**™ will perform the assignment of each detected train to block “A” automatically.

Additionally, you can assign a *schedule* of the Dispatcher as *operation* to the *contact indicator* associated with the *train detection device*. If this is done, then the manually operated train passing the train detection device is not only detected and assigned to block “A”, but a schedule of the Dispatcher is also started, that runs the train automatically to a free block in the hidden yard.

In this way the train is passed from manual to automatic control without further interaction.

In many cases, the manual part of the layout is not even known to **TrainController**™. Indeed it is not necessary to include the parts of the layout, that are not operated by the computer, into the block diagram. Only the automatic part of the layout including all engines and trains, that are to be operated by the computer, must be known to the *Visual Dispatcher*. Control of each engine can be assigned to the digital system (see section 3.11, “Passing control between Computer and Digital System”). When an engine passes block “A” on its way from the manual to the automatic part of the layout and a schedule is started with this engine at block “A”, then the software will gain control of the engine automatically. When the schedule is finished, control is given back to the digital system and the engine can then be controlled manually.

Passing trains from automatic to manual control

With the features outlined above the automatic pass of engines from manual to automatic control is supported.

There is a special option for the opposite direction as well. This option is called **Release Last Block** and should be set as a property of all automatic schedules ending in block “A”. Normally – if this option is not set – each engine finishing a schedule in block “A” will keep this block reserved permanently - even after the engine has been taken over by manual control. As long as this block remains reserved no other engine will be able to perform another schedule ending in this block. To prevent you from being forced to release such blocks manually set this option for all automatic schedules ending in blocks where engines are passed to manual control. If this option is set for a schedule then the destination block is automatically released when the schedule is finished.

Passing trains from manual to automatic control without the use of a train detection system

It is also possible to pass trains from manual to automatic control without the use of a train detection system. This is done by means of *train tracking*. In this case the manually operated part of the layout must be equipped with track sensors and this part of the layout must be included into the block diagram, too. An example how this is done is outlined on page 160.

14.2 Running Conventional Engines without Decoder

Stationary Block Decoders

TrainController™ provides the possibility to control conventional engines, i.e. locomotives without an own engine decoder. This is done with *stationary block decoders*, i.e. decoders or computer controlled throttles, which are mounted at fixed positions on your model railroad rather than in each locomotive.

This feature is useful,

- if you have a large collection of locomotives and not all are digitally upgraded.
- if you have a conventional - i.e. non-digital - operated model railroad and want to control it with your computer without installing an engine decoder in each locomotive first.
- if the models of your engines are very small and the decoders do not fit into the engines (e.g. when you run Maerklin Mini Club).

In all **TrainController™** provides three methods of operating your trains, which are explained below:

- Operating trains with individual engine decoders ("Computer Command Control").
- Operating trains with stationary block decoders with static assignment to track sections ("Computer Section Control").
- Operating trains with stationary block decoders with dynamic assignment to track sections ("Computer Cab Control").

Additionally it is possible to use all these methods simultaneously, i.e. it is possible, to run conventional engines and digital engines at the same time.

Computer Command Control

This is the method supported by most of the digital systems of today. This is also the only method supported in the first versions of **TrainController™** In this case each engine is equipped with an individual engine decoder and can be operated directly by sending speed or function commands to the decoder. More details are explained in the documentation of the digital system.

Computer Section Control

This method is also called "*Computer Block Control*" or "*One throttle per section*". In **TrainController™** this kind of operation is based on the *blocks* of the *Dispatcher*. Unlike *Computer Command Control* it is possible to operate conventional locomotives with this method.

In this case all *blocks*, in which conventional locomotives should be able to run, must be electrically insulated from each other. Additionally each block is electrically connected with a specific decoder, which is mounted at a fixed position of your model railroad. The power in each block is controlled by the associated decoder. This results in a static assignment between each block and a stationary block decoder. To assign a block to its own stationary block decoder you have to assign a digital address to each block - namely the address of the connected stationary block decoder. Whenever a block is reserved for an engine or train, then all consecutive engine commands are sent to the stationary block decoder, which is connected to the block, instead to the engine itself. Since several blocks can be reserved for an engine or train, **TrainController™** sends engine commands to all affected blocks.

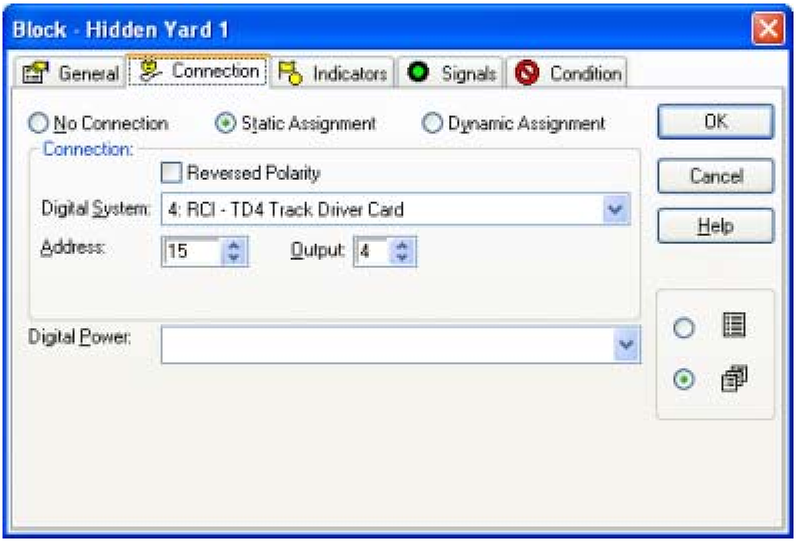


Diagram 174: Computer Section Control - Specifying the Digital Address of a Block

Computer Cab Control

This method is also called "*Progressive Cab Control*". In **TrainController™** this kind of operation is based on the *blocks* of the *Dispatcher*. Unlike *Computer Command Control* it is possible to operate conventional locomotives with this method. This method supports also the possibility to run digital and conventional engines on the same track.

Unlike *Computer Block Control* there is no permanent electrical connection between blocks and *stationary block decoders*. For this reason the number of stationary block decoders may be lower than the number of affected blocks.

All blocks in which conventional locomotives should be able to run, must be electrically insulated from each other. The electrical connection between blocks and decoders is established when required. This results in a dynamic assignment between each block and one of several stationary block decoders, which are mounted at fixed positions of your model railroad. The power in each block is controlled by a dynamically assigned decoder.

In order to arrange Computer Cab Control for a specific block, you have to specify a list of digital addresses **-** namely the digital addresses of the stationary block decoders, from which one should be dynamically selected. But there is one more thing to do: when a stationary block decoder is selected for a specific block, then the power generated by this decoder must be routed to the block. In order to establish the electrical connection you have to specify an *on-off switch* (see section 2.5, “Signals and Accessories”) for each stationary block decoder, which shall be used to turn on or off the connection between the block and the decoder. In most cases a sequence of switching operations (e.g. a sequence of several relays) must be operated to establish the connection between a block and a stationary block decoder. In this case make use of the possibility, to assign a set of *operations* (see section 11.3, “Operations”) to an on-off switch.

Whenever a block is reserved for an engine or train, then the Dispatcher searches an appropriate stationary block decoder. If a decoder was found, then the on-off switch, which is associated with the connection between the block and the decoder, is automatically turned on. When the block is released, this on-off switch is automatically turned off again.

If you have arranged your blocks correctly, then you do not have to take care of the dynamic assignment of decoders to blocks and the routing of the electrical power from the decoders to the affected blocks. This is done automatically by the Dispatcher.

Of course it is possible to arrange your blocks in a way, that one decoder can control several blocks simultaneously, which are reserved for the same train.

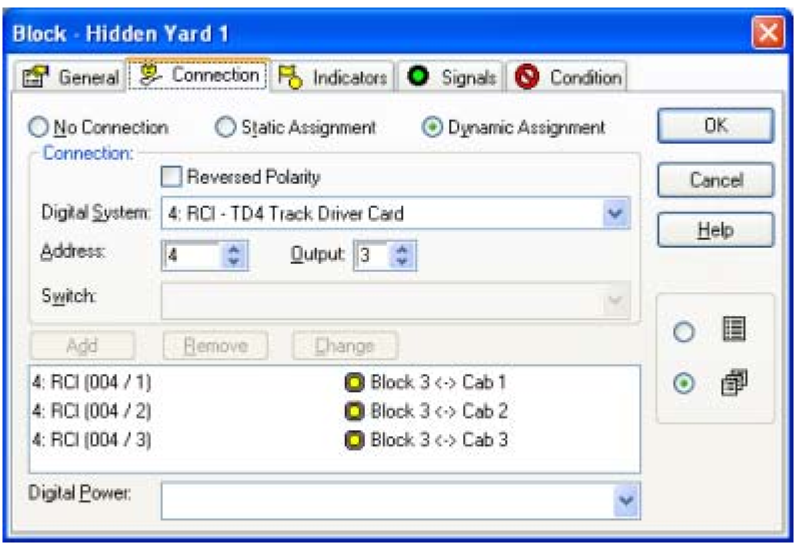


Diagram 175: Arranging a Block for Computer Cab Control

Adjusting the Polarity of each Block

! In order to direct each train to the correct direction of travel and in order to avoid short-cuts **TrainController™** applies a *logical polarity* attribute to each block. For each block **TrainController™** assumes that the following is true:

If a train is located in a block heading to the right/bottom and if the train is directed to move forward then the train moves to the right/bottom.

Unlike computer command control, where this condition is usually true, if the decoder is installed properly, this is not always true when stationary block decoders are used. The direction in which the train moves depends on the wiring of each block. In order to let each block fulfil the above rule without rewiring of your layout **TrainController™** provides an option to adjust the *logical polarity* of each block in the software (see Diagram 174 and Diagram 175).

It is very easy to adjust the polarity of each block in **TrainController™**. Perform the following steps:

- Put a train on the track inside of the block.
- Make sure that the train is heading to the right or bottom, respectively.
- Assign the train to the block in the *Visual Dispatcher*.
- Make sure, that the train image in the block symbol of the *Visual Dispatcher* is also heading to the right or bottom, respectively.
- Select the train in the *Train Window*.
- Drag the speed slider in the *Train Window* to the right.
- If the actual train on the layout is now moving to the right or bottom, respectively, then the polarity of the block is correctly adjusted. Otherwise open the properties of the block and change the polarity of the block by toggling the **Reverse Polarity** option.

Look at the following example:

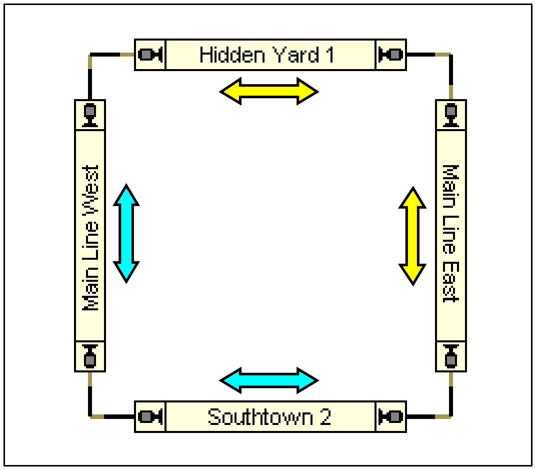


Diagram 176: Block Diagram of a Circular Layout

It can be assumed that the physical wiring of the layout displayed above is done in a way that the track power will not change its polarity when a train cycles around the loop. In other words: the <u>physical</u> polarity of all blocks in the above diagram can be assumed to be identical.

The situation in **TrainController™** is different. **TrainController™** does not want to rely on the fact, that the layout has been wired in a certain way. Additionally, the structure of many layouts is much more complicated. It can contain reversing loops or several levels, it can be based on a modular structure, etc.

For this reason **TrainController™** uses the *logical polarity scheme* described above. If the layout displayed above is wired in a way that the track power will not change its polarity when a train cycles around the loop, then the train will pass "Hidden Yard 1" and "Southtown" to different logical directions (left or right), even though the physical polarity of the track power remains unchanged. A train that passes "Hidden Yard" to the right at positive track polarity will pass "Southtown 2" to the left at the same track polarity. As a consequence "Hidden Yard 1" and "Southtown 2" have different *logical polarity* from the point of view of the software. The differences with regard to the *logical polarity* of the particular blocks are marked with a yellow or blue arrow in the diagram displayed above.

Running conventional and digital Engines on the same Track

This is supported with an additional option. Each block, on which conventional engines as well as digital engines shall be able to run, must be arranged for dynamic decoder assignment (*Computer Cab Control*, unless the RCI system is used - see below). Additionally it is possible, to assign one extra *on-off switch* to each affected block (see Diagram 175). This additional on-off switch is used to turn on and off the “digital power” for this block. Whenever the block is reserved for a conventional engine, then the block is automatically connected to an appropriate stationary block decoder as outlined in the section before. When the block is reserved for a digital engine, then the extra on-off switch is used to turn on the “digital power” for this block.

In this way it is even possible to run conventional and digital engines in different blocks of the same track at the same time.

The Track Driver Cards of the RCI system provide a built-in feature to route DCC power directly to the output points. This feature is used, when a block is statically assigned to a stationary block decoder on a Track Driver Card (*Computer Section Control*). Whenever a block is reserved for an engine with an own DCC decoder, then the DCC mode is automatically turned on for the Track Driver point connected to this block. When the block is released, then the DCC mode is turned off.

Notes

You can use regular engine decoders of any digital system as stationary block decoders. To use an engine decoder as stationary decoder, mount it at a fixed position of your model railroad and connect the wires, which are normally connected to the motor, to the track instead. To be on the safe side you should ask the dealer or manufacturer of the engine decoder, if it can be used as stationary block decoder without the risk to be damaged. The author of the program shall not be liable to you for any damages.

TrainController™ supports also digital systems, which provide computer controlled throttles dedicated to be used as stationary block decoders (e.g. the systems RCI or CTI).

The operation of conventional locomotives with stationary block decoders is based on the blocks of the *Dispatcher* (see chapter 5, “The Visual Dispatcher”). As a consequence engines or trains can only be operated with stationary block decoders, if they are running under control of the *Dispatcher*. In return the *Dispatcher* guarantees, that traveling engines and trains are operated by the appropriate stationary block decoders. Because the *Dispatcher* is able to reserve blocks automatically according to the progress of traveling engines and trains under its control, it can also assign the appropriate stationary block decoders automatically to the engines.

Additional Options

In order to operate *stationary block decoders*, select the Option **Stationary Block Decoder** in the **Setup Digital Systems dialog box** (see Diagram 177).

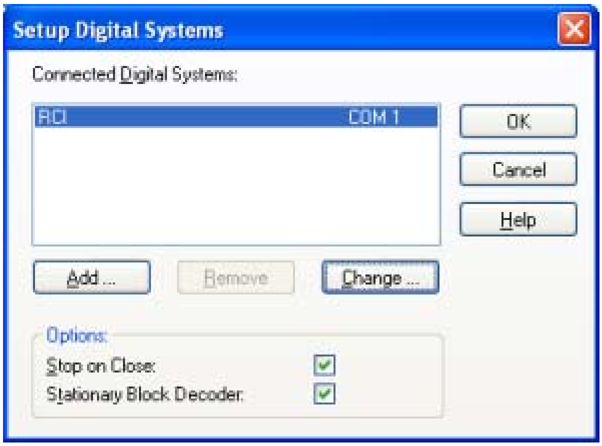


Diagram 177: Arranging the Digital Systems to use Stationary Block Decoders

When stationary block decoders are used, then in the **Block Dialog Box** an additional tab labeled **Connection** appears (see and). Here the digital addresses of the stationary decoders, which are associated with this block, are to be specified.

For each conventional engine select the Option **No Connection** in the tab labeled **Connection** of the **Engine Dialog Box.**

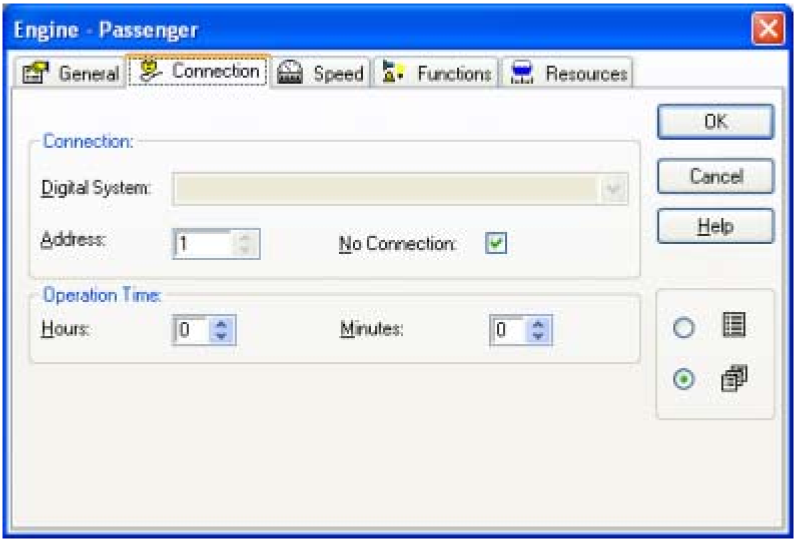


Diagram 178: Entering a Conventional Engine

When a train is assigned to a block which is setup for *Computer Cab Control* through the **Assign Train to Block dialog box**, then the additional option **Connect with stationary block decoder** is provided. Select this option, if the block should be connected to an available stationary block decoder during this assignment. In this case the stationary block decoder is reserved for this train. Until the block is released this decoder cannot be used by other trains. If this option is not selected then **TrainController**™ tries to reserve an appropriate stationary decoder, when the train starts running on a *schedule* or when additional blocks are reserved for this train.

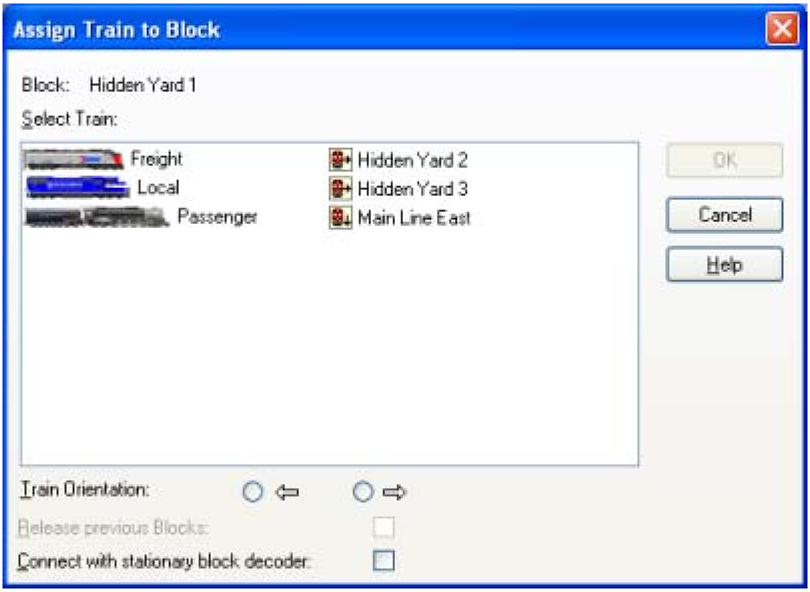


Diagram 179: Reserving a Block for a Conventional Engine

14.3 Operating Several Digital Systems Simultaneously

With **TrainController**™ it is possible to operate several digital systems in parallel. This is for example useful, if

- Your favorite digital system does not support monitoring of track sensors and feedback events.
- All digital addresses provided by your digital systems are already in use and you need more capacity to operate additional items.
- Your digital system is too slow for efficient monitoring of track sensors – especially in case of larger model railroad layouts.
- You want to use separate digital systems for engine and accessory operation.

TrainController™ supports the simultaneous operation of up to 12 digital systems. During operation it does not matter, to which system particular items are connected. **TrainController**™ handles all connected digital systems like one large system. All features can be used without any conditions as if only one large system were connected. It is for example not important if the switches contained in a certain *route* are connected to the same digital system or to different systems.

Only when the digital address of an engine, switch, track sensor, etc. is specified, then you have to take care, that the correct digital system is selected (see Diagram 56).

14.4 Migrating Data Files created with Version 4.x or earlier to Version 5

TrainController™ Version 5 introduces the new *Visual Dispatcher* as the main difference to previous versions of the software.

Several changes apply to existing data files that have been created with previous versions of **TrainController**™. The conversion is done automatically and usually everything will run as before after an existing data file has been loaded into Version 5 the first time.

Nevertheless to benefit from the new powerful features of the *Visual Dispatcher* completely it is recommended to adjust certain aspects in existing data manually.

The following hints describe the details of the automatic data conversion.

Blocks

In previous versions all *blocks* were stored in a plain list. The new *Visual Dispatcher* stores all blocks in the *main block diagram*, that does not only contain the blocks themselves, but also the links between the blocks and optional routes (see chapter 5, “The Visual Dispatcher”).

When a data file has been loaded, that was created with a previous version of **TrainController**™, then an initial main block diagram is created. This diagram contains only the existing blocks in an arbitrary order, but no routes and no links.

This initial block diagram can be used as a starting-point to create your own main block diagram for your layout.

Routes

With the exception that blocks are stripped, routes are not converted any further. The route symbols remain in the switchboard, where they have been located before.

This is the only exception of the rule, that a route, that is located in a switchboard, is a manual route and cannot be operated automatically with the *Visual Dispatcher*. Routes converted from previous versions are initially located in switchboards, can be operated by the Visual Dispatcher, though, as if they were automatic routes. In this sense they are kind of hybrid between manual and automatic routes.

Those routes which had been used in lines and schedules for automatic operation can be conveniently moved to the *main block diagram* by drag & drop after the automatic conversion. Optionally you can have the software create an on-off switch at the previous position of the route, after a route symbol has been dropped to the main block diagram. This on-off switch contains appropriate *operations* for manual operation of the route in the switchboard, if desired.

Lines and Schedules

!

Lines in the sense of previous versions are not provided anymore. Instead the data of each *schedule* is stored in a *schedule diagram*.

When an existing data file is being loaded, then the following conversion steps are performed automatically:

- Lines are converted to schedule diagrams.
- All sub lines of existing lines are removed. The blocks and routes contained in sub lines are directly added to the new schedule diagrams.
- For lines that are not referenced as sub lines by other lines or by schedules, extra schedule diagrams are created. This is done to avoid the loss of data contained in such lines. If these additional schedule diagrams are not needed, it is recommended to delete them manually, if desired.
- In previous versions sub lines were used to control, how blocks and routes are reserved during execution of schedules (“stages”). For compatibility reasons the existing reservation stages are automatically mapped to *critical sections* accordingly (see page 194 for further details).
- If routes that still contain blocks, are detected during conversion, then these blocks are automatically stripped from the routes and directly added to the affected schedule diagrams.
- Due to the lack of an existing block diagram all schedule diagrams are created as free-style diagrams (see the next section).
- The *compass card* of previous versions, with which it was possible to mark the former directions “yellow” and “blue” for each individual block, is no longer supported. **TrainController**™ now works with natural directions left, right, up and

down, which map the corresponding directions on your layout, if the main block diagram is drawn accordingly.

- The former “yellow” and “blue” direction settings are converted in the following way: each block in each converted schedule is drawn horizontally in the diagram and supports left-to-right and back as directions of travel. The former “yellow” is mapped to “right” and the former “blue” is mapped to “left”. This mapping will probably not match the actual directions of your layout. Thus the alignment of each block should be adjusted manually by turning the blocks accordingly in the diagram.
- *Restrictions* are renamed to *Conditions*.
- Existing *restrictions* of *lines* are automatically mapped to appropriate schedule specific *conditions* of blocks or routes. This is not possible under all circumstances, though. In cases where this conversion fails **TrainController™** tries to convert existing restrictions of main lines to conditions of the dependent schedules. If this is not possible, too, then a warning message is displayed in the message window and the restriction is ignored. If you specified restrictions for *lines* in previous versions of the software then it is strongly recommended to check the message window for warnings after the conversion of your data has been finished. If necessary the conditions of affected schedules must be adjusted manually.

Other Elements

!

- Existing *block indicator symbols* in the switchboard remain supported for compatibility reasons only. They are loaded from existing data files without changes, but it is not possible anymore to create new block indicator symbols. All references to block indicator symbols from the *conditions* or *triggers* of other elements are automatically mapped to the corresponding states of the associated blocks.
- Text and image elements that show something else then a plain text label or plain bitmap (such as a train name or image) in the switchboard remain supported for compatibility reasons only. They are loaded from existing data files without changes, but it is not possible anymore to create such elements from scratch nor to change their existing properties.
- The previous indicator based train tracking was based on explicitly specified references between indicators in switchboards. These references told from which indicator to which indicator train IDs were forwarded. This method is superseded by the new block based train tracking, which follows the links specified in the main block diagram (see section 5.5, “Train Detection and Train Tracking”). Indicator based train tracking is supported for compatibility reasons only. If an existing data file contains tracking references between indicators, then these references are loaded without changes. Train tracking based on these references works as before. But existing references cannot be extended nor it is possible to create new indicators for train tracking.

Free-style Schedule Diagrams

Due to the lack of an existing block diagram all schedule diagrams loaded from a former data file are created as so called *free-style diagrams* .

Such schedule diagrams do not depend on the structure of the main block diagram. This is shown in the following image:

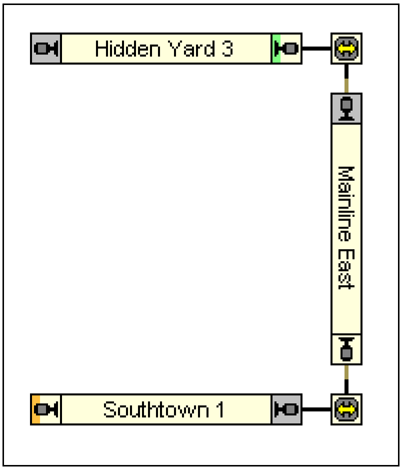


Diagram 180: Free-Style Schedule Diagram

From the operational point of view it does not matter, whether a schedule diagram is a free-style diagram or not.

- Unlike normal schedule diagrams a free-style schedule diagram is not based on the main block diagram (Diagram 180) and may contain links or routes, that are not contained in the main block diagram.
- The route symbols in a free-style schedule diagram initially refer to routes located in a switchboard. As an exception these routes can be operated by the *Visual Dispatcher* even though they are located in a switchboard like manual routes. If you move these routes later from their original switchboard position to the main block diagram then the references to these routes in dependent free-style schedule diagrams are updated accordingly.
- In free-style schedule diagrams blocks can be moved, but not resized or turned.
- If you completed the initial main block diagram and you want a free-style schedule diagram originally created during conversion of old data files to be derived from the new main block diagram, then you can call the **Use Main**

Block Diagram command of the **Schedule** menu whenever you like. If the free-style schedule diagram is compatible to the contents of the main block diagram then the necessary adjustment is then performed automatically. This conversion is irreversible.

- A free-style schedule diagram can be converted to be derived from the main block diagram, if it does <u>not</u> contain any extra links or routes that are not also contained in the main block diagram.

Suggestions for the subsequent manual Data Migration

!

TrainController™ leaves it up to you to decide, how much of your existing data you want to use any further. The software does not force you to start again from scratch.

The "quality" of the data automatically converted from a former version might not be optimal, though. Usually the structure of converted data will me more complex as necessary.

In previous versions the number of schedules, for example, needed to perform a certain task was much higher than the number of schedules required to do the same task with Version 5. Since the number of schedules is not reduced during conversion of existing layout data the converted data contains much more schedules than necessary. This increases the complexity unnecessarily.

The following suggestions are based on practical experience and recommended to be considered:

- Consider to create a block diagram of the entire layout first. Use the existing blocks as the starting point for this new diagram.
- Please note that the change of block alignment from vertical to horizontal or back, respectively (i.e. turning of blocks), is only possible in the main block diagram. If blocks are turned in the main block diagram then this change is reflected in all dependent schedule diagrams.
- Please note also, that the change of block alignment (turning of blocks) in the block diagram does not affect the direction of travel, to which blocks are passed in certain schedules.
- Please note that changing of the sides, where links are attached to blocks, also changes the direction of travel, to which the affected block is being passed. If you do such changes do <u>not forget</u> to adjust the reference direction of dependent brake and stop indicators, block signals and Virtual Contacts accordingly, if any.
- In order to create a suitable block diagram conveniently move all routes previously used in lines of the *Dispatcher* to an appropriate location in the *main block diagram*.

- After completion of your main block diagram consider to remove all existing train tracking references and to make use of the new superior block based train tracking.
- Consider to remove all indicators that are assigned to blocks from their original switchboard position by dragging them to the associated block in the *Visual Dispatcher*.
- Consider to replace block indicators and text and image elements that have been used to display train names or images in switchboards by traffic boxes.
- Consider to reduce the number of existing schedules by replacing the relatively simple free-style diagrams by more powerful and more complex schedules that are based on the main block diagram. Many old layout files contain 400 or more schedules. This number can be probably reduced to a number near 40 or 50. It might be worth to consider to create these 40 to 50 new schedules from scratch instead of editing the 400 existing schedules.

List of Examples

Automatic Car Lighting..**83**
Preventing an Indicator from Flickering..**166**
Automatic Reset of Signals **170**
Emergency Stop Button...**171**
Detecting Train Direction..**173**
Detecting uncoupled cars...**174**
Simple Track Occupancy Detection **176**
Automatic Engine Whistle..**177**
Manual Control of Station Entry **198**
Manual Control of Station Exit **200**
Hidden Yard with Train Length Control and automatic Bypass **201**
Turntable and Roundhouse...**220**
Indexing of an Analog Turntable..**231**

Index

acceleration 84
accessories 58
accessory element 59
address, digital
  of accessories 60
  of engines 69
  of signals 60
  of switches 55
advance signal 123
analog turntables 211
Auto-Detect 198
automatic Operation 92
automatic route 60
AutoTrain 142

bitmap
  in the switchboard 64
block 96
  and indicators 113, 114
  brake indicator 114
  current block 104
  locking a block 106
  locking the exit of a block 106
  main block diagram 98, 180
  occupied block 104
  preferred block 194
  release in a schedule 138
  reservation in a schedule 134
  reserved block 104
  states 103
  stop indicator 114
block diagram
  automatic calculation 98
block signal 122
  advance signal 123
  signal aspect 123
brake 71
brake compensation 79
brake indicator 114
  combined brake/stop indicator 121
bridge 55
bumper 55

clock 164
coal 85
combined brake/stop indicator 121
commands
  properties 167, 169
  properties of accessories 60
  properties of engines 69, 81
  properties of flagmen 172
  properties of routes 61, 63
  properties of signals 60
  properties of switches 56
  properties of turntables/transfer Tables 209
computer keyboard 66
condition
  and blocks 193
  within schedules 193
conditions
  and flagmen 173
  and schedules 197
  protection by 167
contact
  momentary 89
  occupancy sensor 89
  Virtual Contact 187
contact indicator 88
  momentary contact 89
  occupancy sensor 89
control panel 50
critical section 194

crossing 55
current block 104
curve 55
cyclic schedule 140

deceleration 84
decoder
  stationary block decoder 236
destination block of a schedule 128
destination key 63
diesel 85
diesel engine 85
digital address
  of accessories 60
  of engines 69
  of signals 60
  of switches 55
digital system 39
digital turntables 210
direction
  detect train direction 173
direction of travel 102
Dispatcher 92
display of train positions 66
distance, simulated 72
double slip switch 55
  solenoids 57

edit mode 45
electric engine 85
engine 69
  digital address 69
  orientation 102

feedback indicator 88
file
  layout file 44
  status file 44
flagman 171
four aspect signal 59
function only decoders 83

generic turntables 212

hidden yard 145
horse power 84
hot key 66

image 64
image element 64
indexing of turntables 209, 210
indicator
  and blocks 113
  and routes 198
  combined brake/stop indicator 121
  contact indicator 88
  feedback indicator 88
  memory 165
inertia 85
Inspector 147

keyboard 66

label
  in the control panel 64
layout file 44
light 59
link
  between blocks 101
locking block exit 106
locking of blocks 106

macro 177
  and timetable 204
main block diagram 98, 180
maintenance
  of engines 86
maintenance interval 86
manual route 60
maximum scale speed 71
memory of indicators 165
menus
  Edit 61, 63, 169

Edit 167
Edit, Switchboard 56, 60, 172
Edit, train window 69, 81
Edit, Turntable 209
message window 149
mode, of a schedule 141
momentary track contact 89
momentum 84
multiple units 81

node
in the block diagram 185

occupancy sensor 89
occupied block 104
odometer 72
oil 85
on-off switch 59
operations 168
and flagmen 172
by routes 62
in routes 170
system operations 169
optical sensor 88
orientation
of an engine 102

path selection in schedules 137
position
display of train positions 66
power 84
preferred block 194
protection of routes 62
push button 59

Railroad Clock 164
reed contact 88
Reference Indicator of Virtual Contacts 187
release of blocks and routes in a schedule 138
reservation of blocks and routes in a schedule 134
reserved block 104
restricted speed 139
reversing loop 105
route 60
and indicators 198
and protection 62
and signals 62
Auto-Detect 198
automatic route 60
between blocks 101
chaining 64
manual route 60
release in a schedule 138
reservation in a schedule 134
start and destination key 63

scale speed 72
scale speed, maximum 71
schedule 127
and timetable 204
critical section 194
cycle 140
mode 141
path selection 137
release of blocks and routes 138
reservation of blocks and routes 134
shunt 140
starting and destination block 128
successor 144
schedule diagram 127
schedule selection 145
selection
of a schedule 145
sensor
momentary track contact 89
occupancy sensor 89
shunt 140
shuttle train 140
signal 58
Signal 59

signal aspect
  block signal 123
simulated distance 72
single slip switch 55
slip switch 55
sound files
  engine function 80
  system operations 169
speed profile 72
speed, scale 72
speedometer 72
start and destination key 63
starting block of a schedule 128
stationary block decoder 236
status file 44
steam engine 85
stop indicator 114
  combined brake/stop indicator 121
straight 55
successor of a schedule 144
switch 55
switchboard 50
switchboard recorder 61
system operations 169

text element 64
text label 64
three aspect signal 59
threshold speed 71
throttle 71
time 164
timetable 204, 205
toggle switch 59
tonnage 84
track contact 88
track diagram control panel 50
track elements 55
track occupancy sensor 88
traffic box 98, 106
Traffic Control 146
train 69
train detection 106
train groups 197
Train List 67
train position
  display 66
train tracking 106, 111
Train Window 67
trigger, of a flagman 171
triple switch 55
turntable 55
Turntable Window 207
turntables
  analog turntables 211
  digital turntables 210
  generic turntables 212
  indexing 209, 210
two aspect signal 59
type, of an engine 85

uncoupler 59

Virtual Contact 187
virtual occupancy indication 192

wait time 139
water 85