iTrain

Manual

Version: 5.0.14
Date: September 27, 2021
Author: Xander Berkhout

Contents

Preface

After many years of development iTrain has reached version 5.0 that works on common operating systems. The user interface and the manual are available in some of the world's major languages. This combination of operating system and language support makes the program available to a wide audience.

It would cost too much time to create manuals and to keep them up to date for all operating systems beside all the translations. The differences between the operating systems in using iTrain are limited. This manual is written with the macOS (formerly known as OS X) user in mind so all screenshots are made on this platform and the key commands described are targeted on this platform. For Windows and Linux users the screen layout and dialogs will be more or less the same, but with other colors and styles that are common for their environment.

The most important difference between the operating systems is the use of the modifier keys. The 'Command' key on the Mac is not available on other systems and should be replaced by the 'Control' key. The 'Option' key on the Mac is also called 'Alt' and in this manual we use 'Alt', because it is the same on all platforms.

Popup menus or context sensitive menus can normally be accessed by clicking the right mouse button while the mouse is above a specific object. In addition, macOS also shows a popup menu when holding the 'Control' key and pressing the left or the single mouse button.

This manual introduces all the important aspects that you need to know in order to use iTrain. However, it doesn't explain general concepts of digital model railroads in much detail, so knowledge of this is a prerequisite. This has been done to keep the manual compact and readable. I recommend reading all chapters in order and at the same time run iTrain with the demo layout (`demo.tcdz` in the folder `iTrain/layouts`).

There is a dedicated support forum available (https://berros.eu/forum/) for iTrain users. It contains answers to questions you might have and if your question is not there you can ask your own. It is strongly advised to use the forum for support, instead of e-mail or phone, so questions can be answered to a broader audience and others can help in answering. A lot of enthusiastic iTrain users, including me, are active on the forum helping each other. See Appendix C for more information about the forum.

Kind regards,

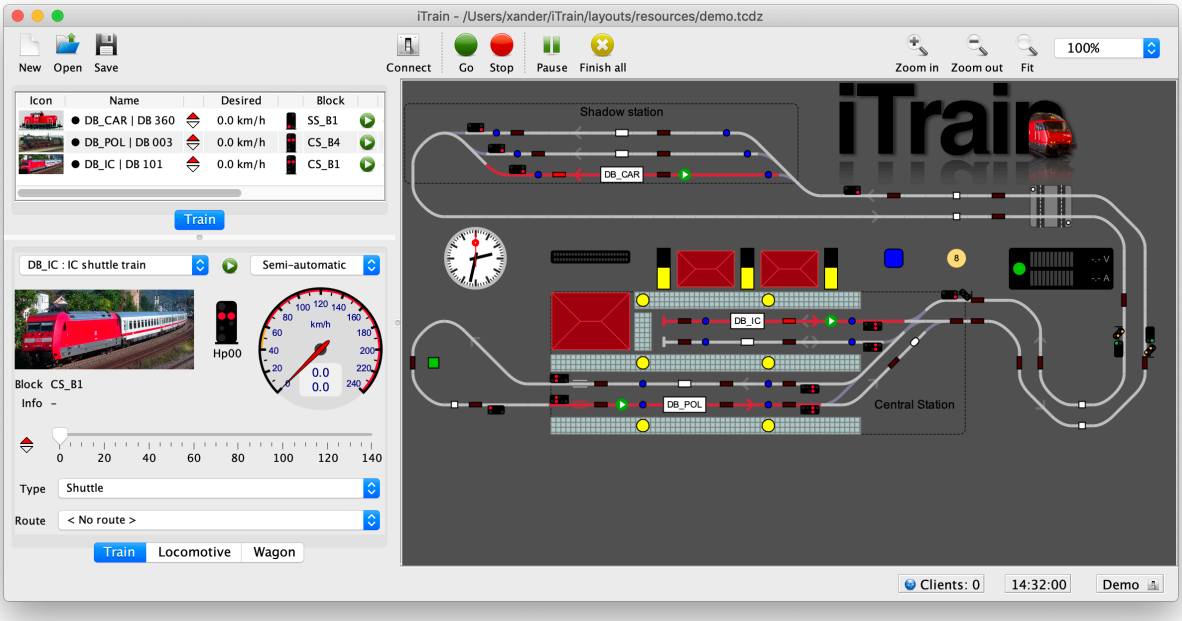
Xander Berkhout

Introduction

When you start working with iTrain for the first time you will need to know some basic concepts of the program.

The main window

An ordinary screen in iTrain may look like the following (see the `demo.tcdz` file):



In this screen we can identify three main areas:

1. The overview in the upper left corner gives an overview of the status of every train and gives immediate control over the main functions.
2. The control or throttle in the lower left corner gives more control over all the settings of a train and is meant to control it in detail.
3. The switchboard on the right side is a simplified drawing of your layout, in which you see where your trains are and what the state of all turnouts, signals and other objects is. You can also change the state of all objects.

Main objects

In iTrain we distinguish several types of objects:

- Interface - refers to the connection with a command station or other device to control the layout.
- Booster - refers to an intelligent booster (amplifier of track power) that can be controlled individually and reports its status individually.
- Feedback - refers to sensors in the track that detect if part of a track is occupied or not (see Appendix B).
- Accessory - switchable objects
  - Aspect - a general accessory with up to 32 states.

  - Turnout - is the switch to either join multiple tracks in one or split one track into multiple tracks.
  - Signal - may either refer to actual signals on the layout or only to virtual signals on the switchboard to indicate if a train should stop or is allowed to drive (optionally with restricted speed).
  - Decoupler - to decouple a loc from a train or to split a train into two trains.
  - Relay - is a switch with an on/off state (for example light) or a two state.
  - Light - represents a light source with a color and a level (on/off or dimmable).
  - Sound - represents a sound either as a decoder in the layout or on the computer itself.
  - Crossing - is a crossing between the railroad and a normal road or water.
  - Turntable - a circular device with track connections on the side and a bridge in the middle so that a loc can drive unto the bridge and be turned to exit at any direction.
  - Transfer table or traverser - a rectangular device with track connections on two sides and a bridge in the middle that can move sideways to connect with the tracks on one side or two opposite sides.
  - Train magazine - a device to shift multiple tracks in either horizontal or vertical direction.
- Track Route - is a list of turnout (and other accessory) states that create a path, for example from one block to a neighboring block, like in the Märklin Memory module.
- Locomotive (also called loc) - refers to the actual loc that you can control.
- Wagon - refers to all rolling stock on the track (including coaches) without a motor.
- Train - refers to a combination of locomotives and wagons as a whole
- Train type - refers to a category of trains with similar behavior
- Block - is a part of a track (not including turnouts) where only one train is allowed to be in.
- Train route - is the route or path a train can follow when driving automatically from block to block.
- Station - is a list of blocks grouped together so they can be used as a destination when driving automatically.
- Event action - also called just ‘action’ is a list of things that should happen when a condition has been met.
- Switchboard - is the presentation of the layout in which most of the objects can be switched or viewed.

Naming

In iTrain, all locs, turnouts, signals and all other objects have a name that is unique per object type. You don’t have to remember all the addresses, because you can give every object a useful name. It is recommended keeping the names short and using a naming convention that you understand. Every object can also have a description to give a more descriptive name to your own wishes, but the description may be left empty as well.

For example, for the name of a loc you can use the operator name like NS, DB or SBB followed by a space and the number of the loc, for example DB 101. If you have multiple locs of the same type, use a different suffix, for example DB 101-2 or the road number DB 101-104-8.

For the names of blocks and accessories you can use logical numbers prefixed by an identification of the station and separated by underscores. You could name the two sides of a block A and B to distinguish feedbacks and signals on the side. For example, CS_S4A is the Signal of the Central Station of track/block 4 on the A-side. All references to this signal in the definition of your layout will show this name. It is difficult to name turnouts, because they are the connection between blocks. You could prefix them with the station and just number them in order, like CS_W5.

Note: It is recommended naming your objects independently of the actual address of the object and choosing a more logical name based on the location in the layout. Using the address in the name violates this principle. The address will be displayed between parentheses behind the name. In many cases you may need it (such as with selection boxes).

File handling

After you have started iTrain for the first time, no file has been associated with your project so you have to save your project with a specific name. It is recommended using the extension `.tcdz` for iTrain project files[1] (tcdz =Train Control Data Zipped), for example `layout.tcdz`, and saving them in the folder `iTrain/layouts` in your 'Base path' (see Preferences).

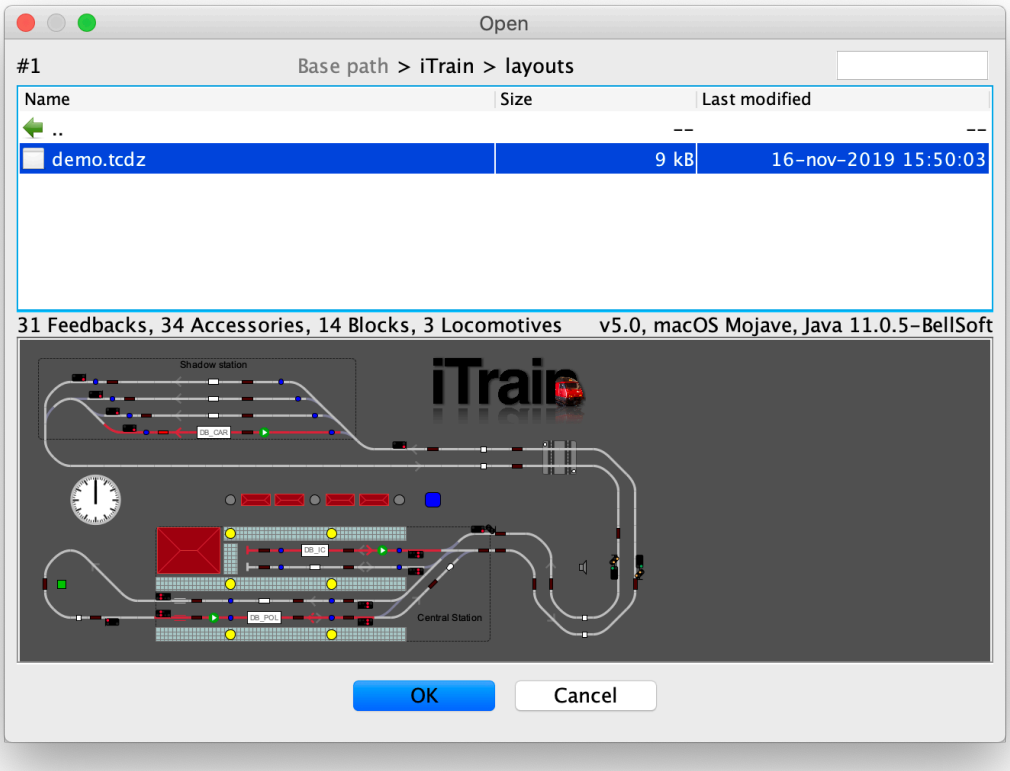
The next time you start the program, it will automatically try to load the same project. The project name will be shown in the title bar of the main window together with the full path to the file and the iTrain version in which it was created.

When closing the program the project will be saved automatically, unless you have chosen to do that manually. This will not only save the definitions of all objects, but also the current state of the layout (where are the trains and what was the last state of all accessories).

Resource browser
To open or save a file, you use the resource browser to browse through the file system. This is done in an operating system independent way covering only a part of the file system below, the so called 'base path'.[2] This is the area in which iTrain is allowed to look for resources, such as project, image or sound files, and it can be set in the Preferences (see the next chapter).

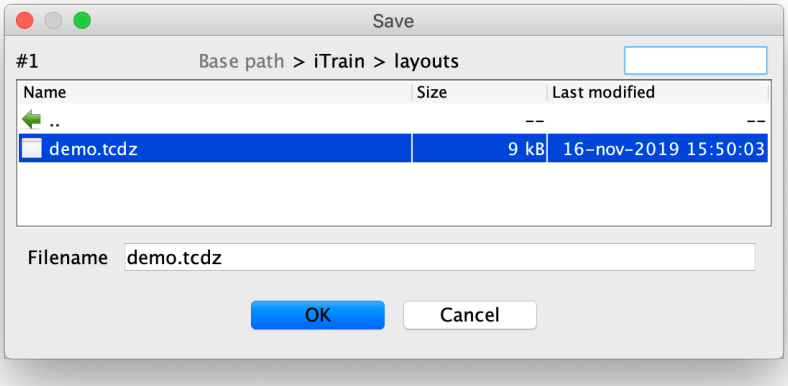
[1] The previous extension `.tcd` and `.tcd.zip` are still allowed, but it is recommended to use the new extension. This new standard has some advantages: It is smaller, it can be used on the forum and it does not unzip unexpectedly.

[2] When using a network client, the files shown in the resource browser are on the server in its base path.



On top the relative path in relation to the ‘Base path’ is shown. You can go directly to a folder by double clicking it. The green arrow to the left followed by .. means the parent directory. You can also use the ‘Backspace’ key to go up in the hierarchy directly. The search field in the upper right corner allows you to narrow down the list of files by typing specific characters that match the file name.

Selecting a file name will show a preview of the project with some info at the bottom. Double clicking the file name will choose the file and leave the resource browser (as if ‘OK’ was clicked).



When you save a project, the preview at the bottom is replaced by a text field to enter the file name.

Backup

Before saving a file, the previous file (with the same name) will be moved to a `backup` subdirectory and the date and time will be added to the file name. This way you always have a backup of the previous states of your layout in case something goes wrong. The name of the backup files have the following format that includes the original name followed

by the word `backup`, the date and time of creation of the backup and the version of iTrain with which it has been saved.

`layout_backup-20200526-105800_v5.0.4.tcdz`

To restore a backup, just open the file from the `backup` subdirectory in iTrain and the original file name and path will be restored. So it is not necessary to copy and rename files from the backup directory. The backup directory may fill up fast with every save, but the files are generally small compared to the available disk space. You could clean them up if you want, but in general this is not necessary.

Recent files
The menu item 'Open recent files' shows a list of up to 12 files that have been opened recently. You can go back to an earlier opened file quickly by selecting it from the list, but do save the current file first if you have made modifications that you want to store.

Import
The import allows you to open a file as well. But it will import only a certain part of it: locomotives and wagons. The import will add this part to your already loaded file. The import dialog contains two tabs, showing all the locomotives and wagons which can be imported. You can now select the objects you want to import and also indicate whether they should be imported actively or inactively. Possible conflicts with existing objects are displayed in the last column.

Export
There are two different exports. The tabular export allows you to export the most important objects (locomotives, wagons, accessories, feedbacks and blocks) as different `.csv` files (tab separated, encoding UTF-8) that can be read immediately by a spreadsheet program. In the export these `.csv` files will be bundled as a zip-file. Because every object can have only one line with columns for their properties, only the single value properties of an object will be exported and not the complex properties that are lists or tables in themselves.

Note: The tabular export files are only for analysis or bookkeeping and cannot be imported by iTrain.

Secondly, there is an 'External inclusive' export. This is a normal save, but now all the images that are referenced externally will be included (with extension `.extern.tcdz`). This export is only meant to share your layout with someone doing support for you in case it is important or comfortable to see the external images. You can open this export as any normal layout file with 'Open'. After saving, the images are removed from the file, but the original references to the images are still there. This is not a problem as the owner of the file has the original images.

Note: This export format could be used on the forum, but normally it should not be used. Because of the images included, the file will be much larger than normal `.tcdz` files. So only do this when there is an explicit request for it by the helper.

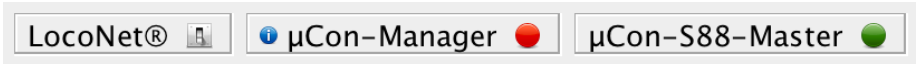
Print
The print option only prints the current switchboard on paper. To print a list of objects, you have to use the export function, import it into a spreadsheet and then print it.

Online

When you are working with the program sometimes you only want to change your configuration and at other times you are really controlling the trains on your layout. In the

second case you will need a connection to the control device and when this is done the program is ‘online’.

With the  button on the toolbar you can connect to all the defined control devices. The status of the individual devices will be shown on the toolbar, in the right lower corner of the main window.



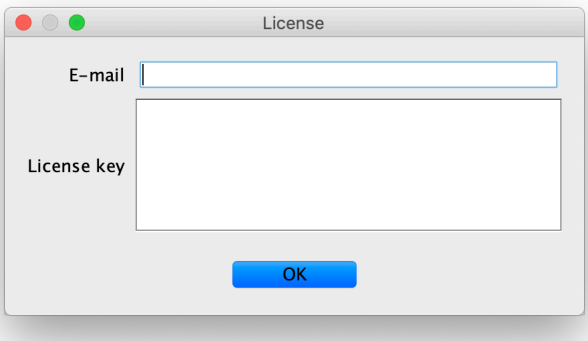
If the device still shows  it means it is offline and you can also double click on it to connect to just one device. In case it is online it will show a colored ball. The color indicates the mode of the device. Green means that the device is ready and there is track power (if that applies). Red means the track power is switched off. Yellow means the initialization did not succeed. Blue means programming mode for some devices.

Note: It is always recommended selecting ‘Disconnect’ or exit iTrain before turning off your control device so that it can disconnect correctly. If you accidentally turn it off before going offline then check if the program did detect it or else do it manually instead.

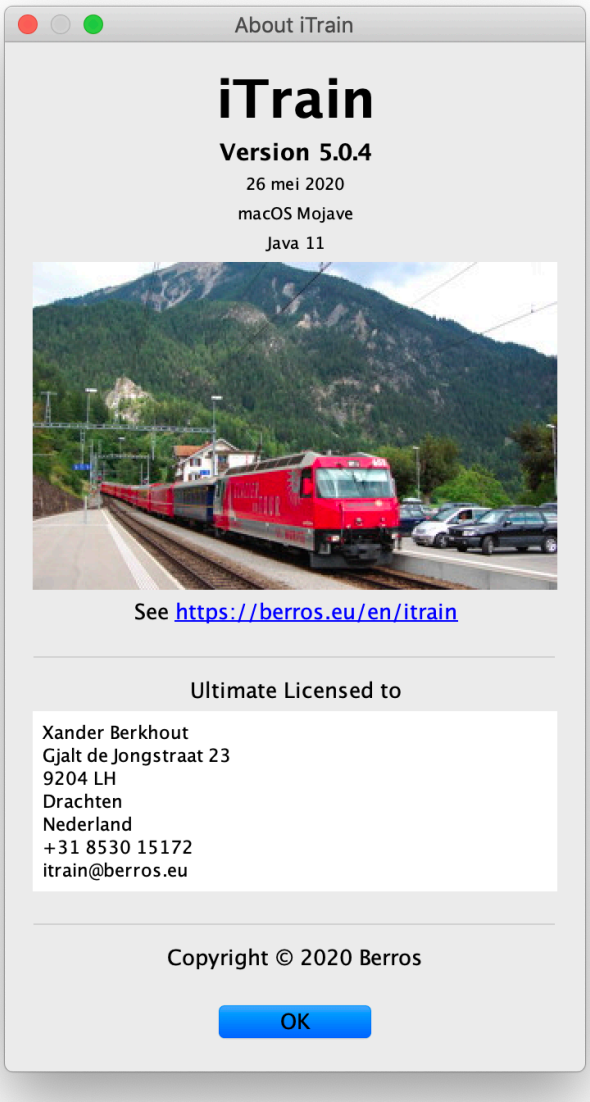
License

iTrain is a commercial product and runs by default in demo mode. You are limited to 3 locomotives and 32 accessories and 32 feedbacks. This allows you to test iTrain on a small demo layout like the one supplied (`demo.tcdz`).

To make use of all features available in iTrain, you have to request a trial license or buy a license on the website to receive a license. This license is linked to your e-mail address. To enter your license number go to main menu ‘Options’ -> ‘Enter license key’.



Type or copy your e-mail address that you used to get a license and paste your license key into the ‘License Key’ box by using ‘Command’ key + ‘V’ or the popup menu. It is not recommended typing it manually.



If the license key is valid you will see the about box with your license type, name, contact information, e-mail and optionally the end date of the license.

Note: To remove the iTrain license from your computer, you only enter the e-mail address without entering a license key. Running the "Deinstaller" only removes the program files and does not remove the license.

Project

Now you are ready to enter your model railroad into iTrain. This is called the project. All the project data can be entered via the different items in the 'Edit' menu and will be saved in the project file (with the extension `.tcdz`) except for the preferences that will be stored somewhere on your computer and will remain the same even if you change the project.

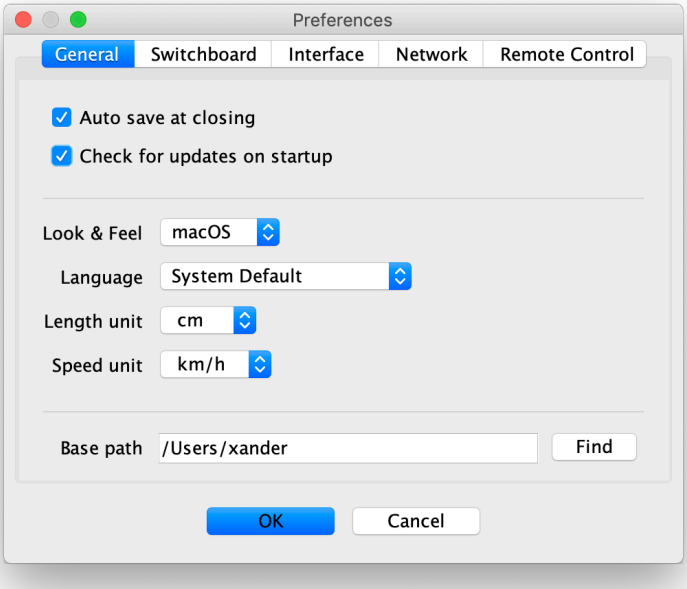
First check the preferences and the settings and choose the interface before you continue with entering your layout. The next step could be entering locomotives to test your connection, but this can be deferred to a later stage as well. A good strategy is to first draw the whole layout, then assign all the turnouts, signals, feedback and blocks to the drawing and finish with the locomotives, wagons, trains and routes.

Preferences

The preferences are settings that apply to all projects and are saved on the computer itself in an application specific location. You can find them in the iTrain menu on macOS or at the bottom of the 'Edit' menu on other operating systems.

General

In the 'General' tab you can change some specific settings of the behavior of iTrain. It is possible to save the project when closing iTrain without asking for confirmation. By default a check is done if a newer version of iTrain is available at startup. If your computer is never connected to a network you can turn this off.



Look & Feel
Every operating system has its own look & feel and sometimes you can even choose themes. iTrain by default follows the look & feel of the operating system, but it is possible to choose another one for consistency among computers. The recommended cross platform look & feel is 'Nimbus' that is now used by default on Linux. Changes to the look & feel will not be active immediately, but only after a restart of iTrain.

Language
The iTrain user interface is available in multiple languages. By default it will choose the same language as the operating system and this is called 'System Default', but you can change it to a language you prefer. The changes will not be active directly, but only after restarting iTrain, because the user interface has to be reloaded.

Length unit
The length unit specifies what kind of measure is used throughout iTrain for displaying length. It is recommended to use centimeters, because that will fit naturally with the scale of most models and is also used internally. Millimeters, meters, inches and feet are also selectable options. Length values are floating point values so you can have 51.5 cm or 0.515 m. In every length input box in the program you can always enter a value in another unit by adding the unit to the value. It is automatically converted to the unit specified in this preference after hitting the ENTER key. The possible suffixes for length units are 'mm',

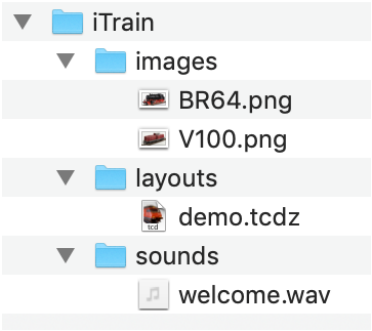
'cm', 'dm', 'm', 'in' and 'ft'. When you do not specify a unit in a length input box, the default specified in the preferences is assumed.

Speed unit

The speed unit specifies what kind of measure is used for displaying speed corrected for scale throughout the application. The natural unit would be km/h or mph, but also m/s is possible. Speed unit values are also floating point values. In every speed input box in the program you can always enter a value in another unit by adding the unit to the value. It is automatically converted to the unit specified in this preference after hitting the ENTER key. The possible suffixes for speed units are 'cm/s', 'm/s', 'km/h' and 'mph'.

Base path

All resources - layout, image and sound files - are stored together in a folder structure with on top the folder `iTrain`. This folder is normally stored in your user folder (in the example called `/Users/xander`) and all references to resources in your project are relative to this folder. This makes it easier to migrate a project to another computer.

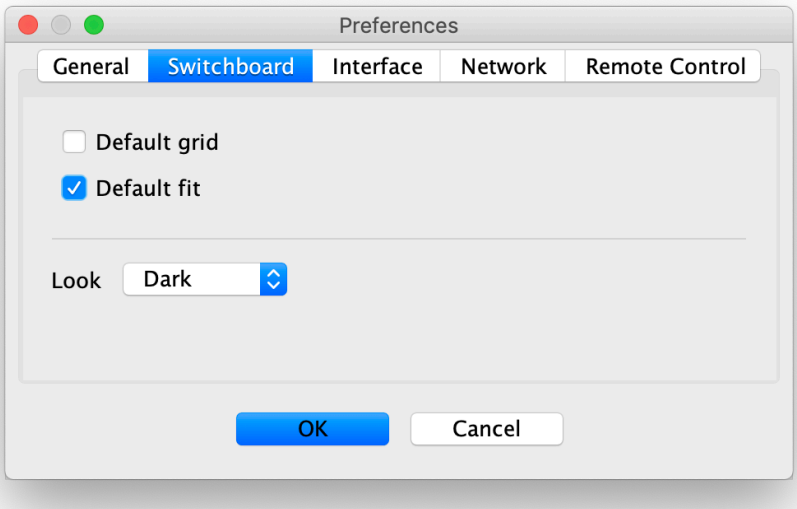


If you want to use another location (or another drive) to store all iTrain related files, you can change this folder. This is called the base path. This base path may contain the folder `iTrain` at the end or it may be its parent folder. Your choice will affect the relative paths stored in the layout file and which files you can access from the resource browser.

Tip: If you want to restrict the resource browser to files below the folder `iTrain`*, then put the name* `iTrain` *in the base path. If you want to be able to also browse folders next to the folder* `iTrain`*, then leave it out of the base path.*

Switchboard

The 'Switchboard' tab has some generic settings for the switchboard that apply to all projects. The option 'Default grid' decides if a new switchboard or the switchboards in a newly opened project will be shown with a grid. It will not affect current switchboards, because you configure it for the existing switchboard via the popup menu. To automatically use the maximum allowed zoom that does not need scrolling for newly opened projects, select 'Default fit'.



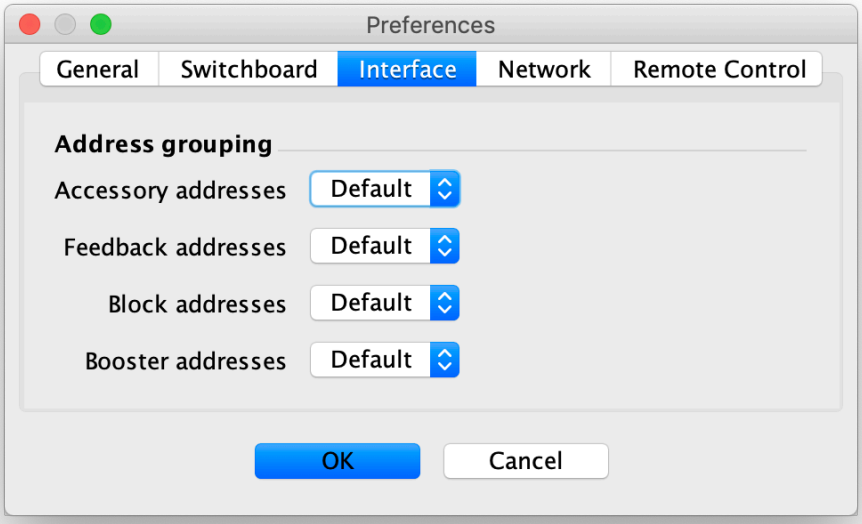
The look or appearance is a set of colors, line widths and text size used to draw the layout. There are three looks available:

- Classic - a white background and drawn as in all previous versions
- Contrast - with a light gray background that is easier for the eyes and makes the colors better identifiable, somewhat bigger font for better readability
- Dark[3] - using a dark background and lighter track lines so colors of feedbacks and signals have better contrast

Interface

On the 'Interface' tab you find some interface related preferences.

Address grouping



Grouping is another way of displaying the address of an accessory, feedback or block with two numbers and a dot in the middle (<module>.<output> format). In iTrain every object has its own absolute number, but in many command systems addresses are written with group or module numbers and sub numbers.

[3] We recommend using the Dark mode as this easier on the eyes and will save energy on OLED-displays.

A few examples:

- the address **1** is written as **1.1** when used with grouping.
- the address **5** is written as **2.1** when used with grouping or module size **4**.
- the address **23** is written as **2.7** when used with grouping or module size of **16**.

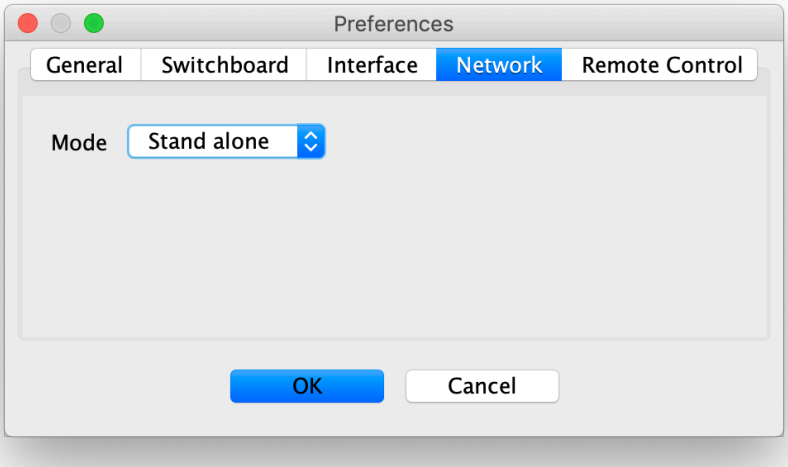
The general formula is: **address** = (**module number** - 1) * **group size** + **output number**

A grouping of 16 is very common with S88 modules where the first number is the module and the second the output on the module. For accessories, 4 can be used when using decoders with four outputs (for example k83) where the first number is the number of the decoder. In Selectrix based systems the grouping is always 8.

In the grouping preferences you can specify how addresses in iTrain are displayed for accessories, feedbacks and blocks. Internally they are stored as absolute addresses. A grouping of zero means no grouping and displays absolute addresses. When entering addresses you can always enter either an absolute address or a grouped address. In case grouping is used the absolute address is shown in the tooltip of the address field.

For every interface another type of grouping (or no grouping) is more or less common. By using 'Default' for grouping, the default for every used interface is used which results, for example, in 8 for Selectrix and 16 for feedbacks in S88 based systems.[4]

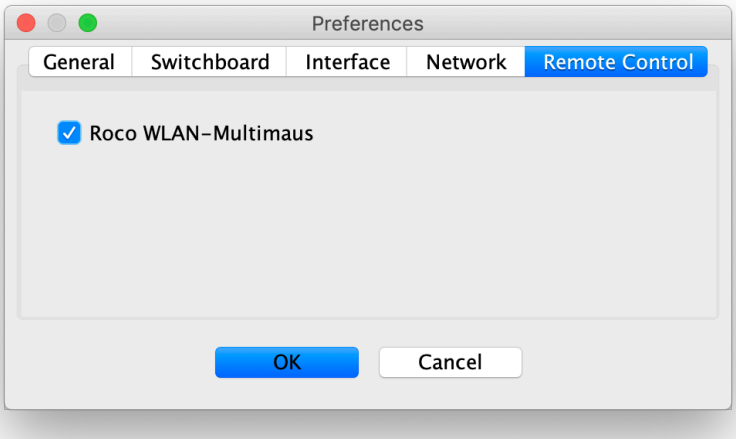
Network



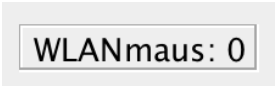
The fourth tab in the preferences is about using iTrain in a network. This is an advanced topic and it will be described at the end of this document when other concepts have also been explained.

[4] In case of default S88 grouping, the format is extended with the bus as prefix for addresses above 1000 (so 1.1.1). See Appendix D for more information where this applies.

Remote Control



The last tab ‘Remote Control’ contains the configuration of a remote control.[5]



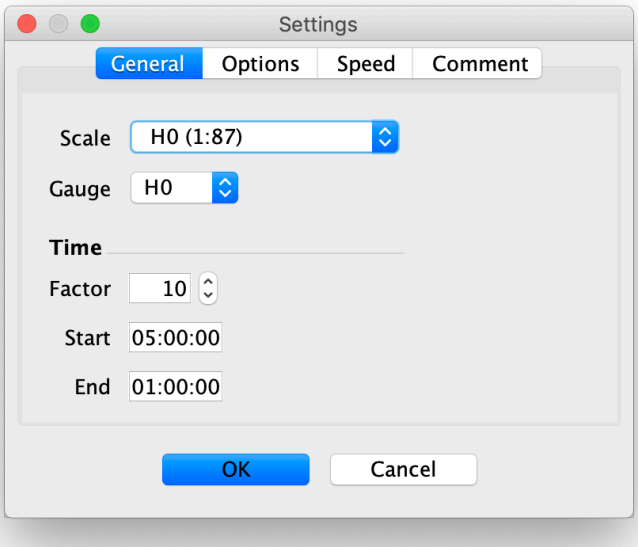
There is support for one or more Roco WLAN-Maus devices that connect directly to iTrain without needing a Z21. Just checking the box will make iTrain listen to incoming requests. In the configuration menu of the WLAN-Maus you have to replace the IP-Address of the Z21 with the IP-Address of the computer running iTrain after you have joined your WiFi-network. See the WLAN-Maus manual for more details.DIR

[5] Support for the Dinasys Infrared Remote (DIR) has been removed, as announced before.

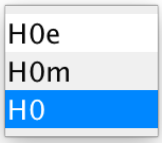
Settings

The project specific settings are saved in the configuration file and will be the same on every computer using this project. You can configure the settings via menu ‘Edit’ -> ‘Settings’. There are three tabs with settings.

General

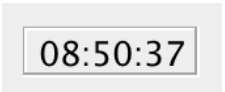


The most important setting is the track scale that is used for speed and distance calculations. By default it is H0, because it is the most widely used scale, but if you use something else please specify it here. The second setting is the default ‘Gauge’ or the width of the track used in most cases. The choices are displayed by using the track scale and adding the letter for narrow gauge, or nothing for a normal track.



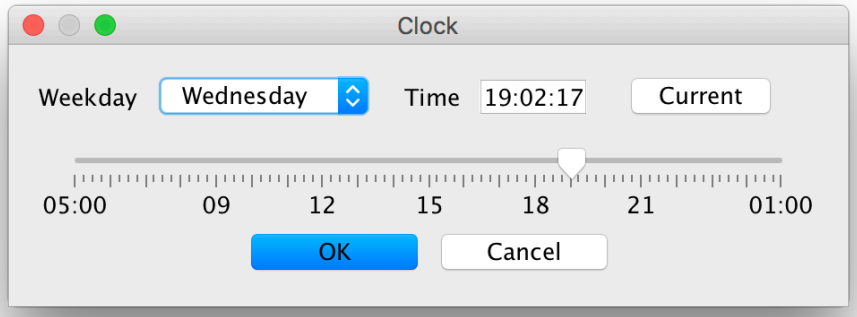
Time

In iTrain there is a general model time clock. This clock can run faster than normal time to simulate a day in less time. This is called the ‘Factor’. The start and end times can be specified to limit the hours at night. In the example the day is limited to 20 hours and with a factor of 10 that means it will take 2 hours to simulate a whole day.



The current model time will always be displayed on the toolbar at the bottom of the main window in a digital format.[6]

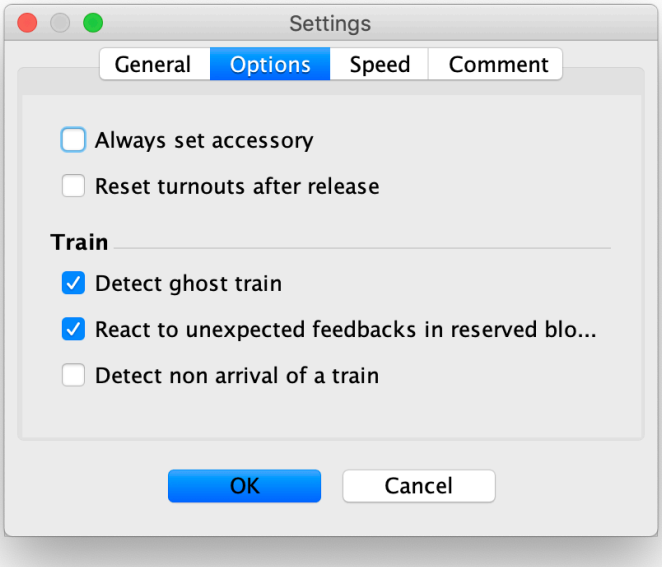
[6] An analog model clock can be put on the switchboard. This is described in the chapter ‘Switchboard’.



You can change the model time and weekday by double clicking the time on the toolbar. A dialog will appear with a slider to roughly set the time between the start and end time. You can also enter the time directly or press the button 'Current' to get the current time of the computer.

The weekday is optional and can be used with actions to let them run only on specific weekdays.

Options



Two general options can be specified:

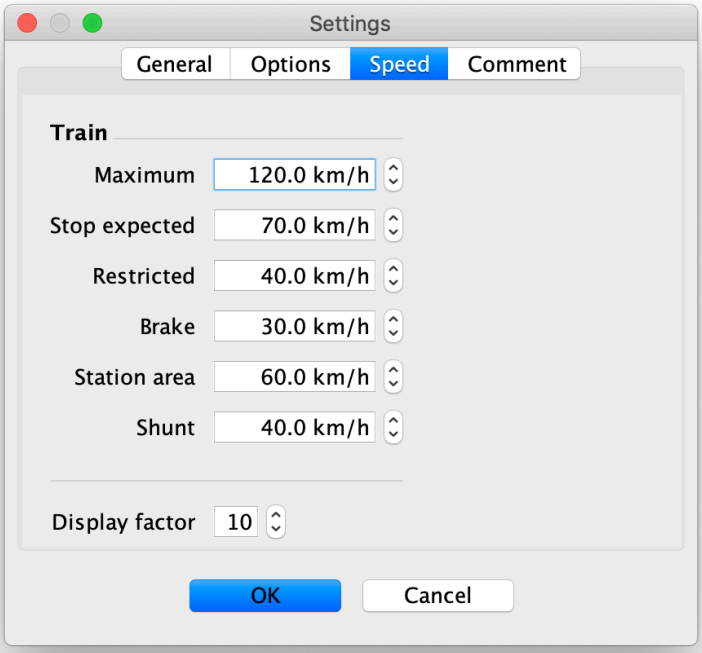
- Always set accessory - an accessory will only be set/activated the first time it is used, or afterwards only when iTrain thinks it does not have the correct state. In case manual changes on the layout are made later on, you can use this option to always set/activate the accessory. This guarantees the correct state on the layout at the cost of extra switching time.
- Reset turnouts after release - a turnout will normally be left in the last state that was activated. To always return the turnout to its initial state after it has been released in automatic control, you can select this option.

Train

As iTrain is now also able to control cars with an additional iCar-license, the train specific options are specified in a separate section. These options are described here, but require more knowledge about routing, reservations, etc. described later.

- Detect ghost train - to detect a train that enters a wrong block, because of a turnout that is not switched correctly.
- React to unexpected feedback in reserved blocks - reserved blocks should only be entered from the expected side and no other feedbacks in the block should be activated suddenly. In case this happens automatic driving of a train will be stopped immediately, because something else probably entered the reserved block.
- Detect non arrival of trains - trains that will arrive too late in the next block will be stopped and an alarm signal will be given. Based on length of the block and the turnouts between them, iTrain can make an estimation when a train should have arrived in the next block.

Speed



The speeds on the layout will be restricted like in the real world. Here you can specify the train speed limits for different categories that will be the default values for the whole layout. Of course, it is possible to specify individual speed limits per block, but in case you do not enter anything the following default speed limit values will be chosen.

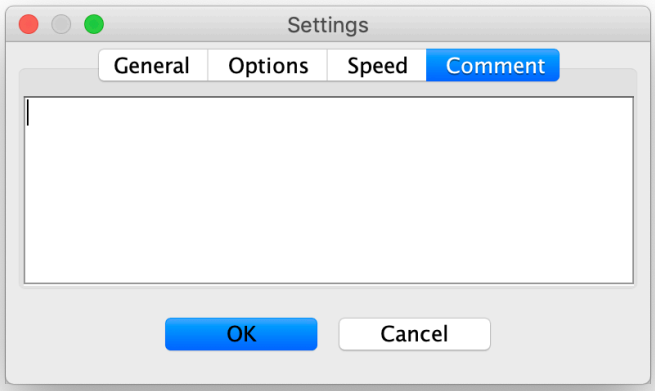
The different categories are:

- Maximum - the maximum speed in the block if there are no other restrictions from signals.
- Stop expected - the maximum speed when leaving the block if the train has to stop in the next block.
- Restricted - the maximum speed if the signal shows the restricted sign, for example when a turnout is branched in the path to the next block.

- Brake - the speed to which a train brakes in case of a red signal until it enters the stop feedback.
- Station area - the maximum speed for blocks of type ‘Station’ or ‘Shunt[7]’.
- Shunt - the maximum speed for shunting operations and for blocks of type ‘Siding’, ‘Turntable’, ‘Transfer table’ or ‘Train magazine’.

The display factor is the multiplication factor for the number displayed on a signal to become a speed in km/h. Normally this will be 10, but if the numbers represent mph you should use a value of 16.

Comment



The last section ‘Comment’ allows you to make general notes about this project.

[7] Blocks of type ‘Shunt’ are normally part of the station area and will be passed by trains with station speed.

Interfaces

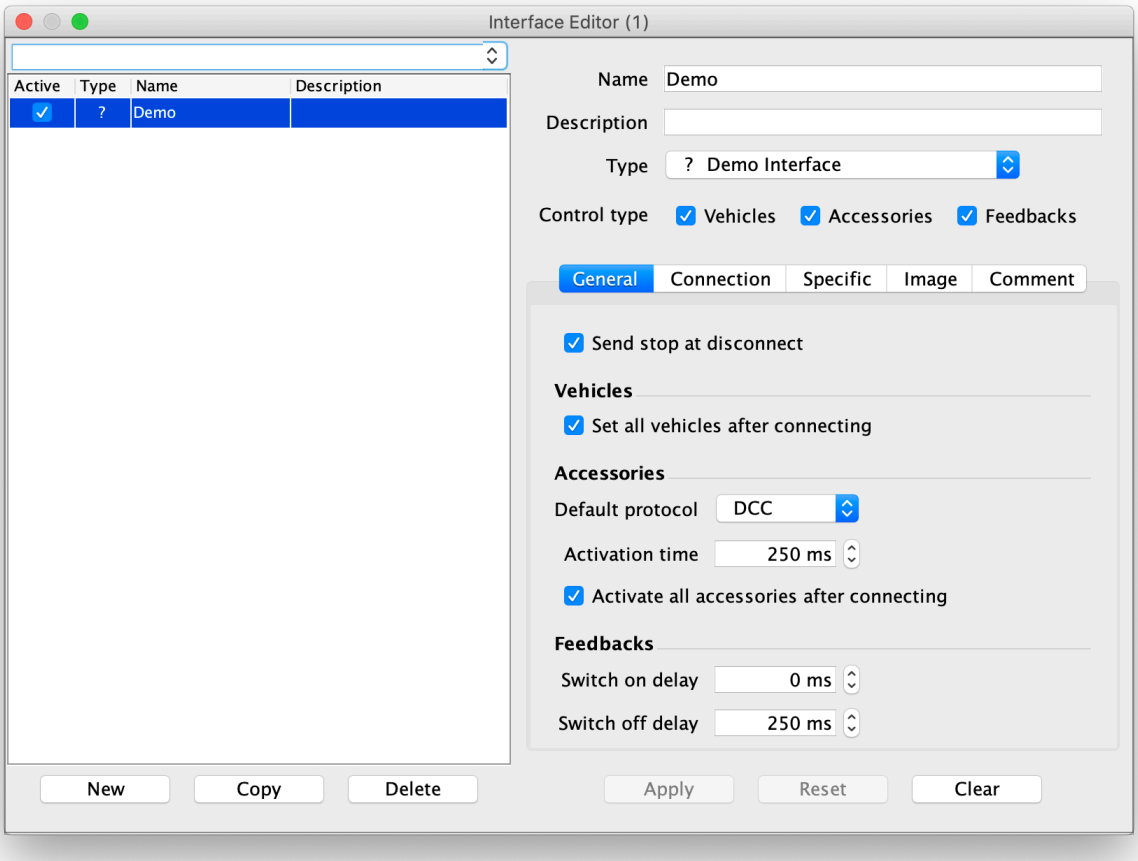
The software normally communicates via a serial, USB or ethernet cable with a command station directly or via a separate device that is connected to the command station. The device connected to the computer is called ‘Interface’ and several types of interfaces from different suppliers are supported. Every interface has limitations in the type of decoders supported and the number of decoder speed steps and functions allowed. Therefore it is necessary to define the interfaces before continuing with other definitions.

iTrain has been designed to work with multiple interfaces at the same time. There are several cases in which multiple interfaces can be useful. A few are listed below, they may be combined if required:

- You want to read the feedbacks in via a separate device to get better results. In that case you can use an HSI-S88, µCon-S88-Master or LoDi-S88-Commander as a second interface and disable the ‘Feedbacks’ in your first interface.
- You want to control signals via a separate device (OC32 or LoDi-Shift-Commander) that directly controls the lights bulbs.
- You want to switch all your accessories via an old command station so that you have all the power of your new command station for running your vehicles.
- You are controlling two layouts or two different systems (for example 2-rail and 3-rail) at the same time and you want to manage them all in one application.
- You have a separate booster management system like the µCon-Manager or LoDi-Rektor.
- You want to use an extra interface with handhelds/throttles to control vehicles or accessories on another interface.

Depending on your license you are allowed to work with one or more interfaces at the same time.[8]

[8] In case you need more interfaces, you can buy an extension of your license via the website.



To get into the ‘Interface editor’, you go to the menu ‘Edit’ -> ‘Interfaces’ or you press Command + F6. On the left side is the list of defined interfaces iTrain. In the ‘Active’ column you can indicate which interfaces are currently attached to the layout.

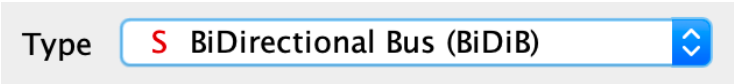
To manage the list of interfaces you use the buttons below the list. You can add a new interface by using the ‘New’ or ‘Copy button. In case of ‘Copy’ the currently selected interface is used as a template and its definitions are copied to the new interface. Only the name is adapted to create a unique name. The ‘Delete’ button removes the currently selected interface from the list.

On the right side you find all definitions in the editor. The buttons below the editor give some extra control over the editor:

- The ‘Apply’ button applies changes you made in the input fields. This will immediately be reflected in all windows. If you select another interface the previously selected interface will automatically be applied.
- The ‘Reset’ button discards the changes you made in the editor and reloads the fields with the current value. After an ‘Apply’ a reset will only discard changes made after the ‘Apply’.
- The ‘Clear’ button clears all the fields.

Tip: This layout, with on the left side a list and on the right side an editor and the same buttons, is used for most objects in iTrain.

Always first fill in the name of the interface. A name is needed to identify any object in iTrain. This name should be unique and it should not be too long. There is always an additional, but optional, field ‘Description’ to include more details. This field can be used as a long name.



In the box 'Type' you can select the type of interface. This is a rather long list with names. A type name often refers to a specific device, but it can also refer to an interface protocol[9] used by many devices (even from different vendors).

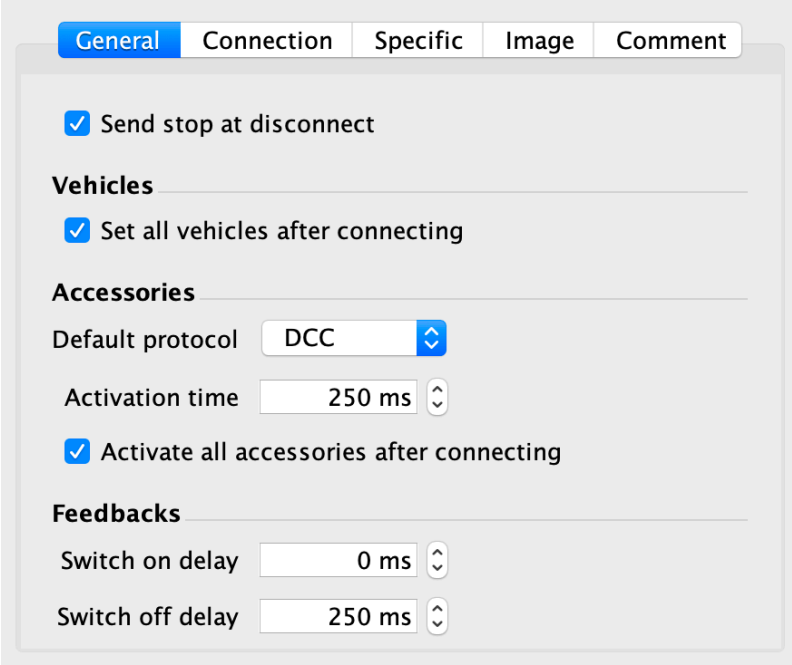
By default an interface will control the vehicles, accessories and boosters, and read the feedbacks in iTrain, but it is also possible to restrict the control of the interface by changing the checks for 'Control type'.

Some interfaces are only made to control accessories and/or read feedbacks. This is called a partial interface, and they have a lower case character in front. In a partial interface some 'Control types' will be invisible.

The other interface settings are divided over multiple tabs called 'General', 'Connection', 'Specific', 'Image' and 'Comment'.

General

On the 'General' tab you can set some properties that involve almost all interfaces and are about vehicles, accessories and feedbacks.



Normally a stop signal will be sent to the tracks before the interface disconnects. You can prevent this by unchecking the box in front of 'Send stop at disconnect' in case the layout is also operated at the same time by other users and you don't want to remove the track power.

[9] Protocols used by many vendors are XpressNet, LocoNet® and Selectrix.

Vehicles
At connection time iTrain tries to synchronize the vehicles (locs, wagons, etc.) by reading their status from the interface or, if this is not possible, by sending commands to all vehicles. You can force to always send the commands to all vehicles by checking the box 'Set all vehicles after connecting', but it will only be enabled if reading the statuses from the interface is also possible.

Accessories
The default accessory protocol is the protocol used by default when creating an accessory. In case your command station also supports a default value for accessory protocols (for example the ECoS), it should be set identical.

The 'Activation time' is the default time between the activation of an accessory and its release. This can be overridden per accessory. Not every interface supports this default 'Switch delay', but in case the interface will not use it then it is also the time iTrain will wait until the next accessory is activated. This prevents overloading of the input buffer of the command station.

By default all accessories are activated when going 'Online' so that the state on your layout and in iTrain will be identical. This takes some time and you can switch it off if you are sure that you did not change accessories manually or with the control device when iTrain was not running. If you switch this option off, iTrain will try to read the state from the command station if the interface supports this.

Feedbacks
Feedbacks are the eyes of the program and it is important that they deliver good results. In some cases it is necessary to filter the raw input to remove short spikes, because of bad contact between the wheels and the track. Some higher priced feedbacks modules have this filtering built-in. In this case it is recommended to use the hardware filtering, because it does not cost processing time and is probably more accurate. If it is not available you can use the software filtering.

The 'Switch on delay' is the time a feedback needs to be on before the feedback sees it as on. The same holds for 'Switch off delay', so an off is only seen as off after some time without an on. In general it is important to keep the 'Switch on delay' short, because most actions are based on a feedback going on and it might delay the action. In most cases you can leave it to zero, but if you use it then use a small value (< 50 ms).

The 'Switch off delay' is less critical because a feedback going off is mostly used to notify that a a block is not occupied anymore. Short contact loss should not result in a feedback going off, so a short delay is preferred to an unwanted release. Common values for 'Switch off' are in the range 100-500 ms.

This feedback filtering is the default setting for all feedbacks attached to this interface. It can be overridden for any individual feedback. This is useful if you combine feedback modules with different characteristics.

Note: Filtering the feedback output also results in fewer transitions from on to off and back again and will improve the overall performance of the application, because actions and checks will not be executed unnecessarily.

Connection

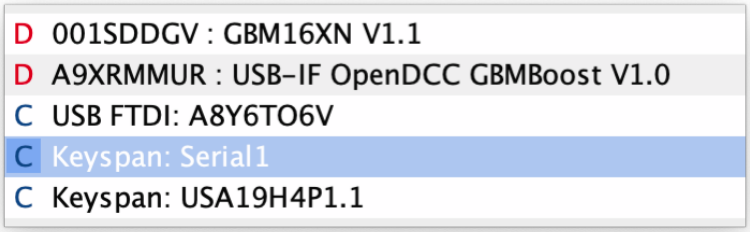
Depending on the interface type, the 'Connection tab' will show different connection settings for a serial, network or file interface.

Serial interface

All interfaces in the box 'Type' with a red S or s have a serial interface. A serial interface always has a direct connection with the computer via a USB (Universal Serial Bus) connector or a serial (COM) port with a 9 or 25 pins connector (the so called RS232). On most new computers there is no serial port available and you will need a USB to serial converter[10] to be able to connect your computer to an interface with a serial port. In case the interface has a USB connector this is generally a built-in USB to serial adapter so you don't have to buy one. In all cases when you use USB you have to install VCP (Virtual Serial Port) drivers[11] before you can use it.

On Linux systems these drivers are often already included in the distribution, but you have to make sure that the actual user is in the `dialout` group[12] to be able to use the serial port. It is also recommended to install `setserial` when working with real RS232 ports.[13]

A built-in USB connector often implies an FTDI chipset (except for the Intellibox and TAMS). In that case it is also possible to install the direct D2xx drivers instead of the VCP driver [14] to bypass the serial port in the operating system. The advantage of these direct drivers on Windows is being able to choose a port by name instead of by a number that may vary (such as COM1, COM2, etc).[15] The serial port library for Mac and Linux already shows a human readable name. D2xx drivers for these platforms are much harder to install than on Windows. They are only needed in case the ports do not show up because of an unusual VID/PID-combination.[16]



All ports that use a serial port connector or a USB connector with VCP driver are prefixed with a blue C. The ones that use the FTDI direct driver have the red D in front of the name.

---

[10] We recommend the Keyspan USA-19HS or one based on the FTDI chipset, because these are the ones used while developing iTrain.

[11] See the download page on our website for the location of the mostly used drivers.

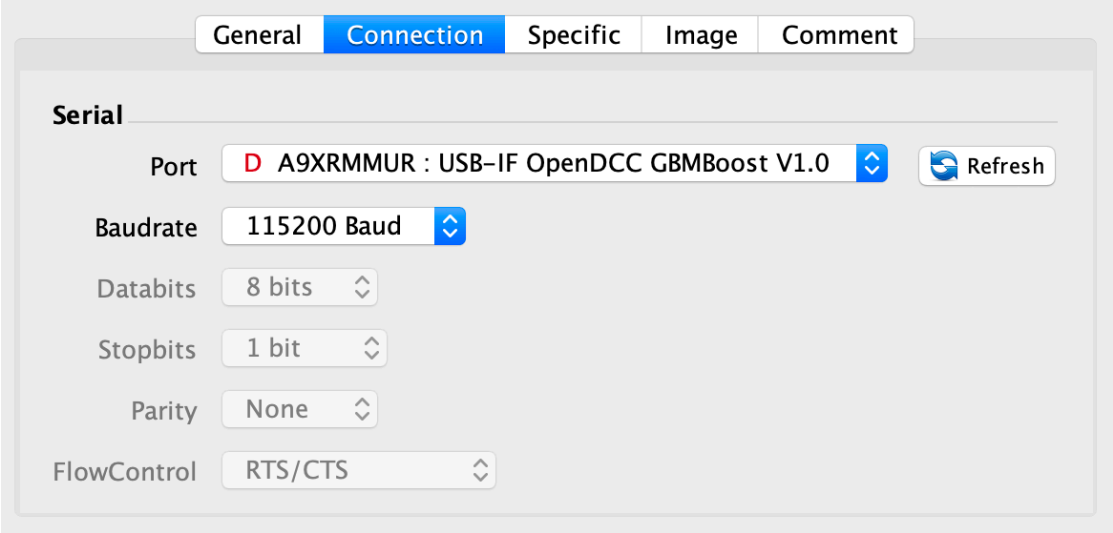
[12] To add a user to the `dialout` group on Linux, use the following command in the terminal:
`sudo adduser <username> dialout`

[13] To install `setserial`, use the following command in the terminal: `sudo apt install setserial`

[14] On Windows both the VCP and D2xx driver package are combined into one CDM installer so you only have to install one driver to get both.

[15] COM1 and COM2 are normally reserved for the physical RS232 ports even if they do not exist. The virtual ports made by the USB-driver are often COM5 or higher.

[16] Some editions of the following devices might not show a port on MacOS without D2XX: Rautenhaus RMX950USB, OpenDCC, BiDiB GBM-Master, Blücher GBM16XN.



You always have to select a port to make a connection. In case the port is not listed, you can use the 'Refresh' button to get a more recent list of choices. All other settings have a more or less default value for this interface and cannot be changed in many cases. Sometimes different command stations use the same protocol, but with other serial port settings. In that case it is possible to change these properties.

Note: In many cases the baudrate can be changed on the command station. iTrain should be set so that the value matches. iTrain will not change the baudrate of the command station.

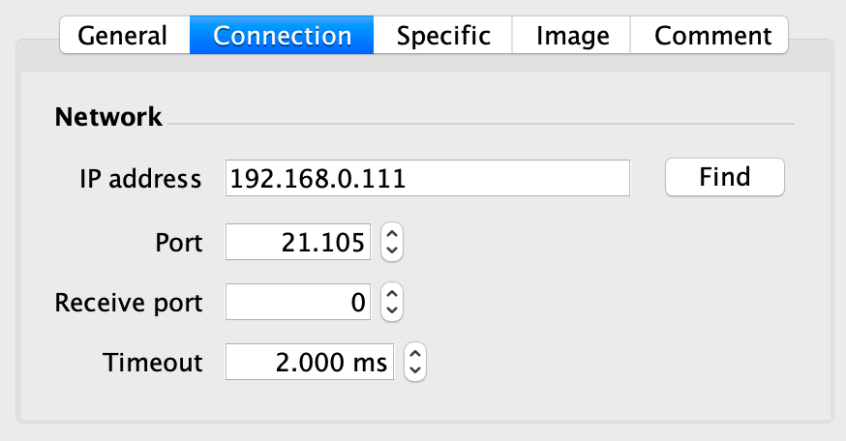
Network interface

All interfaces in the box 'Type' with a green N or n have a Network interface. A network interface means that there is a network to which the interface and the computer connect. This network can be just a cross-cable between the interface and the computer, but it can be your local network at home as well, including your wireless network.

A cross-cable is the easiest option to physically connect the interface with the computer, but the configuration might be somewhat more difficult, because you have to set up a network by choosing your own IP addresses for both the computer[17] and the interface. In general I would recommend using 192.168.2.x with x between 2-253 and a different value for the interface and the computer.

The other option is connecting the interface to your existing network so you don't have to set up your own network. In that case the interface will get an IP address automatically when using DHCP, or you can choose one manually with the restriction that it should match the network number and have a unique host number (2-253 to prevent conflicts with routers). An advantage is that the interface is also connected to the internet for updates, and other handheld devices can connect to it as well without changing cables.

[17] Although you might think that your computer already has an IP address because you can communicate via the wireless network, you have to specify a separate IP address on the cable/ethernet side, because you are setting up a second network that is limited to the cross-cable.

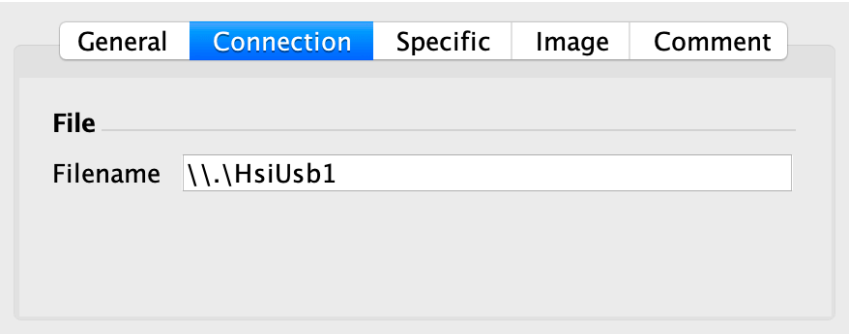


The main information that is needed by iTrain is the IP address or the host name[18] of the interface. For some network interfaces[19] a 'Find' button will be available to automatically find the IP address by sending a broadcast message.

In most cases there is no need to change the default port number, because it is specific for the interface and set by default. The receive port (only visible when using UDP) should not be changed either.

The timeout is used while making the initial connection and to determine if the command station is still reacting. Values should be in the range from 250 to 5000 ms (0 means wait for ever and should not be used).

File interface
All interfaces in the box 'Type' with a blue F or f have a File interface. A file interface means that there is a file to which the interface connects. This is not a real file, but it represents some device in your operating system.



Currently only the type 'HSI-S88 USB' uses this type of connection and it is restricted to Windows.

Specific

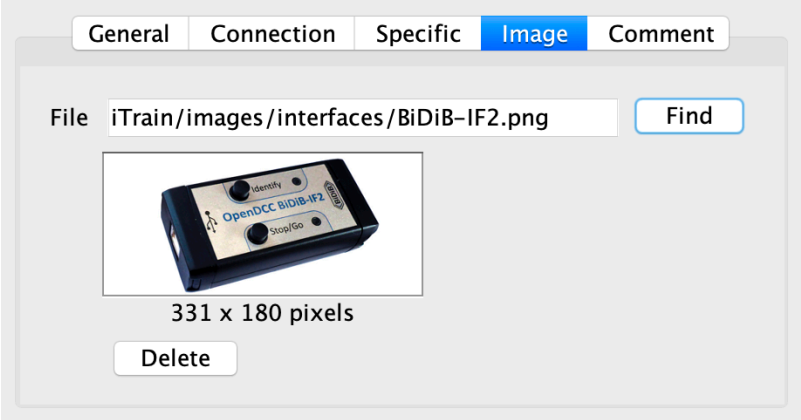
The third tab called 'Specific' is for settings that are different for every interface. The list of supported interfaces has become very extensive and you probably need only one or two for your project. Therefore we refer you to the Appendix D for interface specific information.

---

[18] When using DNS (for example via the router or a server) you can use a name instead of an IP address.

[19] This 'Find'-button is only available for selected network interfaces. In general those are the interfaces that (also) support the UDP protocol (CS2, CS3, Z21, ECoS, µCon), because TCP allows no broadcasts.

Image



For every interface it is possible to select an image that represents the actual device you are using. You can select the image via the resource browser when pressing the button ‘Find’. To remove an image from the interface you can use the button ‘Delete’.[20]

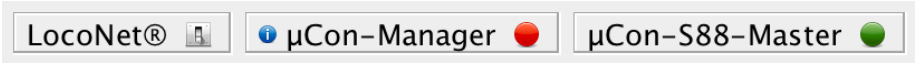
Comments

The last tab ‘Comment’ allows you to add comments about the interface.

[20] The interface image is not used in the current desktop application, but might be used in future desktop or mobile applications.

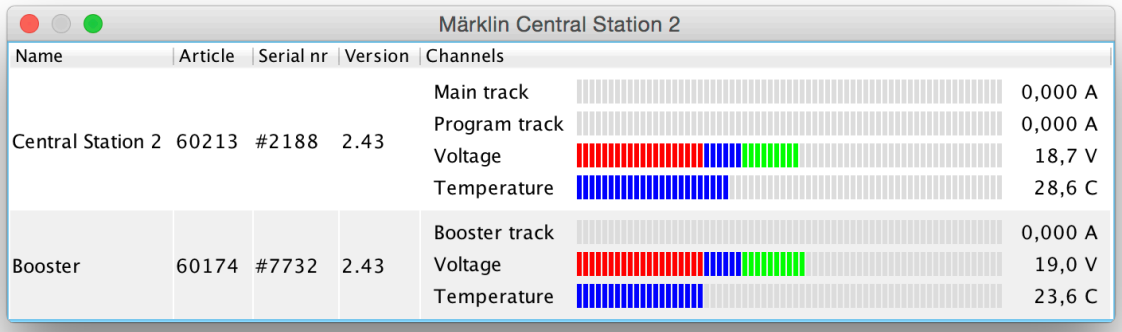
Status

The status of all interfaces is shown on the right side of the status bar. It shows the name or description of the interface and an icon with the status[21]. If the icon shows the  symbol, then the interface is still 'Offline' and you can double click it to go 'Online'. If the interface is 'Online', it shows the track power status ('Go'/'Stop') with a colored green or red ball. You can toggle both statuses by double-clicking on it. To disconnect an interface, hold the Shift-key while double clicking.

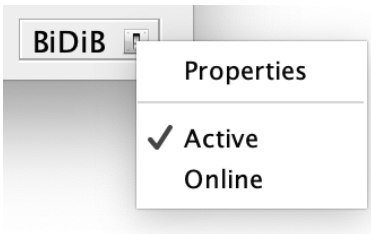


By hovering over the name of the interface it will show a tooltip with extra information about the interface like the firmware version (if available) and possibly other properties. Between brackets the letters VAFB or a subset indicate what the interface has been configured to control (V = Vehicles, A = Accessories, F = Feedbacks, B = Boosters).

In some cases it will show an extra information icon in front of the name. This means more information is available by double-clicking on it, such as with the Central Station 2.



Every interface on the status bar has a popup menu. Via this popup menu you can switch off the active state of the interface. This means it will not react on the global Connect/ Disconnect button at the top anymore. This is useful if the device connected to the interface is currently not operational and should stay offline.



[21] For some interfaces extra information like the voltage or current may be shown.

Trains

A train is a combination of locomotives and wagons that moves as one unit on the rails. Next to that, a set of wagons without a locomotive on a side track is a train, but a consist of three locs is also a train.

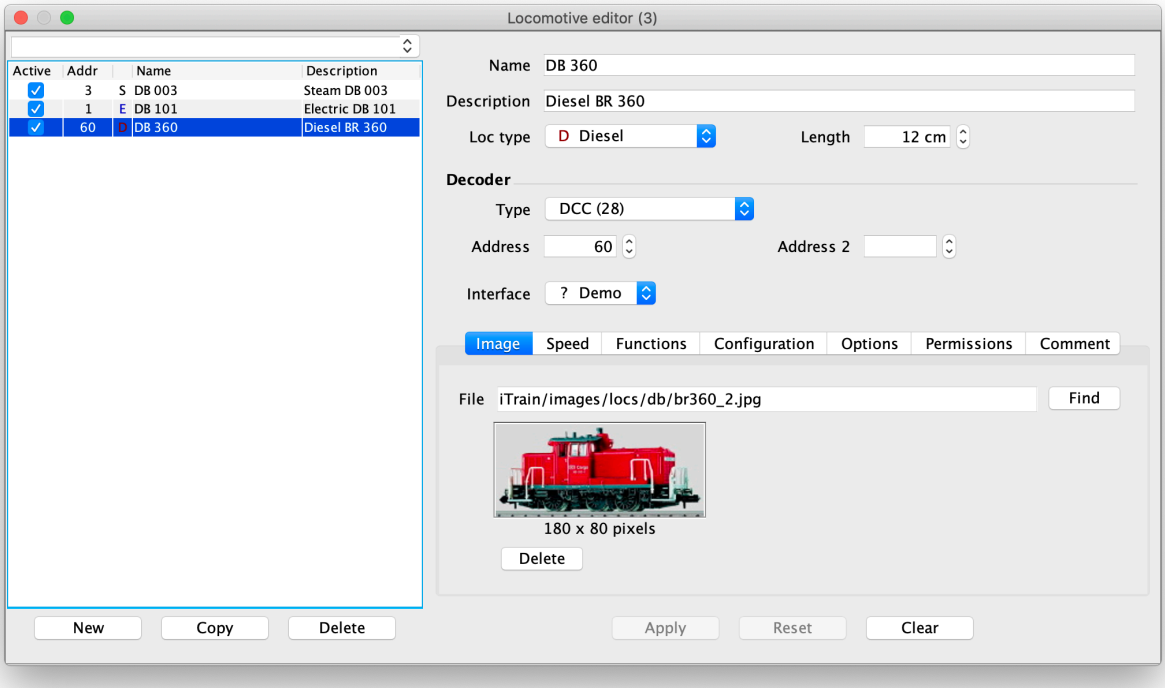
Later on when discussing the switchboard and routes we will refer to the train as the unit to control and to display. Within the train the main locomotive is responsible for moving the train, so speed and direction commands for the train are transferred to this locomotive. Optional other locomotives in the train will follow with the same speed and the correct direction.

We will first explain how to create locomotives, wagons and train types, and describe their properties, before we continue with trains.

Tip: Some interfaces will import your locs automatically (like the ECoS and Central Station) when you go 'Online'. In this case you can start immediately, but iTrain likes to store more information about your locs than these interfaces supply, so then you want to edit them.

Create or edit a locomotive

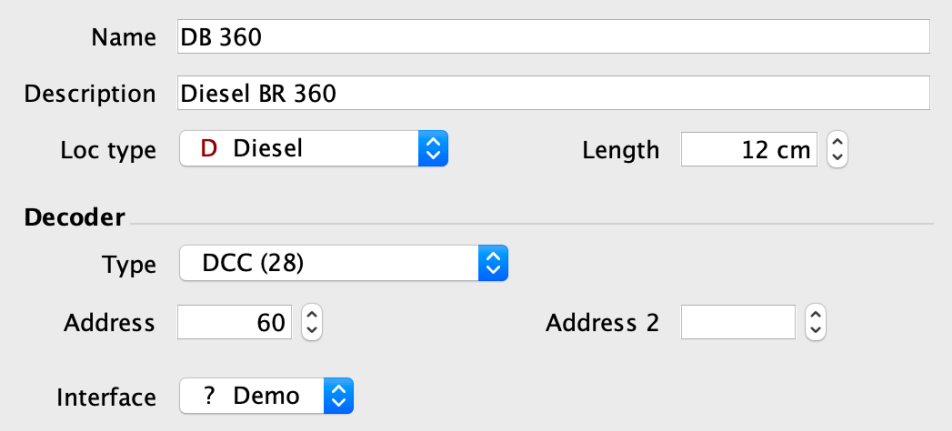
To get into the 'Locomotive editor' you go to the menu 'Edit' -> 'Locomotives' or press Command + F2. The way of creating a locomotive works the same way as with the interfaces, with on the left side the list with locomotives and on the right side the definition of the locomotive.



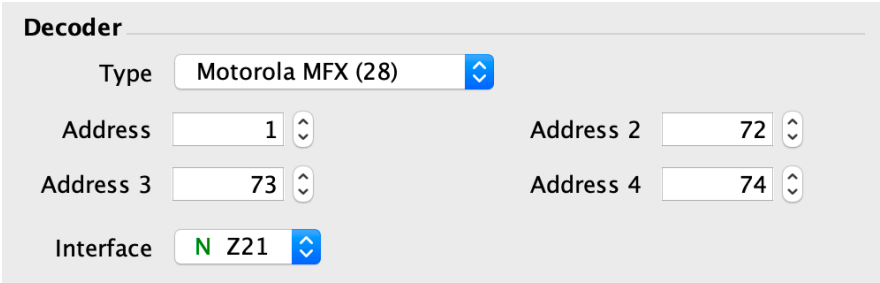
In the 'Active' column you can indicate which locs are currently available on the layout, so that elsewhere in the program where you have to select a loc only the active locs are shown. When in doubt just mark it active.

Locomotive definition

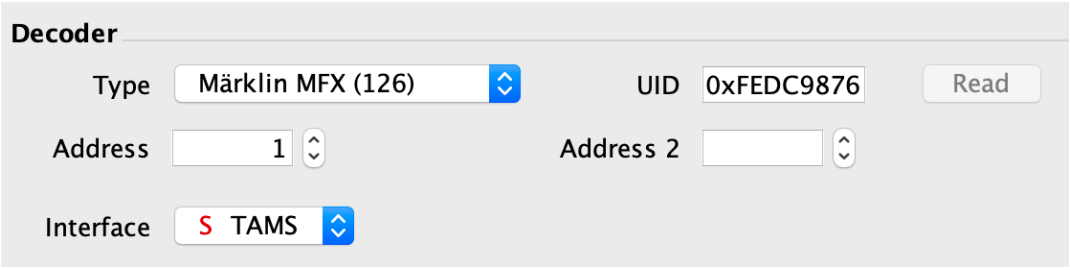
It is important to specify the 'Loc type' correctly when driving automatically (explained later), because electric locs cannot run on tracks without a wire. Specifying the 'Length' of the loc is necessary in order to calculate the total length of the train. This is important, among others, for releasing blocks and turnouts. Take the length from buffer to buffer.



In the 'Decoder' section you can specify the decoder 'Type', the 'Interface' and the 'Address' (for a loc without decoder you choose analog and no address). The additional address boxes are used to switch functions via extra addresses if the decoder protocol is limited in the number of allowed functions per address. This way it is possible to switch f0-f9 or f0-f16 with a Motorola or MFX decoder on any system with Motorola support[22], or f0-f9 on a Selectrix system.



The 'Motorola MFX (28)' choice is used for MFX decoders with a digital system that does not support MFX at all. It controls the decoder with possibly 28 steps (or 14 if the system does not support it) and up to four addresses to support 16 functions, depending on the decoder.

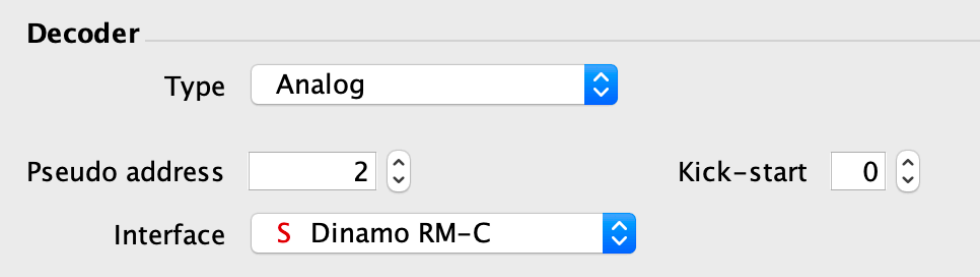


[22] The functions f5-f8 represent f1-f4 of the second address. When using two addresses, function f9 represents the function f0 on the second address, but this is not supported by most manufacturers of decoders. When using more than two addresses, f9-f12 represent f1-f4 of the third address and f13-f16 represent f1-f4 of the fourth address, so f0 of the extra addresses is not being used.

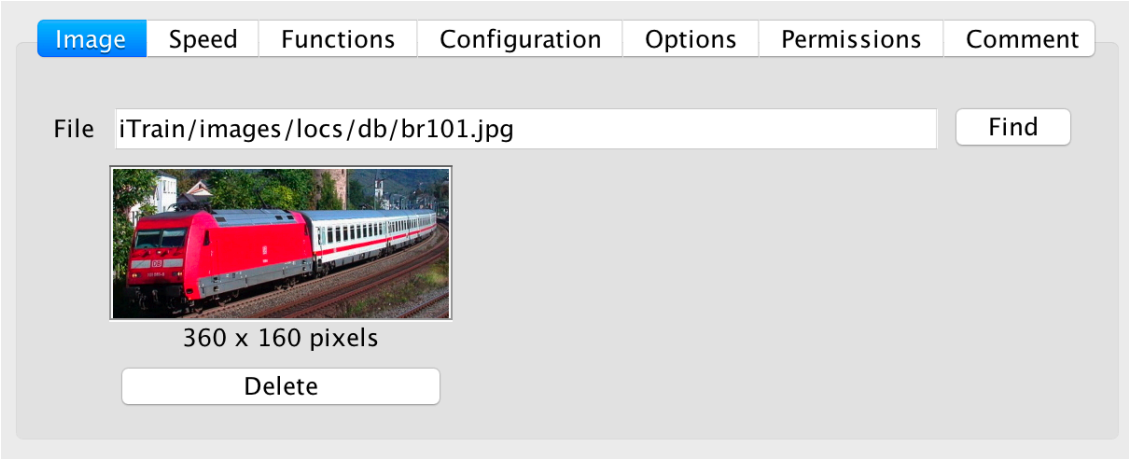
Some interfaces do not fully support the MFX protocol as they cannot receive responses from the decoder, but they do allow you to send the speed, direction and all functions to MFX decoders.[23] If you use such an interface, you have to fill the Unique ID or UID[24] of the decoder to be able to control it.[25] The address, that represents a session ID or SID, can be chosen freely and does not have to be programmed into the decoder, but it needs to be different from other MFX addresses on the layout.

When using an MFX locomotive with the Central Station 2 or 3, you have to fill in the MFX address instead of the Motorola address. In iTrain this is an address starting with 1024, to prevent conflicts with the Motorola addresses.

Tip: Normally the MFX address will already be filled in after the loc has been automatically imported when the interface goes online. But if it is not, then place the cursor in the address field and if it is not empty, fill in '0' and press the 'Enter' key, then change the direction of the loc on the command station. Now the address will appear.



The 'Analog' decoder choice is for block controlled systems like Dinamo with analog locs without decoder. The address field is replaced with a field 'Pseudo address' that is optional and should only be filled in when controlling the loc manually from an address based remote control. An additional field 'Kick-start' is available for better slow driving characteristics when starting the loc..

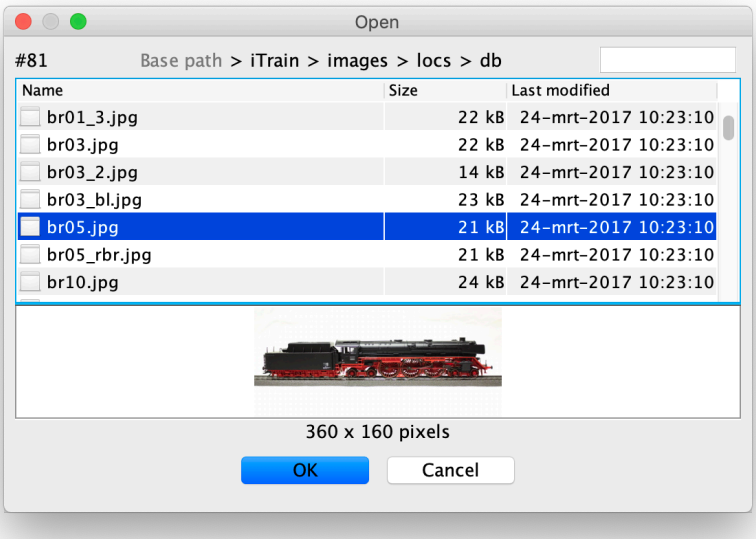


[23] TAMS calls this protocol M3, as something between M2 (Motorola version 2) and M4 (the name ESU uses for MFX).

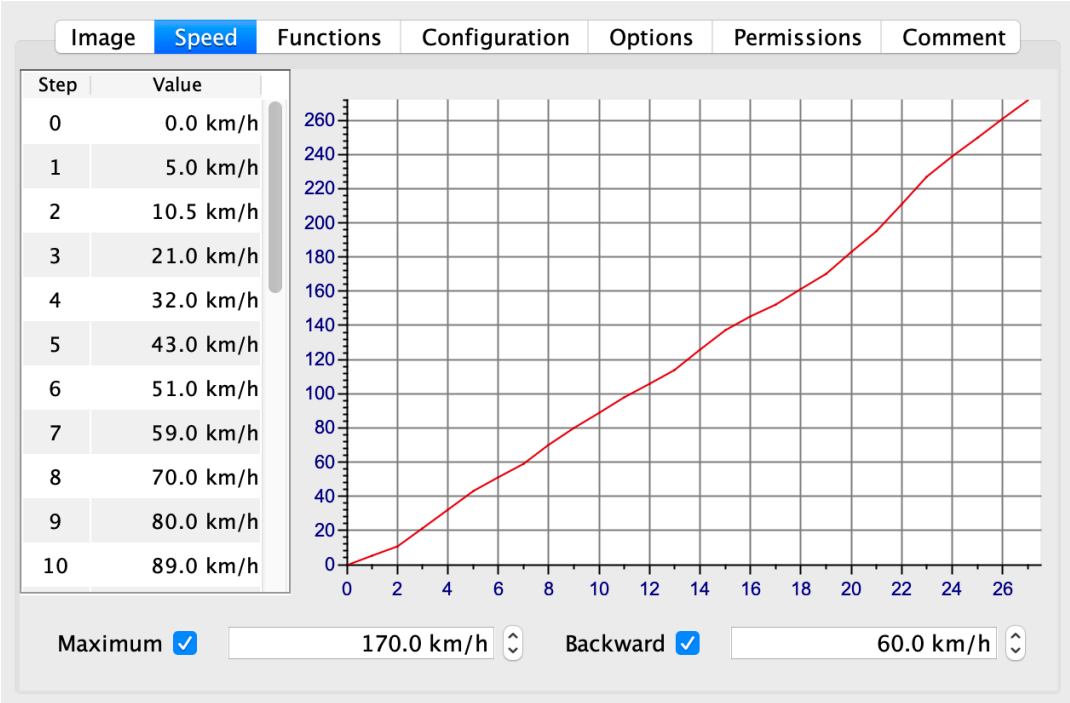
[24] The UID starts with '0x' indicating it is a hexadecimal value with characters from 0-9 and A-F.

[25] The button 'Read' is currently only available with the interface 'Tams RedBox'. Reading happens on the programming track and can take up quite some time (5-60 seconds) depending on the decoder.

The first tab ‘Image’ allows you to add an image to your loc. An image has a preferred size of 360 x 160 pixels that is prepared for high-resolution displays[26], but images with 180 x 80 pixels are still allowed although not recommended. Use the ‘Find’ button to open the image resource browser to select an image.



Large locomotive images are always downscaled to fit in 360 x 160 pixels before they are used in iTrain. If your image is much larger, you can first scale it down in another program. Save all your images preferably in a folder `images` under your base path. Now you have all the images in the correct size and together in one location, so it is easy to migrate your iTrain installation to another computer if that is necessary later on.

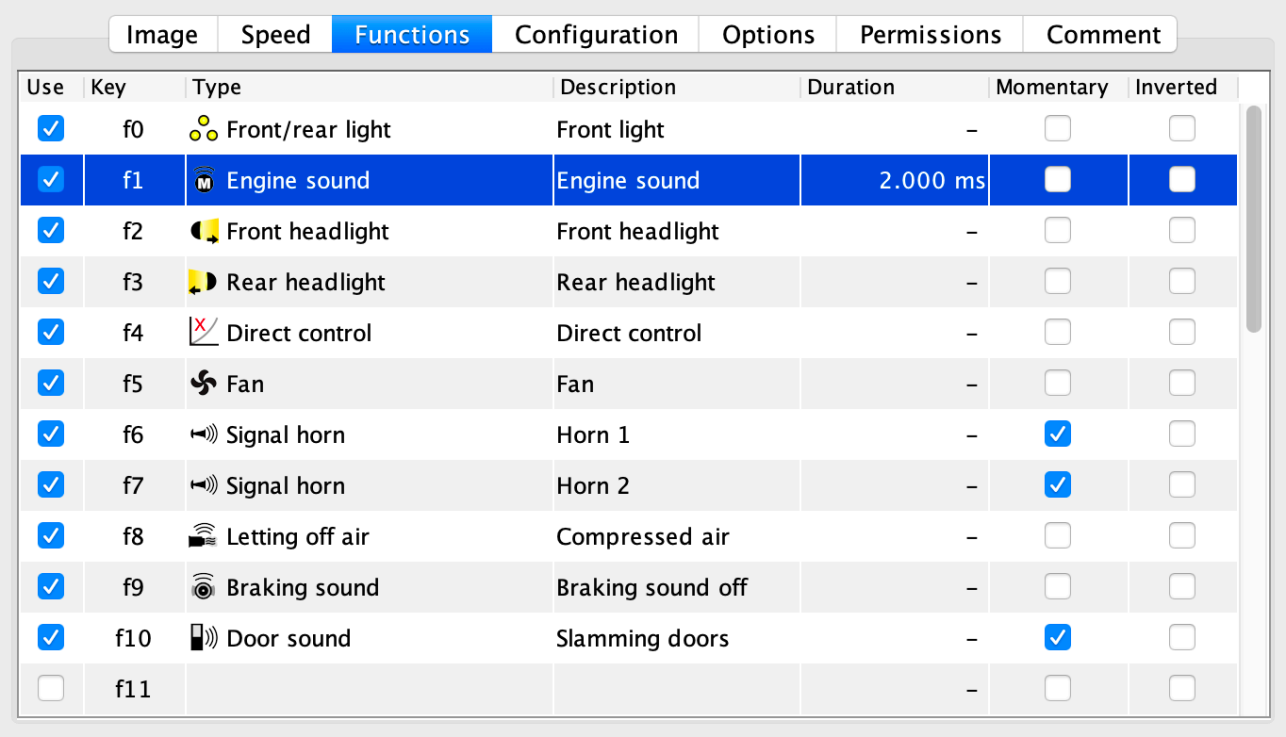


[26] High resolution display or HiDpi displays can be found on an iPhone, iPad, iMac, MacBook (Pro), Mac Pro and many Android devices.

The second tab ‘Speed’ shows the speed related properties. The upper part shows the real speed characteristics in km/h of the loc against the decoder steps. You can manually enter the values if you calibrate outside of the program or use the speed measurements dialog (described in the next chapter and available via menu ‘View’ -> ‘Speed Measurements’) to do the actual speed measurements in iTrain.

At the bottom, the maximum speeds when running automatically can be specified. Uncheck the box ‘Maximum’ if there is no maximum speed for the loc. You can specify a separate maximum speed for the backward direction, but if you don’t, it will be the same as the forward direction.

On the third tab ‘Functions’ you can assign the loc functions to the correct f-keys. Depending on the decoder type and the availability of a second address, the maximum number of allowed functions is calculated independently of the allowed number of functions of the ‘Interface’. So all available functions in the loc can be defined, but maybe not all of them can be activated, because the interface does not support so many functions (for example with an MFX decoder on an old system). When migrating to a new system, these functions will be available automatically.



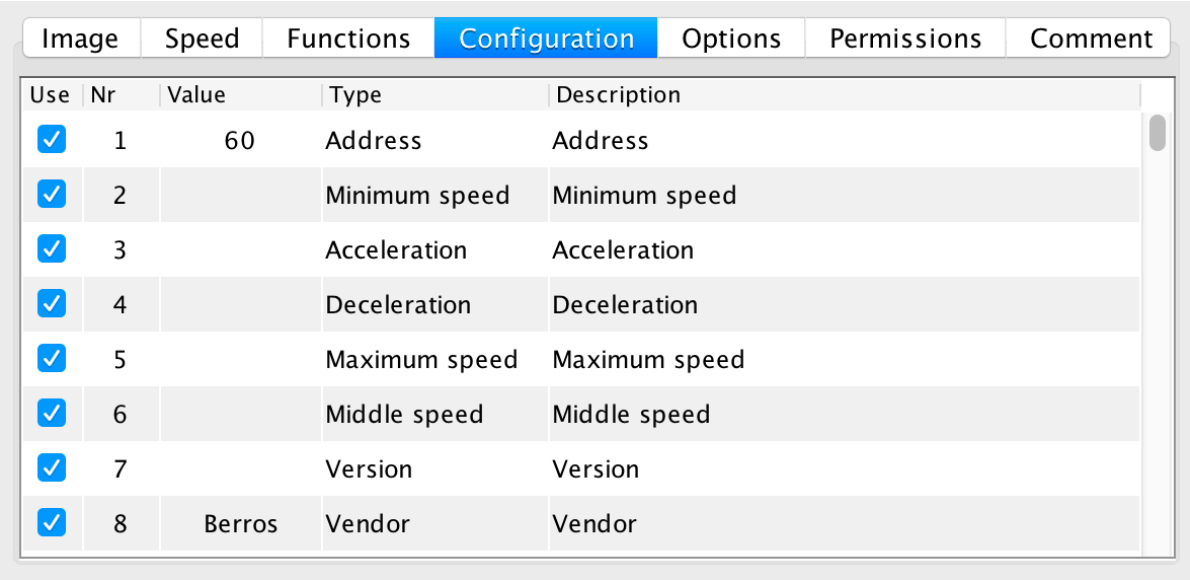
A function can be assigned to an f-key by checking the first column ‘Use’ and selecting a type from the ‘Type’ column. Automatically, a default description will appear that you can edit to describe the function in more detail.

In the column ‘Duration’ you can specify how long a function takes to execute. In case of ‘Engine sound’ this is used to delay acceleration when starting after speed step 1 with the specified time period. It prevents acceleration in iTrain while in reality the locomotive is not moving, because the sound decoder prevents it.

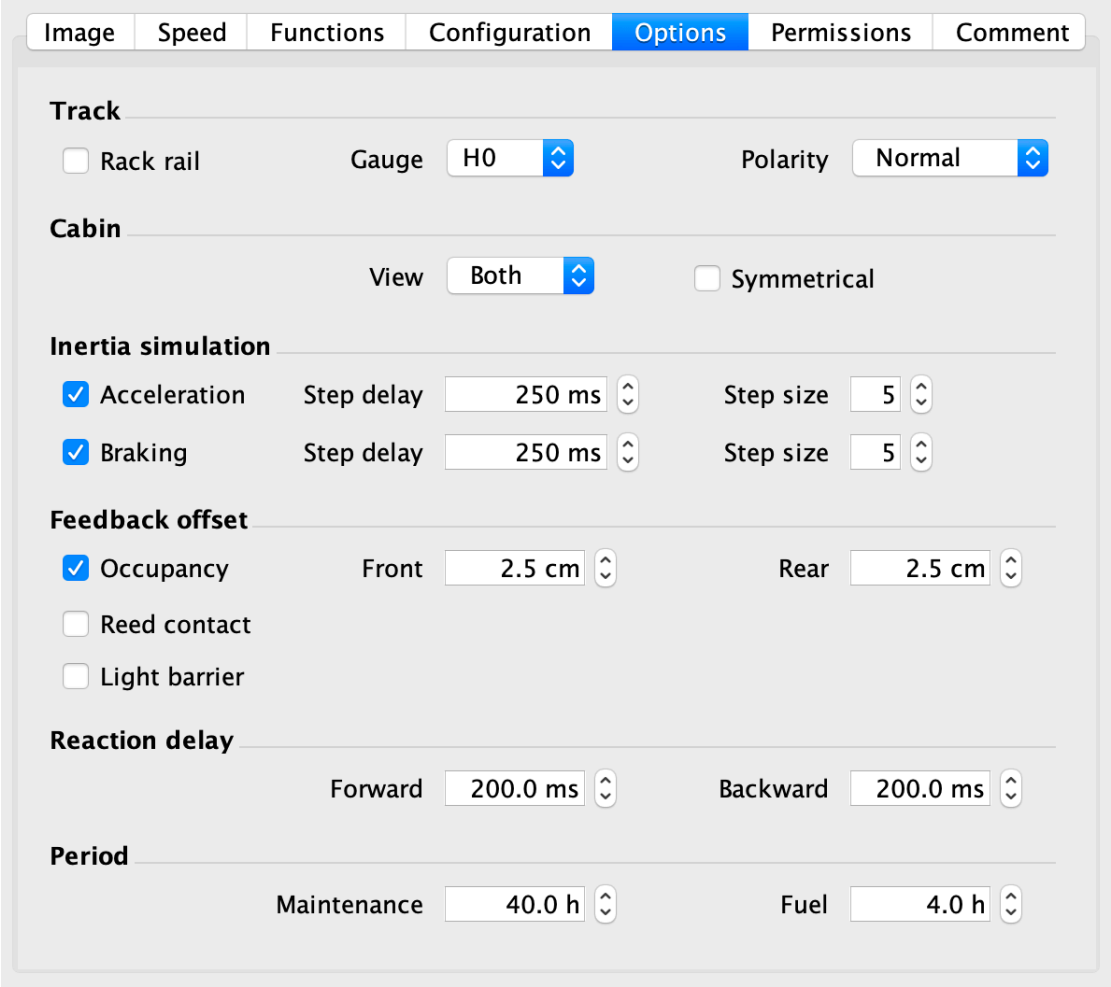
The ‘Momentary’ box checked means that the function will only be activated as long as the corresponding button or key is pressed, for example for a whistle. This in contrast with the default behavior that will toggle between the on and off state of the function.

The ‘Inverted’ box checked means that the state of the function is inverted. This can be used for functions in which you can switch off a certain behavior (sound or light), but you want to define it in iTrain as something to switch on.

Note: The function called ‘Direct control’ in iTrain disables the acceleration and deceleration settings of the decoder without changing the speed (normally on f4). In case the activation of the function also reduces the speed for every decoder step, the function ‘Slow’ should be used instead.



The fourth tab ‘Configuration’ shows the decoder configuration. Here you can define the configuration variables/parameters that apply to this decoder. How to use this is described in the next chapter called ‘Decoder Programming’.



The fifth tab ‘Options’ shows some extra options for locomotives. These have been divided into subsections.

Track

The ‘Gauge’ is the necessary width of the track for this locomotive and can be used when you use both narrow and normal tracks in one layout. Additionally you can specify whether the loc supports a ‘Rack rail’. In combination with the same block settings this limits the loc directly to specific tracks on the layout.

The polarity settings are used to indicate how the loc has been wired. There are two appliances:

- Analog locomotive (without a decoder) - to correct the loc when it always drives in the wrong direction with a Dinamo system.
- Digital locomotive with RailCom® decoder - to correct the direction when a RailCom® detector always shows the wrong direction only for this locomotive.[27]

Cabin

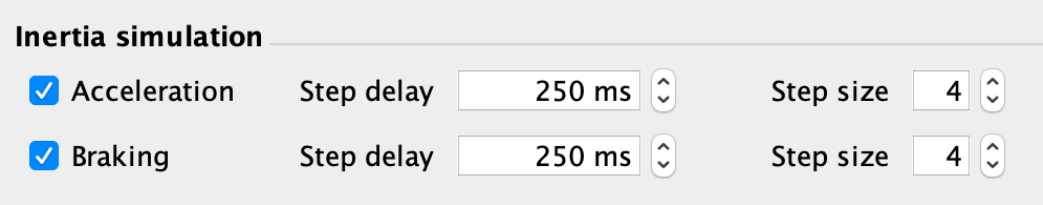
The ‘View side’ indicates on which sides the loc driver has a view on the tracks. Normally this will be ‘Both’, but in some rare cases a loc only has a cabin on one side.

[27] For 3-rail the polarity and the direction of the loc are not related. Therefore you have to use ‘Unknown’ for the polarity when using RailCom®.

The flag ‘Symmetrical’ can be used to indicate that the locomotive does not have a clear front and rear side, but is more or less symmetrical. This will prevent unnecessary turns on a turntable.

Inertia simulation

The inertia simulation works by delaying the speed steps sent to the decoder. Single steps are executed almost immediately, but when a speed change requires multiple decoder step changes, the intermediate steps are sent with a delay between the step change as specified. If the decoder has many decoder steps, it is better not to use all the intermediate steps, but use a bigger step size. For 126 decoder steps we recommend a value of 4 or 5.



You can set the inertia simulation separately for the acceleration and deceleration. The first value is the step delay and the second value the step size. To disable the iTrain Inertia simulation, uncheck the check box.

Note: Good values for the inertia simulation are typically between 100 ms and 500 ms. Do not use lower values than 100 ms, because that does not allow a speed command to be sent by the ‘Interface’ and processed by the loc before another speed comment will be sent.

Feedback offset

Feedbacks are the eyes of the system to see what is happening on the layout, so they must have the correct information. Depending on the type of feedback (see Appendix B) sometimes a correction needs to be applied.

When using occupancy feedbacks these are normally triggered by the wheels of the loc and possibly not by all the wheels. So when the feedback is triggered the loc is already somewhat further than expected. To correct this error, you have to specify the offset from the front buffer to the first wheel and from the rear buffer to the last wheel of the loc that activates a feedback.

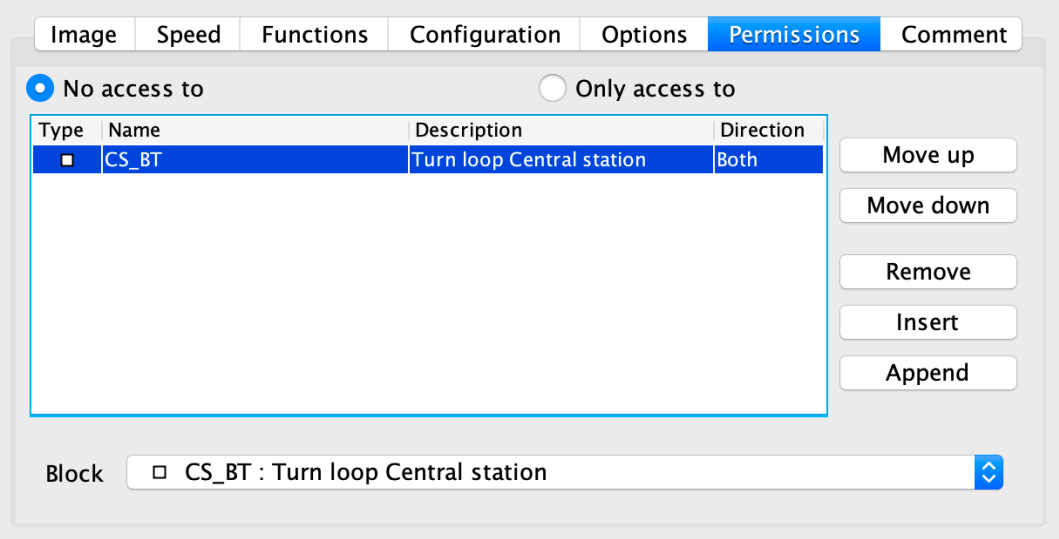
In case of reed contacts the correction is dependent on the magnets positioned below the locomotive. You have to specify the offsets from both buffers to the closest magnet. For light barriers this correction is normally small, but to be complete this has been added as well.

So depending on what type of feedbacks you use in combination with a loc, you have to specify the offset for that type of offset from both the frontside and the rear side.

Reaction delay
The reaction delay is only used when using 'Positions', described later on in the manual. With 'Positions' you can use exact positions (for example in cm) in the block to stop a loc based on time/distance calculations. Normally there will be an offset error at the entry of a block, because calculations are relative and not absolute. By experimenting with the reaction delay you can correct this offset for multiple blocks. This delay can have different values for the forward and backward direction.

Period
Every locomotive counts the time it is running. You can set maintenance and a fuel period, so that when this time has been reached there will be an indication in the loc overview. For maintenance you will get red text in the 'Time' column. For fuel there is an extra column 'Tank' with an indicator of the fuel level that will turn red when it is almost empty. You can use the Shift key + double click with the mouse to reset the fuel level again.

iTrain will not influence the running characteristics after the time period, so it is just an indication at the moment.



The sixth tab 'Permissions' is to exclude access ('No access to') or only allow access to specific blocks for this loc. Use the buttons on the right to add rows and use the block input box at the bottom to change the block of the selected row. In some cases the block should / should not be accessible in a specific direction. For those cases change the 'Direction' column as required.

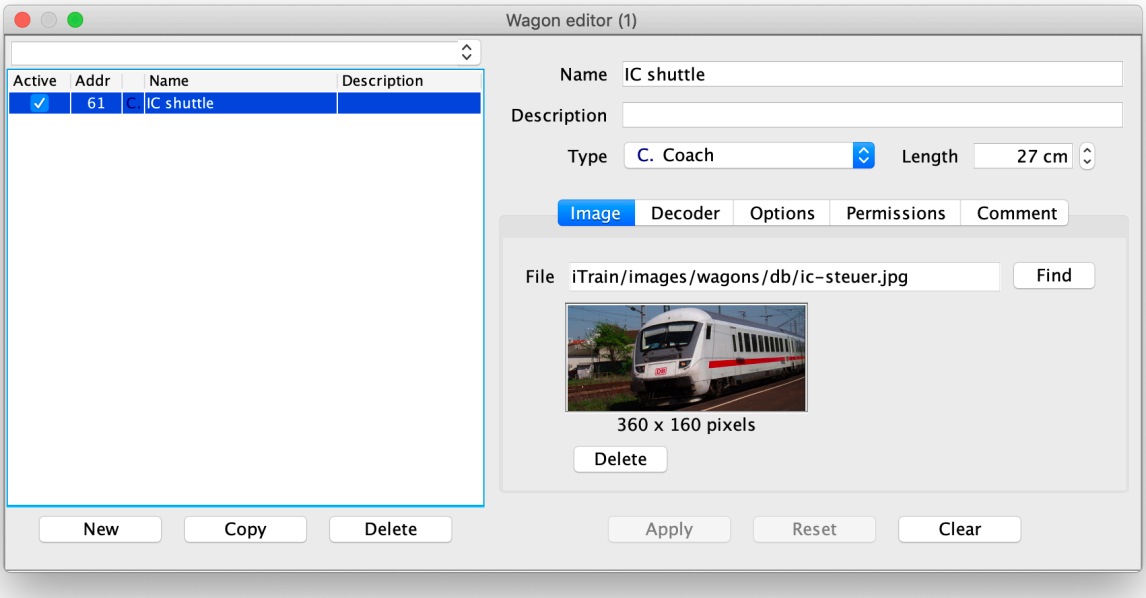
Note: These permissions should only be used to exclude the loc from certain blocks in which it is always forbidden to enter, because it might cause accidents (for example narrow curves or other reasons for derailment). It is not necessary to add blocks, because of gauge or no electric power, as this is already implicitly forbidden.

The seventh tab 'Comment' allows you to add comments about the loc. You can use it to write maintenance remarks, info about the loc in real life or anything else you find useful.

Create or edit a wagon

A wagon is an item of rolling stock that has no motor. In iTrain you can enter all your wagons with their properties, but it is not necessary in order to drive automatically. It is mainly useful if you change your train compositions regularly. It is nevertheless recommended entering wagons (and coaches) with a function decoder in it so you can

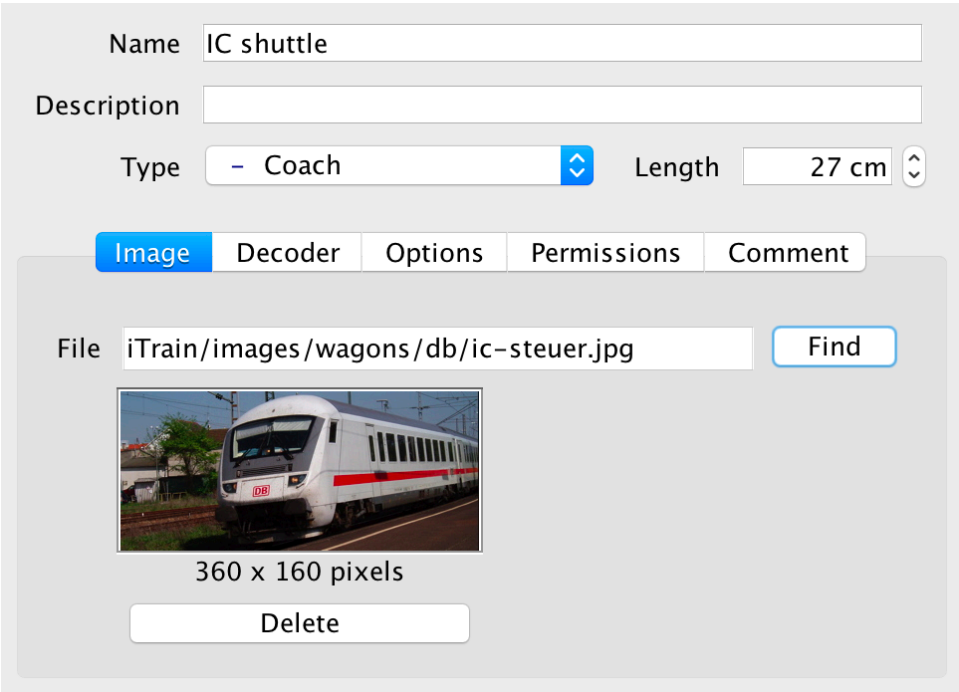
control their functions. If you have multiple wagons of the same kind, you cannot create one wagon and use it multiple times, but you have to copy it. Every wagon knows to which train it belongs and a train can only include every wagon once.



To get into the 'Wagon editor', you go to the menu 'Edit' -> 'Wagons' or press Shift + Command + F2. The way of creating a wagon works the same way as with the locomotives, with on the left side the list with wagons and on the right side the definition of the wagon.

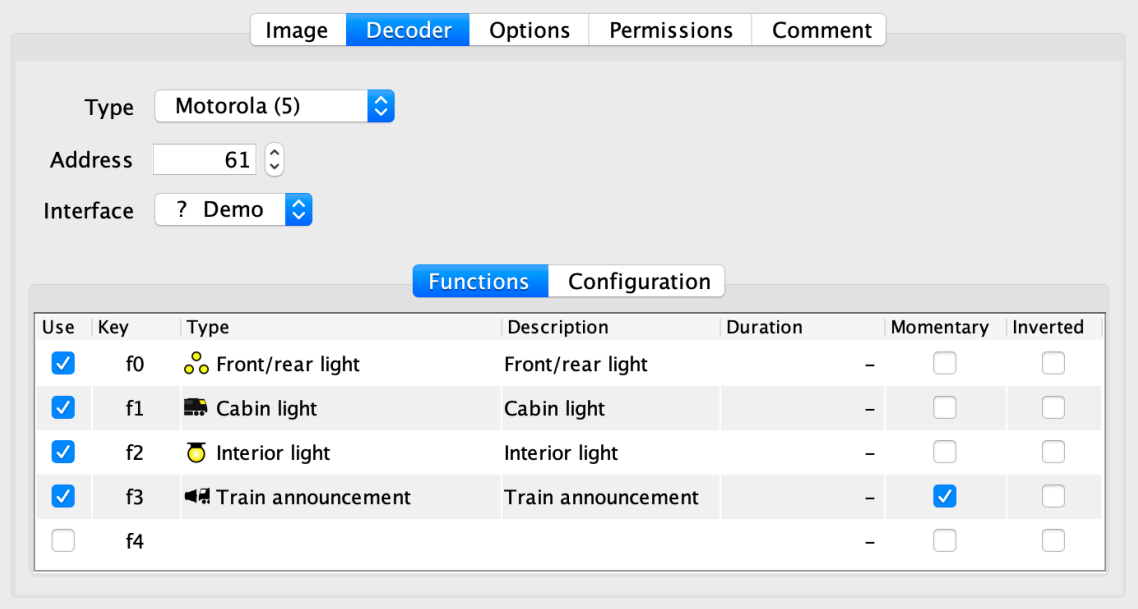
Wagon definition

The settings of wagon properties are similar to the locomotive properties, so we will focus on the differences. We assume you have read the chapter about the definition of a locomotive.



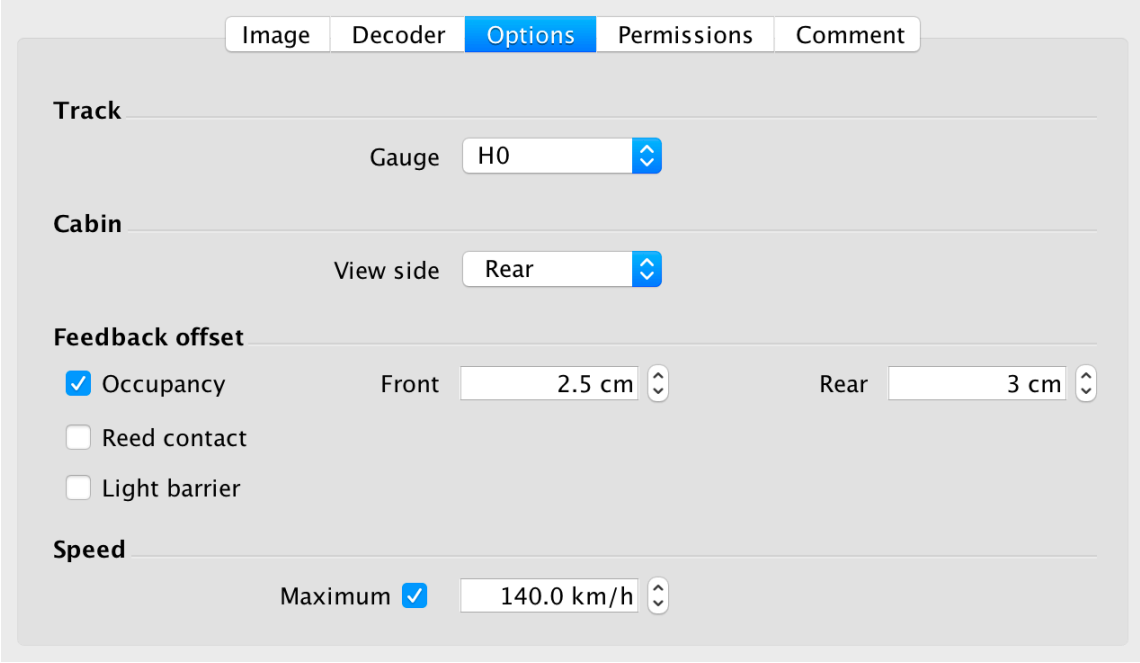
The ‘Type’ specifies if it is a freight wagon (red letter) or a coach (blue hyphen) and of what kind. At the moment this information is not used in the routing logic, so if you don’t know what to choose just take something. For completeness it is there and it might be used in future versions. The ‘Length’ of a wagon should be entered to calculate the total length of the train. Take the length from buffer to buffer.

The first tab ‘Image’ is similar to the one with the locomotives.



The second tab ‘Decoder’ specifies the decoder settings for this wagon if applicable. You first have to specify the decoder before you can enter anything. Only select a decoder if this wagon has its own decoder with an address that is not equal to the address of a locomotive in the same train. If multiple wagons share the same address then they can be considered one large wagon, because using them in separate trains conflicts with the concept in iTrain that every independent vehicle should have its own address.

The tab ‘Decoder’ has been split into two sub-tabs for the ‘Functions’ and the ‘Configuration’. Both can be defined in the same way as for a locomotive.



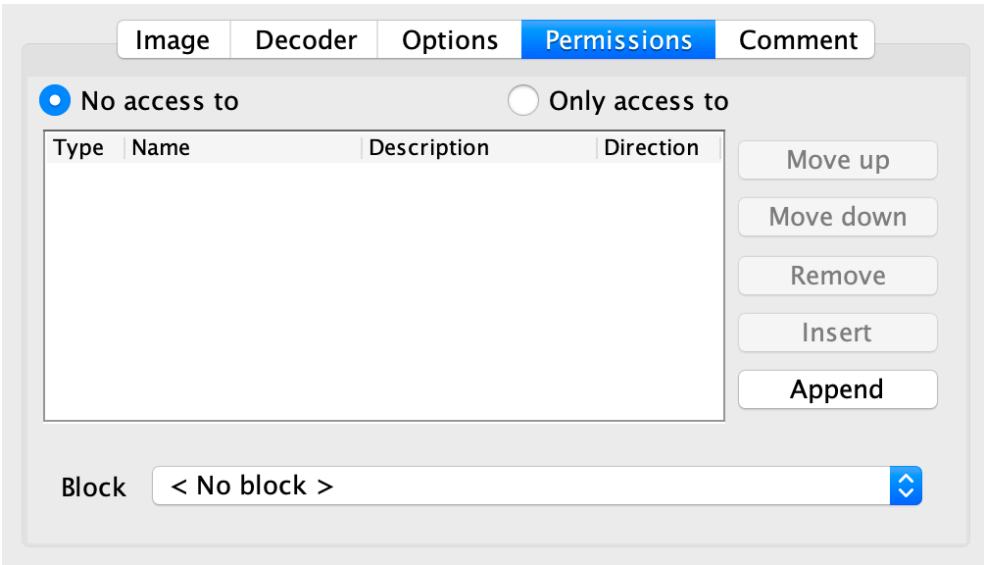
The third tab ‘Options’ shows different properties that are not related to other tabs.

In the ‘Track’ section you can specify the ‘Gauge’ or track width.

The option ‘View side’ in the section ‘Cabin’ indicates if the wagon has a driver’s cabin and if so on which side. An available cabin allows a train to drive in this direction safely without speed restrictions, but only if the cabin is at the outside of a train.

The section ‘Feedback offset’ is the same as for a locomotive. If the wagon cannot be detected by any feedback type used, just uncheck all options. In this case the length of the wagon will be added to the feedback offset of the total train if there are no detected items at a side of a train.

In the section ‘Speed’ a maximum speed for this wagon can be specified if necessary.



The fourth tab ‘Permissions’ is to exclude access (‘No access to’) or only allow access to specific blocks for this wagon. Use the buttons on the right to add rows and use the block input box at the bottom to change the block of the selected row. In some cases the block

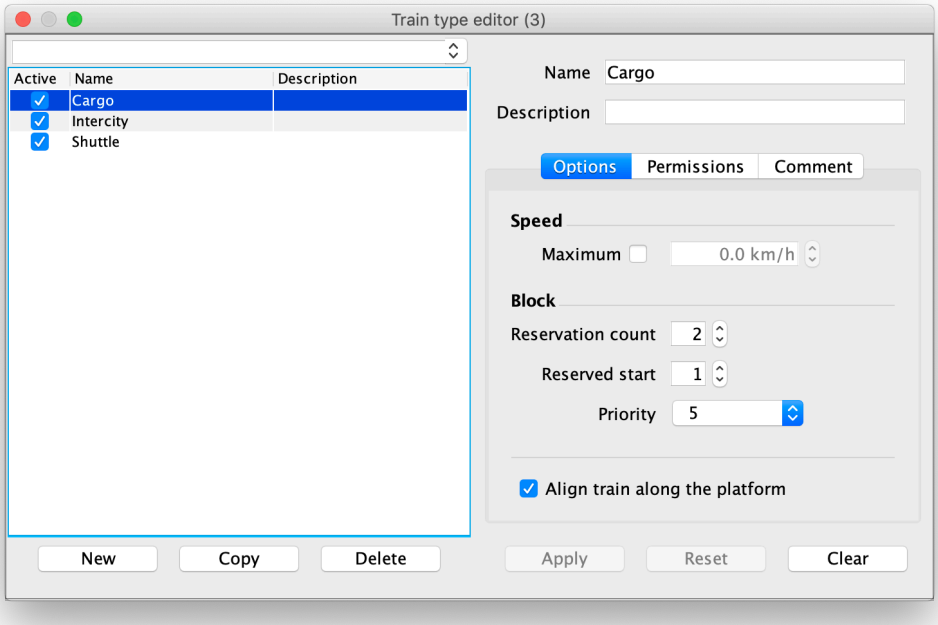
should / should not be accessible in a specific direction. For those cases change the 'Direction' column as required.

Note: These permissions should only be used to exclude the wagon from certain blocks in which it is always forbidden to enter, because it might cause accidents (for example narrow curves or other reasons for derailment). It is not necessary to add blocks, because of gauge, as this is already implicitly forbidden.

The fifth tab 'Comment' allows you to add comments about the wagon.

Create or edit a train type

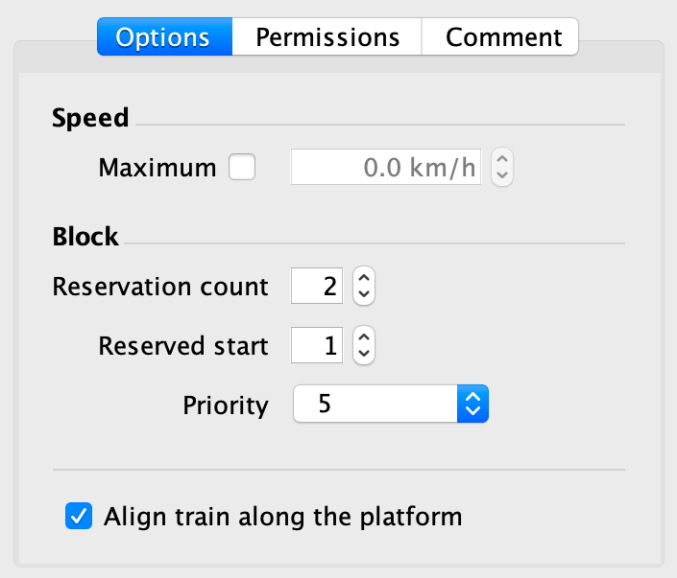
A train type is a category of trains. Train types are used to specify specific properties just once per train type instead of per train, but also to indicate a kind of train in relation to a station. Per station you can specify which train types are allowed to make a stop and if so, how long (explained later).



To get into the 'Train type editor', you go to the menu 'Edit' -> 'Train types' or press Shift + Command + F3. The way of creating a train type works the same way as with the locomotives and wagons, with on the left side the list with train types and on the right side the definition of the train type.

Train type definition

The settings of train types properties are limited to only a few properties. The most important settings have to be made at the station object discussed later on.

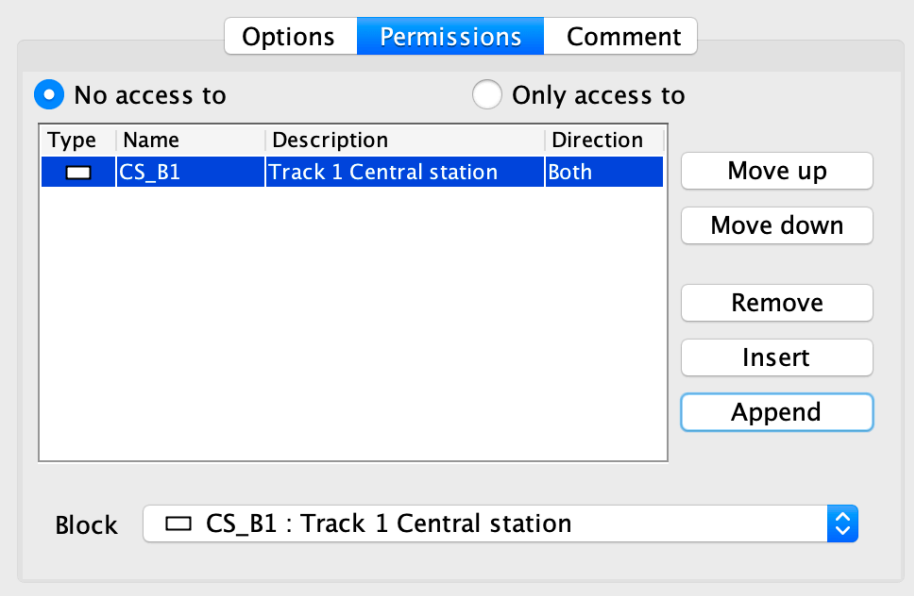


On the first tab ‘Options’ you can specify the maximum speed for this train type in the section ‘Speed’.

For running automatically you can specify the number of blocks that should be reserved in front of the train in ‘Reservation count’. Take into account that this value is a suggestion and sometimes more might be reserved in case of critical blocks (explained later) or fewer if no more blocks can be reserved. The field ‘Reserved start’ indicates the number of blocks that need to be reserved before a train should leave after a scheduled stop. This value should not be higher than the ‘Reservation count’.

If multiple trains want to reserve the same block, then normally the train that first requests it will get it when the block becomes available. The field ‘Priority’ allows you to influence the order. At first the trains with priority 1 will get it and so on. So the list of trains waiting to reserve a block is sorted on priority. Once a block has been reserved by a train, another train with a higher priority cannot take it over it and has to wait until the block becomes free again.

In a station block with a platform a train will normally align along the platform. To let the train stop at the end in front of a signal and ignore the platform, you can uncheck the box ‘Align train along the platform’.



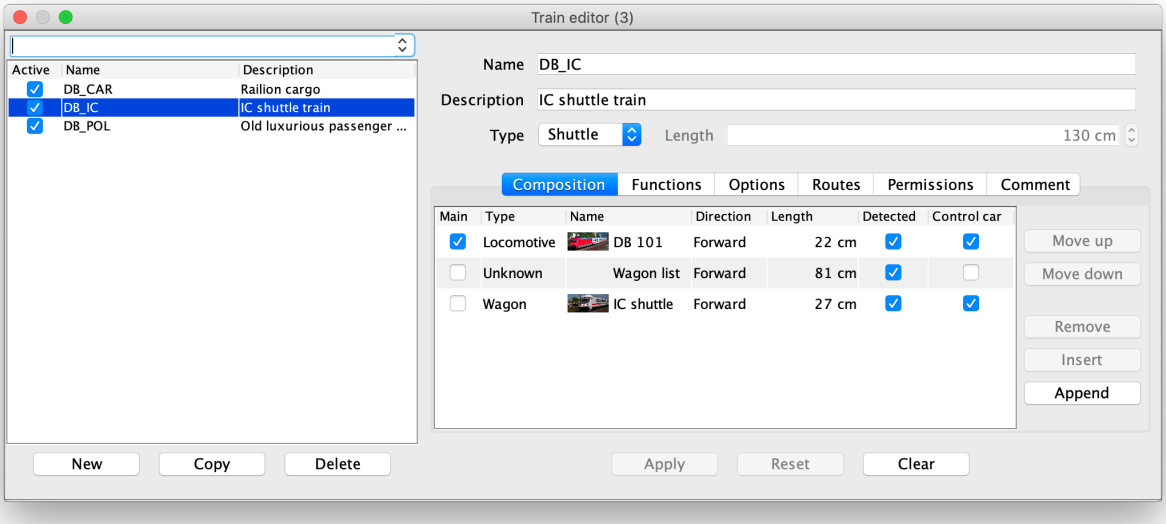
The second tab ‘Permissions’ is to exclude access (‘No access to’) or only allow access to specific blocks for this train type. Use the buttons on the right to add rows and use the block input box at the bottom to change the block of the selected row. In some cases the block should / should not be accessible in a specific direction. For those cases change the ‘Direction’ column as required.

Note: The permissions of a train type are essential for automatic routing. These permissions limit access of all trains with this type to certain parts of the layout. This role is different from the role of permissions for locomotives and wagons.

The third tab ‘Comment’ allows you to add comments about the train type.

Create or edit a train

Now that we have described how to create all objects necessary to define a train, we can continue with the train itself.

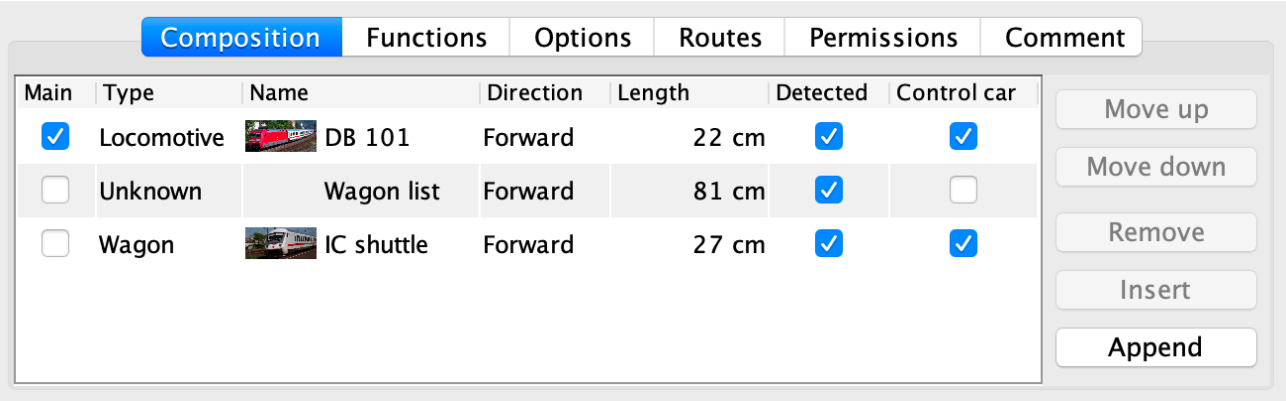


To get into the ‘Train editor’ you go to the menu ‘Edit’ -> ‘Train types’ or press + Command + F3. The way of creating a train works the same way as with the locomotives, wagons

and train types with on the left side the list with trains and on the right side the definition of the train.

Train definition

A train is a composition of locomotives and wagons regarded as a unit. This unit has a user-definable train type that can be specified in the field ‘Type’. The total composition has a length that is normally calculated from the individual lengths of the vehicles in the train. This length is displayed in the field ‘Length’ and should not be edited, because it will be overwritten again. You should only enter a value if one or more vehicles have no length specified, but it is better to have specified the length for the individual vehicles.



Composition
The first tab ‘Composition’ is a list of all the vehicles in the train in order. When the train is driving ‘Forward’ the vehicle at the top is considered to be the front of the train and the last element in the table at the bottom is considered to be the rear. The buttons on the right side allow you to insert, append, remove or move a vehicle in the train.

N.B. The direction of the train is determined by the direction of the main locomotive and the ‘Direction’ definition of the main locomotive in the composition table. So if the main locomotive is defined ‘Forward’, then the train and locomotive direction are the same.

For every new vehicle in the train you have to specify if it is an already defined locomotive or wagon, or an unknown part of the train - named a ‘Wagon list’. This ‘Wagon list’, whose properties will be specified in the table itself, prevents you from having to enter all wagons before you can use the train. However, you can only specify the length of this ‘Wagon list’, if it is detected and if it has a control car at the end. For the locomotive and wagon these properties are also displayed in the table, but they cannot be edited because you have to change them in the locomotive or wagon editor.

Tip: Defining all wagons in your trains separately is a powerful feature, but absolutely not a necessity. When starting with iTrain it is recommended creating wagons only for wagons with a function decoder with its own address and using the ‘unknown wagon list’ to fill up the other space in the train next to the locomotives. Later on you can decide to define wagon objects for wagons that are not in fixed combinations, so you can move them easily from train to train.

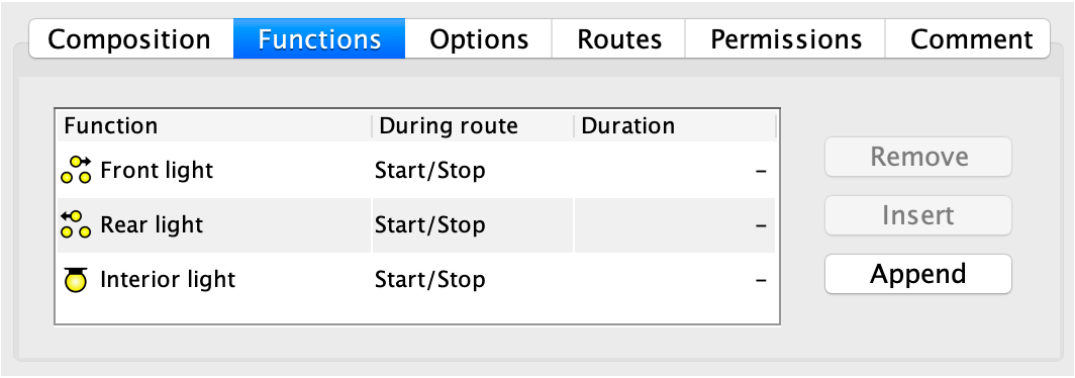
Consists can be created by adding multiple locomotives to the train. When adding multiple locomotives to a train you have to specify what the main locomotive is by checking this locomotive in the table. The main locomotive is the locomotive that gets the desired speed

requests and the other locomotives will follow the main locomotive when accelerating and decelerating.

For every locomotive you have to specify the direction of the locomotive in the train in the column ‘Direction’. When you specify ‘Forward’ the direction of the locomotive and the train are the same, but if you specify ‘Backward’ the train will drive forward if the individual locomotive drives backward. In a consist it is very important to set all the directions correctly to get the train moving.

Functions

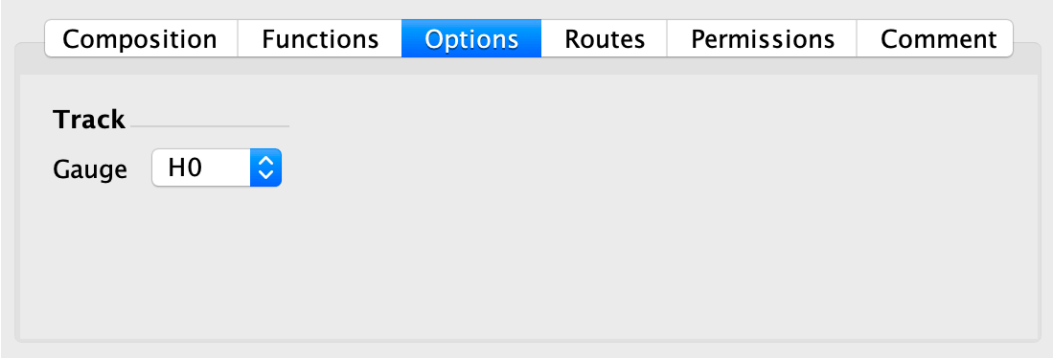
The second tab ‘Functions’ defines the functions in the train that will be activated when driving automatically. These functions are defined per train and will be executed on one or more of the vehicles in the train depending on their type. Functions that do not exist in the containing locomotives or wagons will not be executed.



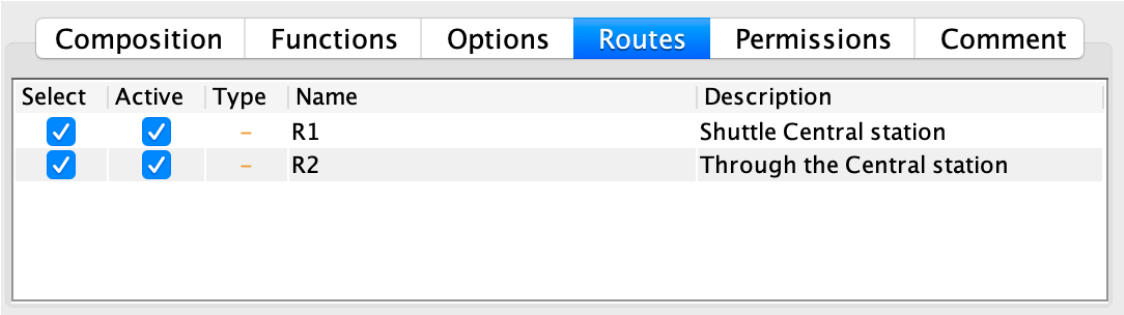
The buttons on the right allow you to append, insert or remove a function for this train. In the first column you can select the type of function optionally with a side (front or rear). In case the function has a side specified, this side is of the train itself and not of the individual locomotive or wagon. iTrain will convert this function to a function on the side of a locomotive or wagon at the front or rear. If a function has no side, but multiple locomotives or wagons have this function, then normally the function will be activated on the first vehicle in the driving direction that has the function. However, many light functions related to wagons (for example interior and platform light) are activated on all possible locomotives and wagons.

The second column determines if the function is activated (Start) when automatic driving starts, deactivated when automatic driving stops (Stop) or both (Start/Stop). Only starting a function could be useful for some lights that you want to be turned on, but never off. Only stopping a function is useful for sound functions that you manually want to start, but should be switched off automatically.

In the third column you can specify how long a function takes to execute. The time specified in ‘Duration’ is waited before the train really starts. For example, the sound of switching on the motor may take a few seconds.

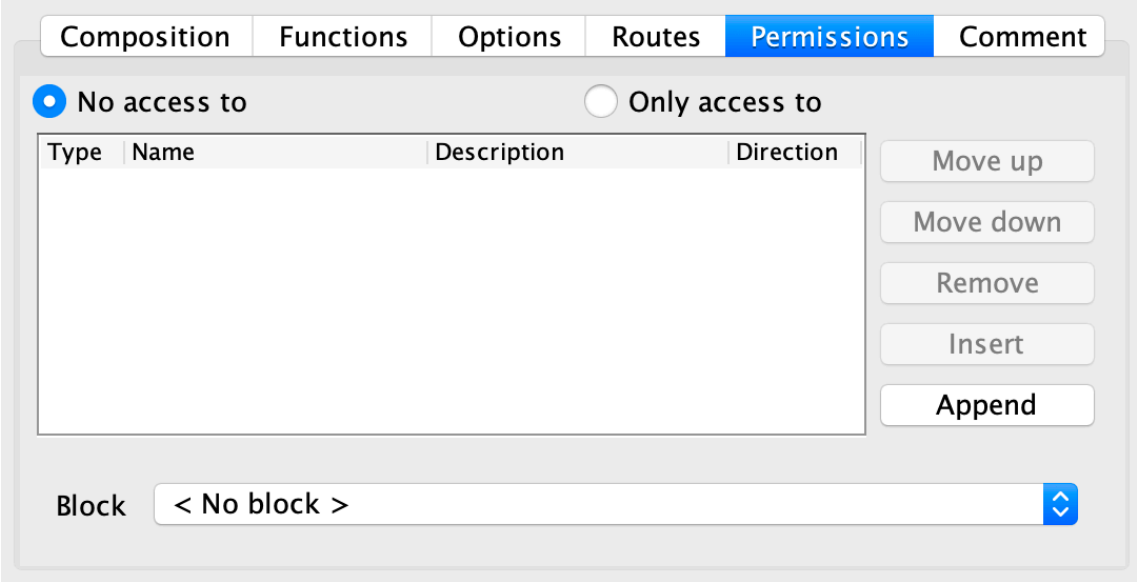


The third tab 'Options' has just a few options[28], because most are already specified per locomotive or train or in the composition table. In the 'Gauge' selection box you can select on what kind of tracks the train is allowed.



The fourth tab 'Routes' is to select all defined routes that apply to this train. It limits the routes that can be selected for this train elsewhere in the program.

Note: It is important to select the routes that apply, to be able to change the active route for a train in an action.



The fifth tab 'Permissions' is to exclude access ('No access to') or only allow access to specific blocks for this train. Normally you will specify this per train type, but you can also

[28] The editable field 'Maximum' has been removed in version 5. The maximum speed will be calculated based on the maximum speeds of the locomotives, wagons and the train type, and will be stored in the train.

change permissions per trains itself. Use the buttons on the right to add rows and use the block input box at the bottom to change the block of the selected row. In some cases the block should / should not be accessible in a specific direction. For those cases change the 'Direction' column as required.

Note: The actual permissions used for the train are calculated based on the permissions specified here and on the permissions of the locomotives and wagons.

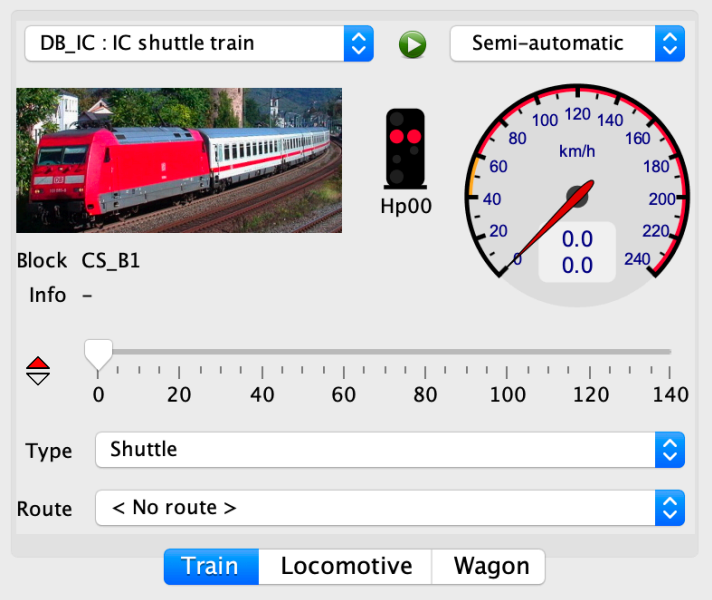
The sixth tab 'Comment' allows you to add comments about the train.

Train control

A train is normally controlled by a driver in the locomotive. The 'Train control' is used to have full control over the train, locomotives and wagons in the train. It consists of three tabs: 'Train', 'Locomotive' and 'Wagon'.

Note: The contents of these three tabs are synchronized meaning that when selecting a train, loc or wagon on one of them, the contents of the other tabs change accordingly.

Train



The first tab 'Train' is the main tab for directly controlling the train. To select a train in the 'Train control', you can use the drop-down box (only the active trains are shown) in the left upper corner or you can select it via the 'Train overview' (described later). You will get a picture of the main loc, the current signal state the train is facing in the block, a speedometer, the current block of the train, some route info, the direction, the desired speed, the train type and the currently selected route.

To modify the speed there are several options:

- Move the slider that represents the desired speed.
- Use the scroll-wheel of the mouse while the mouse is positioned in the 'Train control' in combination with the 'Alt' key on the keyboard to change the desired speed.
- Use the '-' key to decrease the speed and the '+' key (or '=' key so that you don't need the Shift key) to increase the speed. The speed will be changed to the previous or next

speed that can be divided by 5. For example, starting with speed 72 km/h the following values will be 70, 65 for '-' or 75, 80 for '+'.

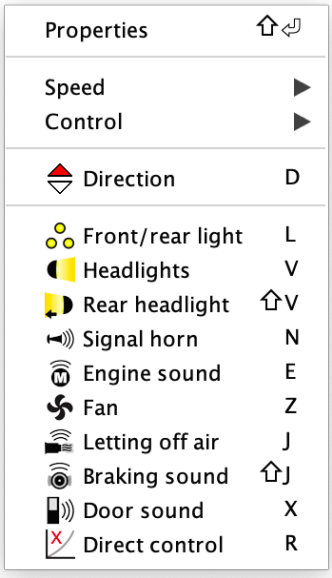
- Use the number (0-9) keys on the keyboard to set the speed from 0 to 90 km/h in steps of 10 km/h, and use the Shift key in combination with these keys to set the speed from 100 to 190 km/h.

The speedometer shows the actual speed in km/h according to the speed measurements. In case of no speed measurements the maximum decoder speed is considered to be 140 km/h. In the gray rectangle of the speedometer you see two speeds: the desired speed (as set with the keys or by the program) on top and the actual speed that the train is driving at the bottom. These two speeds may be different if the desired speed cannot be matched exactly with the decoder step of the main loc. In that case the decoder step is chosen whose real speed closely matches the desired speed while at the same time not topping it by more than 5 km/h (to prevent it from going much faster than desired).

The speedometer becomes a bit darker when there is no activity, and it lights up when the train is driving. In case of multiple train panels it is now immediately visible in which panel there is activity.

The rounded yellow/orange bar on the speedometer indicates the range between the reduced and maximum speed of the block the train is in. The rounded red bar indicates the range beyond the maximum speed allowed in the current block. If the train is driving faster than allowed, the pointer will be highlighted in red. This is useful if you are driving manually, but at the same time want to drive according to the speed limits.

To change the direction of the train you can press the direction button in the left lower corner. When using the keyboard, you can use the 'D' key to change the direction. This will set the speed to zero and change the direction. Another option is to use the 'Backspace' key. This will stop the train if it is running and change the direction if the train is not running.



Functions of the main loc can be changed by a special key, as can be seen in the popup menu attached to the 'Train control'. Momentary functions will only be activated as long as the button or key is down. All other functions will change their state when pressing the button or key.

Tip: To use the keyboard, the Train control needs to have focus. To give the Train control the focus via a key, just press F3. You can now use all the key commands. A full list of commands is available in Appendix A.

The drop-down box at the right upper corner is for setting the control type of the train. There are three options:

- Manual control - the program will not influence the train in anyway, but it still tries to follow where the train is on the layout.
- Semi-automatic control - the program will only reduce the speed of a train when the train needs to stop.

- Automatic control - the program automatically sets the speed to the maximum speed allowed by the train in combination with the block, and stops the train if necessary.

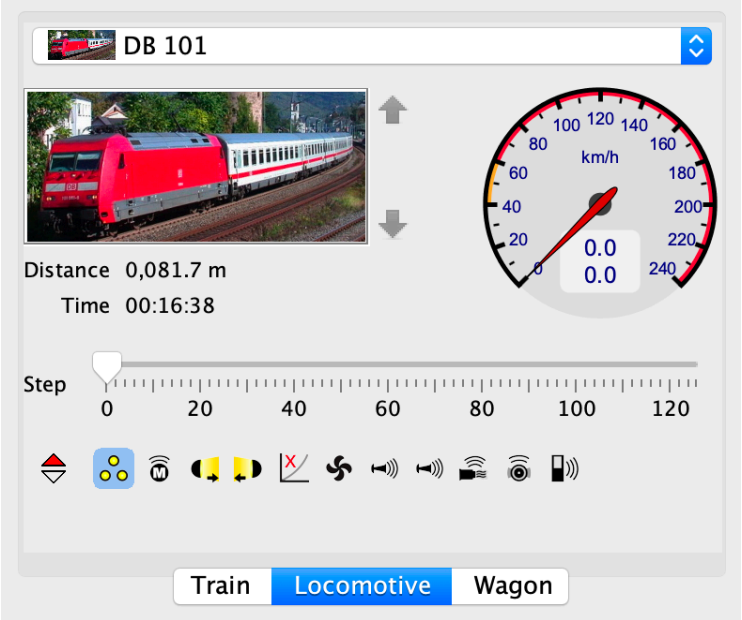
The button will start automatic driving for a train and the button will stop automatic driving. In case a train has been selected, one of the two buttons will appear before the control type input.

Note: It is important to distinguish the control type from the button for automatic driving. The control type is only used to indicate if and how the speed of the train will be influenced based on signals and maximum speeds. So they are focussed on the driver itself. The buttons for automatic driving will influence how the train will drive and select a path over the blocks for the train under control of the program.

As extra information the facing signal for the driver is shown by a symbol between the image and the speedometer. If no signal has been defined on the layout on that position, the general signal will be used. Below the image, the current block and some information about the current route is displayed. In case the line starts with an arrow this means that the block name(s) of the next stop will be displayed. In case of alternative stops, they are all displayed separated by a 'I' and they will be reduced to one when the final selection has been made.

Locomotive

The second tab, 'Locomotive', shows all properties that directly belong to the locomotive and not to the train as whole.

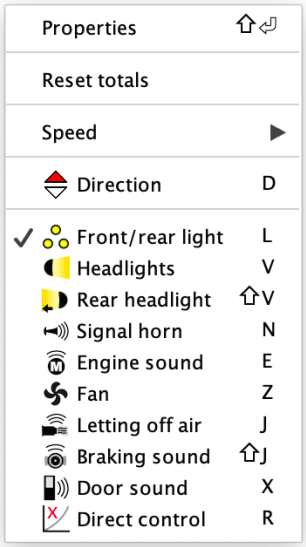


To select a loc, you can use the drop-down box (only the active locs are shown) in the left upper corner. You will get an image of the loc, a speedometer, the distance and time travelled, the decoder step, the direction and all the loc function states.

The speedometer works the same as on the train panel, but the slider represents the actual decoder step and changes are directly applied without inertia delays. To modify the desired speed with the inertia simulation, there are two options:

- Use the '-' key to decrease the speed and the '+' button (or '=' button so that you don't need the Shift key) to increase the speed. The speed will be changed to the previous or next speed that can be divided by 5. For example, starting with speed 72 km/h the following values will be 70, 65 for '-' or 75, 80 for '+'.
- Use the number (0-9) keys on the keyboard to set the speed from 0 to 90 km/h in steps of 10 km/h, and use the Shift key in combination with these keys to set the speed from 100 to 190 km/h.

Functions can be changed by pressing the button with the function icon. For many important function types a special key is reserved, as can be seen in the attached popup menu. Momentary functions will only be activated as long as the button or key is down or for the time as defined in the 'Duration' of the function. All other functions will change their state when pressing the button or key.

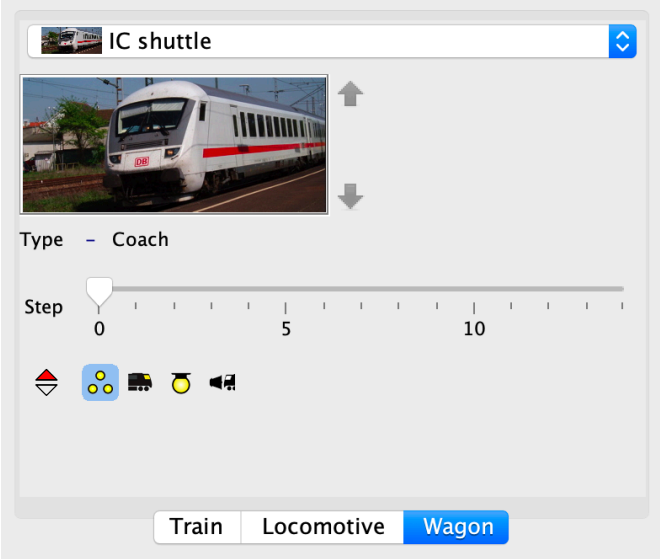


The popup menu item 'Reset totals' is used to reset the total distance and time travelled by a loc to zero. For example, use it after service of the loc to start a new maintenance period.

The vertical arrow button next to the loc image can be used to loop quickly through the available locs in the same train, in case of a multi-traction.

Wagon

The third tab, 'Wagon', shows only properties that directly belong to a wagon. It is similar to the tab 'Locomotive', but with less information. Below the image you see the type of the wagon, the decoder step slider[29], the direction and the functions.

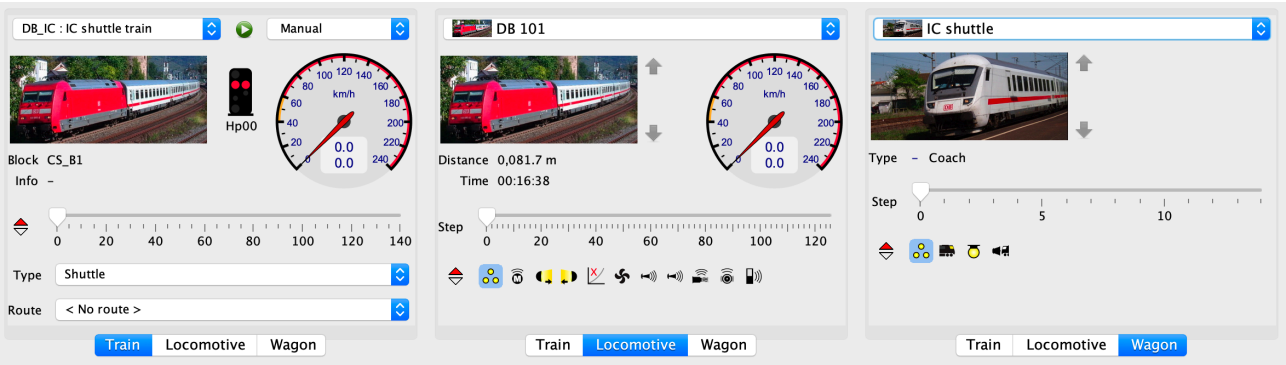


The vertical arrow button next to the wagon image can be used to loop quickly through the available wagons in the same train.

[29] The step slider is for wagons with a decoder to manually set the decoder step such as with vacuum cleaner wagons. It is not meant as a speed step, because it is a wagon and not a locomotive.

Grid

Instead of having one ‘Train control’ it is also possible to have multiple ones. They will be shown in a grid, so next to each other and/or above each other. For each control you can decide which tab should be shown, so you can mix trains, locs and wagons.



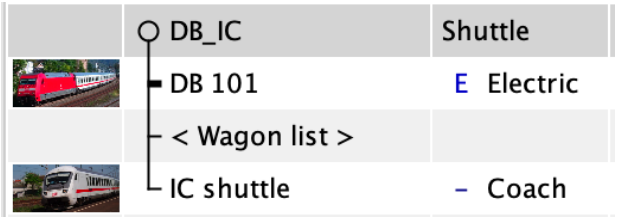
To add or remove extra train controls, one of them needs to have focus (press F3) and you can use the Alt key + cursor keys to size the grid. So Alt + ‘cursor right’ will add controls to the right (or add columns) and Alt + ‘cursor down’ will add them to the bottom (or add rows). To remove controls, use Alt + ‘cursor left’ to remove a column and Alt + ‘cursor up’ to remove rows.

The focussed train control has a border drawn around the image. To navigate between the controls, use ‘Shift’ + cursor keys. To change the selected train, loc or wagon within a control, you can use the ‘ENTER’ key to popup the list with items and select one with the cursor keys followed by pressing ‘ENTER’ again.

Overview

The train overview, in the upper left corner, shows all active trains with some of the important properties, but you can also expand a train to look which locomotives and wagons are inside.

By double clicking on the closed circle before the train name, you can expand or unfold the train to show its composition.

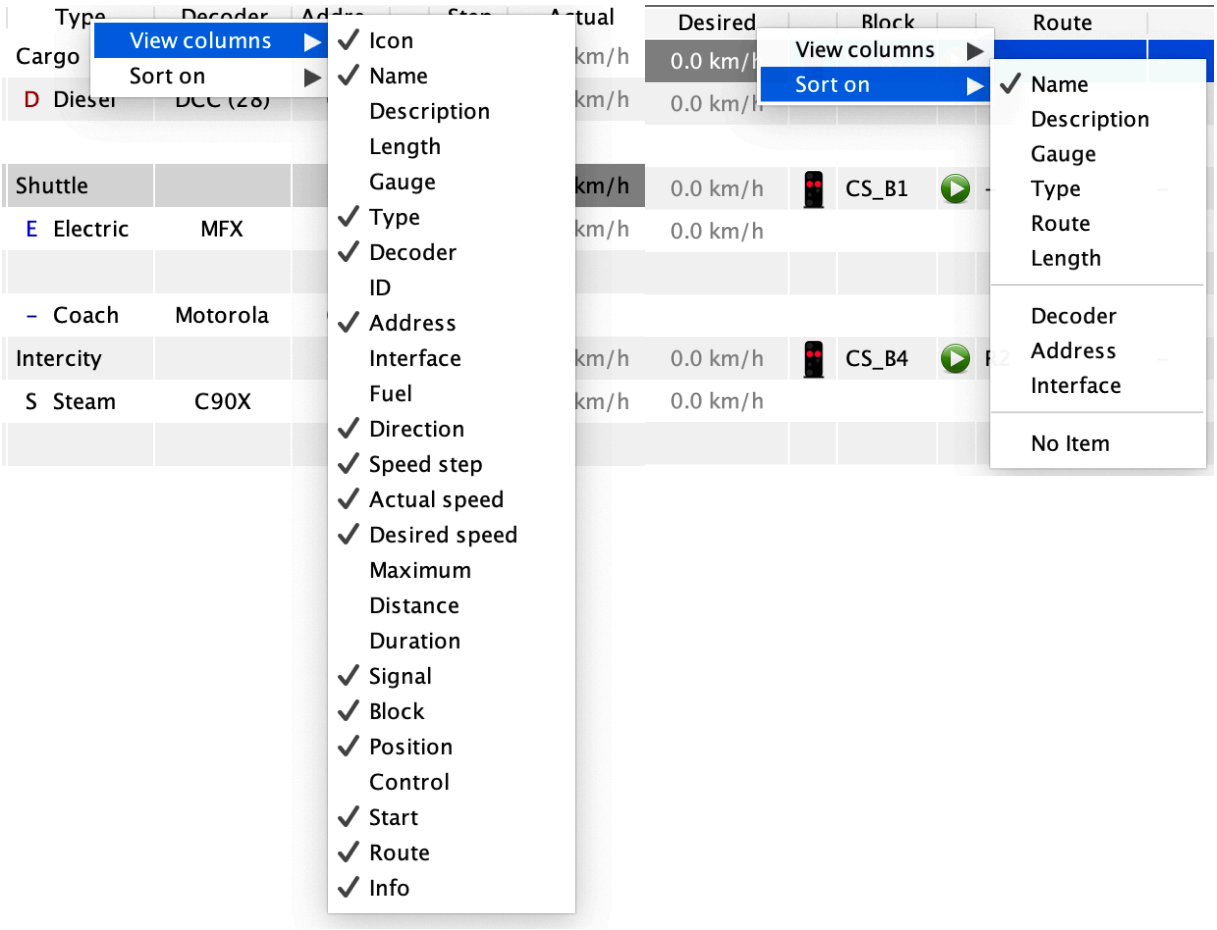


Double clicking again on the open circle will collapse or fold the train. In collapsed state, the train will show some properties that actually belong to the main loc, such as the icon, decoder, address and decoder step that would otherwise not be visible.

Icon	Name	Type	Decoder	Address		Step	Actual	Desired		Block	Position		Route	Info
	DB_CAR	Cargo					0.0 km/h	0.0 km/h		SS_B1	155 cm		-	-
	DB 360	D Diesel	DCC (28)	60		0	0.0 km/h	0.0 km/h						
	< Wagon list >													
	DB_IC	Shuttle					0.0 km/h	0.0 km/h		CS_B1	1,250 cm		-	-
	DB 101	E Electric	MFX	1		0	0.0 km/h	0.0 km/h						
	< Wagon list >													
	IC shuttle	- Coach	Motorola	61		0								
	DB_POL	Intercity					0.0 km/h	0.0 km/h		CS_B4	126 cm		R2	-
	DB 003	S Steam	C90X	3		0	0.0 km/h	0.0 km/h						
	< Wagon list >													

You can also use the keyboard. The cursor key to the right means expand and the cursor key to the left means collapse. When you use the ‘Shift’ key at the same time, it will expand all or collapse all.

The and button are the same as in the ‘Train control’ and one of the two buttons will appear in the column according to what is appropriate for the state of the train.



Which property is shown in the columns of the overview can be changed by the menu ‘View Columns’ of the popup menu of the column headings. It is also possible to automatically sort the trains on some of its more static properties (Name, Description, Gauge, Type, Route or Length), or on some of the properties of the main loc (Decoder, Address or Interface).

To assign a train, loc or wagon of the overview to the ‘Train control’ you can do three things:

- Double click on the ‘Icon’, ‘Name’ or ‘Description‘ column in the table.

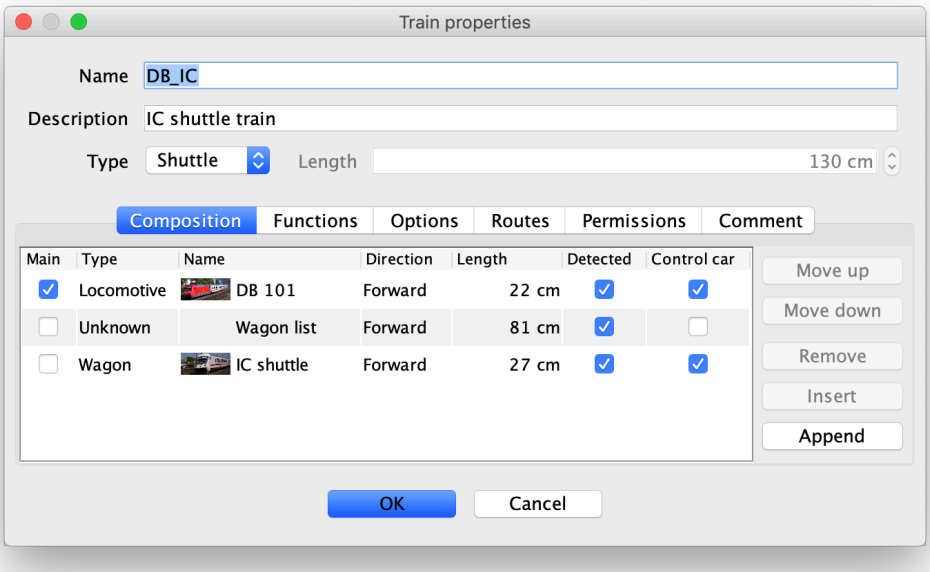
- Select a row in the table and press ‘Enter’.
- Drag from a row in the table and drop it onto the image or its placeholder of the ‘Train control’.

All the key combinations of the ‘Train control’ are also available in the ‘Train overview’. They apply to the selected train, locomotive or wagon in the table. The same popup menu is available to control many things from the overview.

Tip: To use the keyboard, the ‘Train overview’ needs to have focus. To give the ‘Train overview’ the focus, key press F2. You can now use all the key commands. A full list of commands is available in Appendix A.

Composition

In the train overview you can change the composition of the train directly. You can remove an item in the train by using the key ‘Delete’. To add a locomotive or wagon to the end of the train, drop it on a train in the overview and it will be removed from its original place and be added to this train. To alter the order of the locs or wagon in the train, you can use drag and drop in the train overview.



Only to justify the exact direction of a locomotive or wagon in the composition you have to open the train properties via the first item in the popup menu or pressing Shift + ENTER. In the column ‘Direction’ you can check or edit this.

Consist

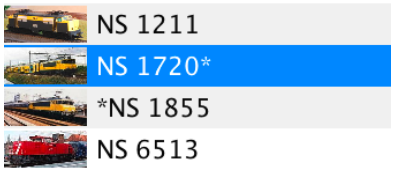
A consist is a combination of locs in one train for extra power. To create a consist just create a train with more than one loc. You only have to take care that the directions of the locs in the train are correctly defined, to prevent that the locs drive in different directions.

Once a loc is part of a train you can use it to control the whole train. So changes to the desired speed and direction are passed on to the train and will affect all locs in the train. The actual speed is always determined by the main loc and its acceleration or deceleration profile. The other locs in the train will follow.

Note: In theory you can add any number of locs to a train, but only locs with more or less the same characteristics will drive nicely together. The program tries to match the real speeds of the locs to the main loc (and not the decoder speeds), but if the

number of speed steps is low it will be difficult to find a nice match and they will not run exactly at the same speed. If the locs are connected to each other, this will in general not be a real problem, but be careful when trying to put wagons between two locs in this case as they might derail in a curve.

In all places where the loc name appears (input boxes, switchboard, etc.), the name of a loc in a consist is rewritten with a number of * symbols before or after the loc name to indicate they run in a consist. So 'NS 1720*' means it is the first loc of a consist of two, and '*NS 1855' means it is the second loc in a consist of two.



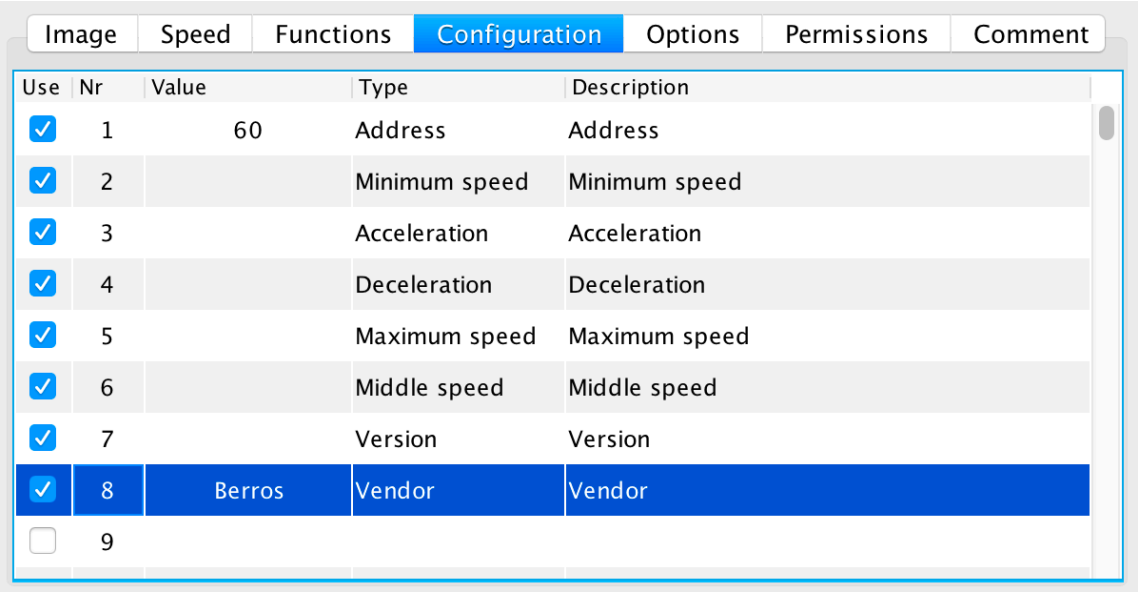
Decoder Programming

Every decoder has a set of configuration variables (for DCC) or parameters (for Selectrix) that define its behavior. One simple example is the address of the decoder, but there are also more difficult settings. A lot of tools are available, sometimes from the supplier of the decoder itself, to assist you in configuring your decoder exactly as you would like.

This feature in iTrain is not here to replace all other tools. It is meant as a basic programming tool for the common settings available in most DCC and SX2 decoders[30] without having to disconnect the interface in iTrain. This can be very helpful when you want to program the decoders on the main layout (PoM)[31] instead of on a service track.

Configuration

The first step in programming a decoder is specifying the configuration. It is possible to specify the configuration for a locomotive, a wagon or an accessory. In this example the locomotive will be used as this is the most common target for programming decoders.



In the properties of the locomotive there is a tab ‘Configuration’ with a table. In this table all possible CV or parameter numbers are available. You have to define which numbers are used to store a value in the decoder (by checking it) and what the type of the value is (by double clicking it and selecting it from a list). Additionally you can enter a description in your own words. It is not necessary to fill in the column ‘Value’ at this time, but if you know the value already it is recommended entering it directly so you need no other administration for that anymore.

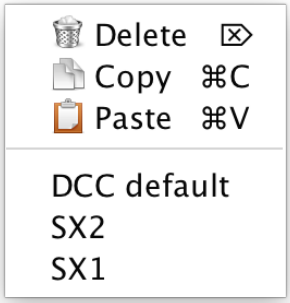
Values are always stored internally as decimal values, but depending on the type they will be displayed in another format if that makes it easier to interpret them (for example the vendor is translated into a name). You can always use the tooltip to see it as a decimal, binary, or hexadecimal value or an ascii character. Entering a value in the column is by

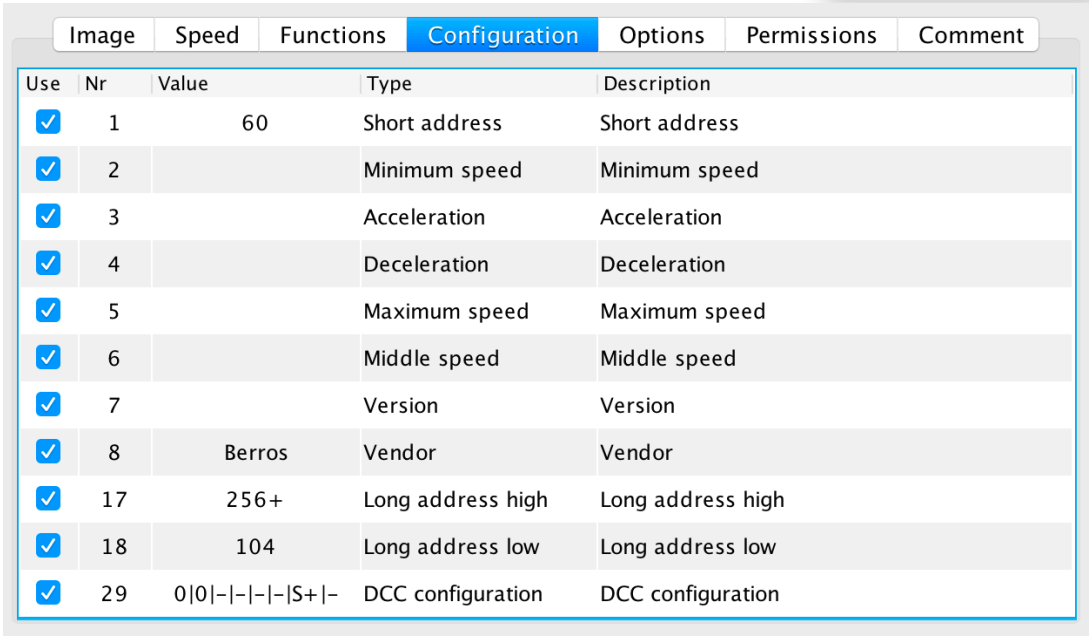
[30] Only DCC and SX2 are supported, because they are very similar. MFX is totally different and not officially documented and SX1 cannot be programmed per value, but only as a set of 5 values.

[31] With new techniques, such as RailCom®, you can even read decoders on the ‘Main’. This is also much faster than with the old methods of reading on a service track.

default always a decimal value, unless you prefix the value with a ‘b’ for a bitmask, ‘h’,‘x’,’#’ or ‘$’ for hex or a quote for a character (so ‘A’ is the same as 65).

It is a lot of work to define the configuration for every locomotive again. By using the popup menu of the table you can make that more comfortable by copying & pasting definitions of other locomotives. Select the rows you want to copy and paste them in another configuration with the same numbers. There are also a few basic templates for the most common variables/parameters for DCC and SX2 to get you started quickly.[32]





Use	Nr	Value	Type	Description
☑	1	60	Short address	Short address
☑	2		Minimum speed	Minimum speed
☑	3		Acceleration	Acceleration
☑	4		Deceleration	Deceleration
☑	5		Maximum speed	Maximum speed
☑	6		Middle speed	Middle speed
☑	7		Version	Version
☑	8	Berros	Vendor	Vendor
☑	17	256+	Long address high	Long address high
☑	18	104	Long address low	Long address low
☑	29	0|0|-|-|-|-|S+|-	DCC configuration	DCC configuration

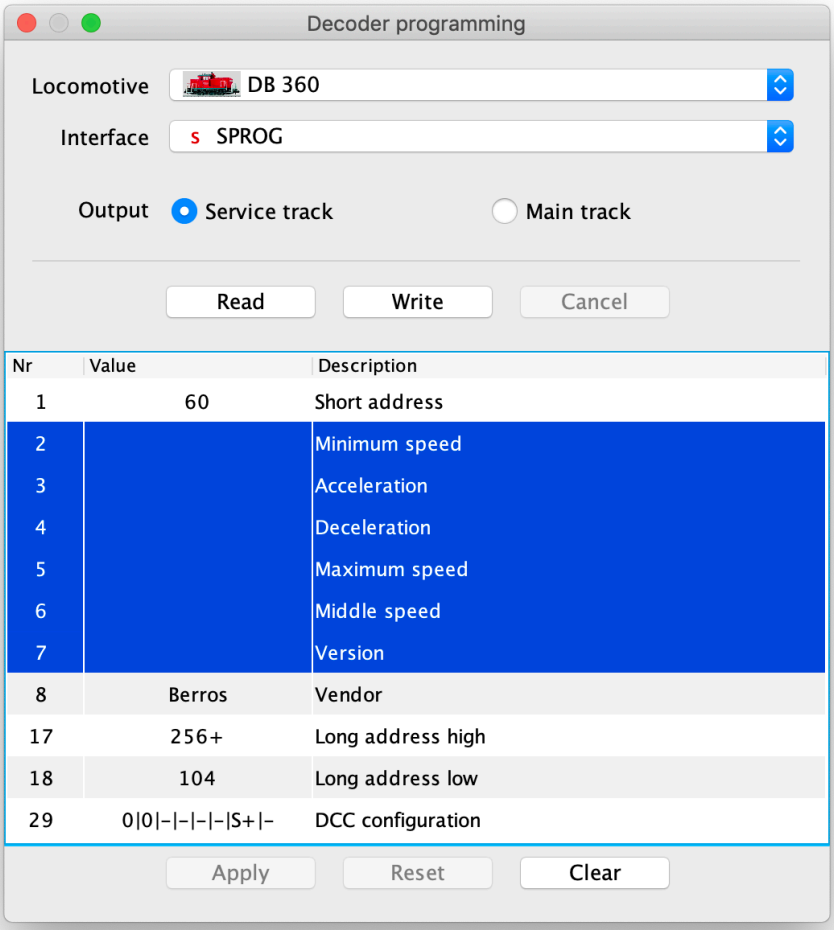
Tip: The list of CV numbers can be large and will probably have a lot of holes (inactive numbers). To remove these holes temporarily to get a better overview of the defined numbers, you can double click in the column ‘Nr’ to filter them out. Double clicking again brings all numbers back again.

Programming tool[33]

If you have defined a configuration for a locomotive, wagon or accessory and iTrain supports programming with your interface (command station or programming hardware), you can actually read/write values via the programming tool in iTrain. This tool can be found in the menu ‘View’ -> ‘Decoder Programming’. There you have to select the type of object you want to program via a submenu. We will explain the ‘Locomotive’, but programming the other objects works in a similar way.

[32] SX1 cannot be programmed from iTrain, but you can store the parameters for administration purposes.

[33] Programming itself is only available in the Plus and Professional edition of iTrain.



First you have to select the locomotive. A table with the configuration will appear with only the relevant numbers. Secondly, you select the interface on which to program the decoder from the list with usable interfaces, and you select if this should be on the 'Service track' or on the 'Main track' (=PoM). Depending on the interface an option may be disabled, because it is not supported, and then there is no choice.

Now you can start to exchange data between iTrain and the decoder by selecting the rows in the table and pressing 'Read' or 'Write'.

Note: You need to have defined the configuration first, as described in the previous section, before being able to read or write anything. This way you know what you are reading or writing. There is no 'Read all values' as this would take a long time on a service track and you would not know how to interpret the values.

Before actually reading or writing values you need to be aware of the following. Writing is an action that is normally fast on both the service and the main track. However, reading is different story. Reading on a service track can be slow to very slow, depending on the technique used. Modern decoders and command stations will use bitwise reading, but some old decoders do not support that and that means that higher values will take longer to read (all values are tried and only if the value is correct the decoder acknowledges it). The possibility of reading values on the main track depends on your hardware, detectors and wiring. If reading on the main track is possible, it can be very fast.

2		Minimum speed
3		Acceleration

When reading or writing the values, a green arrow will appear in the column ‘Value’ meaning that it is busy with this row. If the action is not successful (for example because a timeout occurs), a red icon appears to indicate that the value has not been read/written.

You can edit the table before writing a value or reading a value from the decoder, but the values will only be stored with the locomotive when pressing the button ‘Apply’. In case you accidentally read the values of another decoder, you can restore the original values by pressing ‘Reset’. If you want to clear all values in the table before reading them, you can use the button ‘Clear’.

Special values

The settings for some DCC variables need some extra attention, because they will be used often and are available on almost every decoder:

DCC configuration
A very important value is CV29. It contains the DCC-configuration. This value must be interpreted as a bit mask in which every bit has a special meaning. In iTrain the value will be displayed in a special way (so you can see immediately what it means) by using abbreviations separated by vertical lines, for example `0|0|L|M+|R|A|S+|I`

- `L` the long address as defined in CV17 and CV18 should be used instead of CV1
- `M+` a special motor configuration (table with decoder steps) will be used and not the values in CV2, CV5 and CV6
- `R` RailCom® is enabled
- `A` Analog mode is supported
- `S+` either 28 or 126 speeds steps will be used (recommended) instead of 14 speed steps
- `I` invert the direction in the decoder, so swap forward and backward

For more details about these values you have to dive into the DCC specifications.

Long address
In the past, most decoders had a short address (7-bits) with values from 1-127 (sometimes even restricted to two digits as 1-99). Later on the option to use a long address with 4 digits was introduced. To specify such a long address, two variables are needed. For DCC these are CV17 and CV18 and they need to have the type ‘Long address high’ and ‘Long address low’. To enter the address, you have to make an integer division to split the address into the quotient and the remainder:

`CV17 = (address divided by 256) + 192`	the quotient plus 192
`CV18 = (address modulo 256)`	the remainder

For example, for the address 360 you have to use:

`CV17=1+192=193, CV18=104, (L=1*256+104=360).`

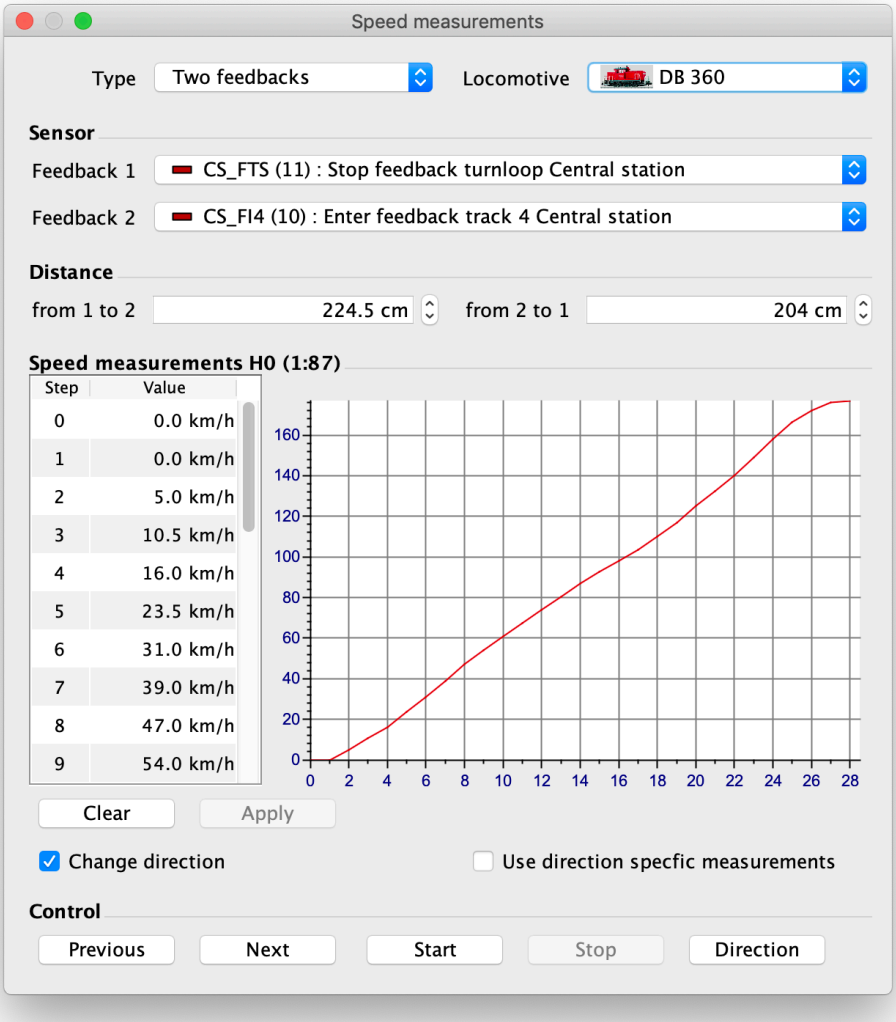
In the table, CV17 will be displayed as `L256+` and CV18 as `104`. The `L` means that you have correctly added 192 which means a **L**ong address, and the `+` at the end means that the next value (CV18) needs to be added to get the full address.

Make sure you set the correct bit in CV29 to activate the long address. Sometimes this happens automatically; so after writing CV17 or CV18 please reread CV29.

Tip: A special trick to do all three steps at once is to type in the letter L followed by the long address such as `L360` *for the value in CV17. This will edit CV17, CV18 and CV29 at once. To use the short address again type in* `S60` *or* `K60` *again for the value in CV1 and both CV1 and CV29 will be adapted to use the short address. CV29 will only be adapted if it already contains a value.*

Speed measurements

In all the settings of speed, a normalized speed in km/h is used. This makes it easier to compare speeds of different locs. By default the program doesn't know which decoder speed step belongs to which real speed and it assumes that the maximum decoder step is 140 km/h and all the steps in between are linearly interpolated. Of course this is rarely the case and you have to measure the speeds of your locs for correct behavior when driving automatically.



To calibrate the speed of your loc, you can use the special calibration tool by going to menu 'View' -> 'Speed measurements' -> 'Locomotives'. The loc in the main loc control will be selected already.

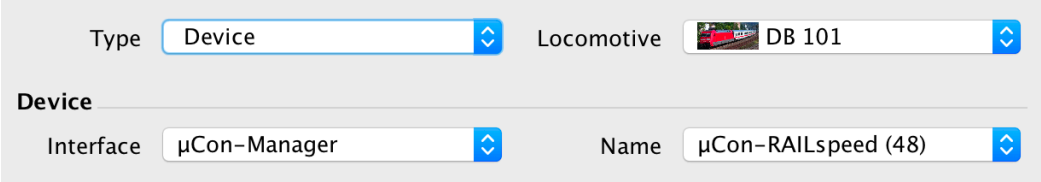
Tip: Before doing speed measurements, it is good to first configure the decoder so that the minimum and maximum speed have a reasonable value for this loc. After every motor configuration change in the decoder, it might be necessary to redo the speed measurements and that takes time.

Method

Speed measurements are all based on a loc running with a fixed speed between two positions and dividing the distance by the time it takes to drive this distance. The value is then corrected with the model scale.

There are three methods in iTrain to measure speeds. You have to select the type you want to use before you can continue.

Device
There are specific devices to measure speed that will be controlled via an interface. These devices are only available while the interface is online. After selecting the type 'Device' a subsection 'Device' will appear in which you can select the interface with a measurement device. In the field 'Name' you can select the available devices for that interface.



The following devices are currently supported:

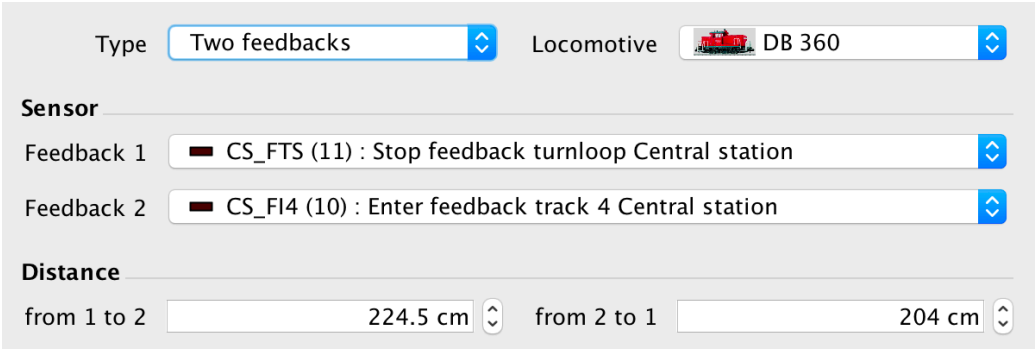
- *µ*Con-RailSpeed in combination with a *µ*Con-Manager or LoDi-Rektor.
- LoDi-TrainSpeed in combination with a LoDi-Rektor.
- RZTec SpeedBox as a separate interface.
- KPF Zeller Speed-Cat as a separate interface.

The first three are devices with two sensors - spaced 10 cm apart - that will calculate the speed automatically when placed parallel to the track. The measurement starts when the device detects a vehicle and it ends when it detects nothing anymore. The last mentioned device directly measures the speed on a roller bank.

The other measurement methods are based on feedbacks in the layout or on a separate measurement track connected to one of your available interfaces.

Tip: Feedbacks have not been described yet, but you can define a feedback via the main menu 'Edit' -> 'Feedbacks' (or Command + F7). For more on feedbacks check out Appendix B.

Two feedbacks
Use two short feedbacks, feedback 1 and feedback 2, with some distance in between. The measurement uses the length of the first activated feedback plus the distance between the two feedbacks.

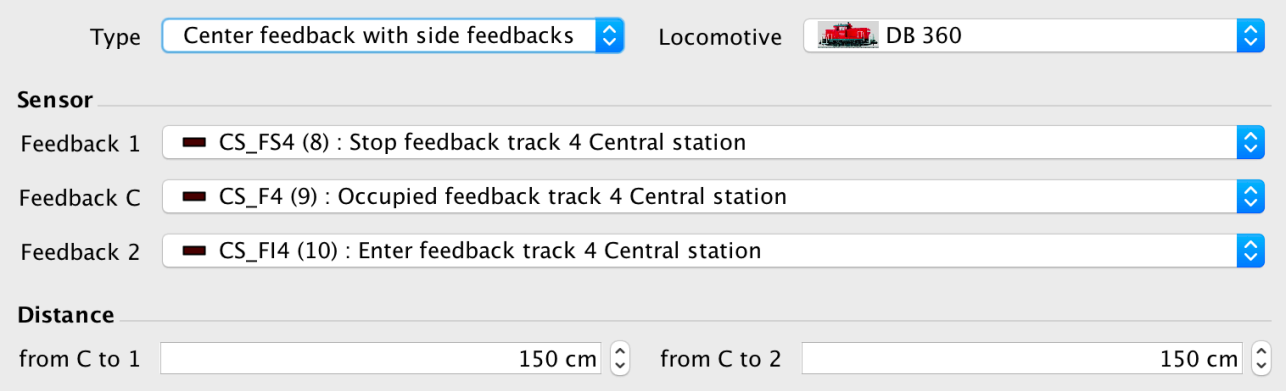


One measurement starts when feedback 1 is activated and is ready when feedback 2 is activated. The loc will continue to run until feedback 2 is released again so that it will be ready for the next measurement in the other direction, starting at feedback 2.

The distance ‘from 1 to 2’ is the length of feedback 1 plus the distance between the two feedbacks. The distance ‘from 2 to 1’ is the length of feedback 2 plus the distance between the two feedbacks.

Center feedback with side feedbacks

Use three feedbacks that are next to each other (it is allowed to have a non-detected part between them). The measurement is mainly done along the center feedback called C.

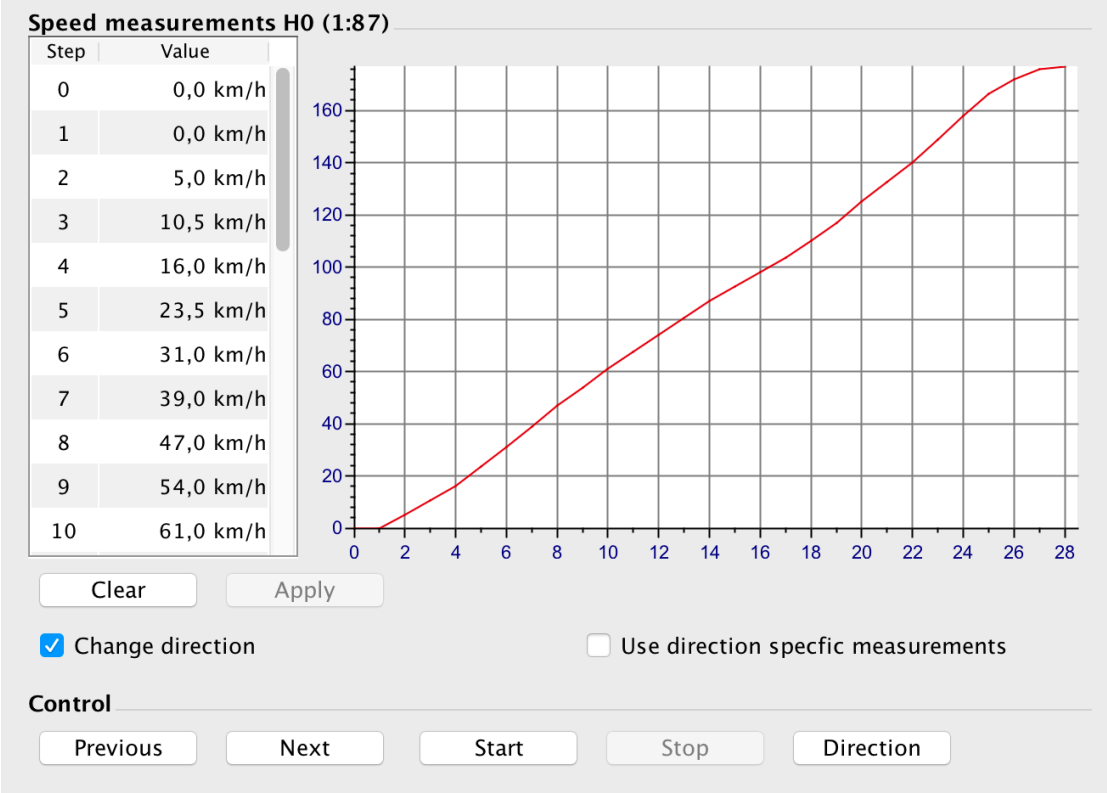


One measurement starts when the center feedback C is activated and is ready when the side feedback 1 is activated. The loc will drive on until feedback C has been released to be ready for the next measurement. The measurement in the other direction uses the side feedback 2.

The distance ‘from C to 1’ is the length of feedback C plus the distance between feedback C and 1. The distance ‘from C to 2’ is the length of feedback C plus the distance between feedback C and 2. In case there is no distance between feedback C and the side feedbacks you can just fill in the length of feedback C for both distances.

Tip: When your layout has one feedback per block the method with a ‘Center feedback’ is preferred to ‘Two feedbacks’. Use the feedback of the block in which you want to measure as the Center feedback, and use the feedbacks of the side blocks as side feedbacks.

Measuring

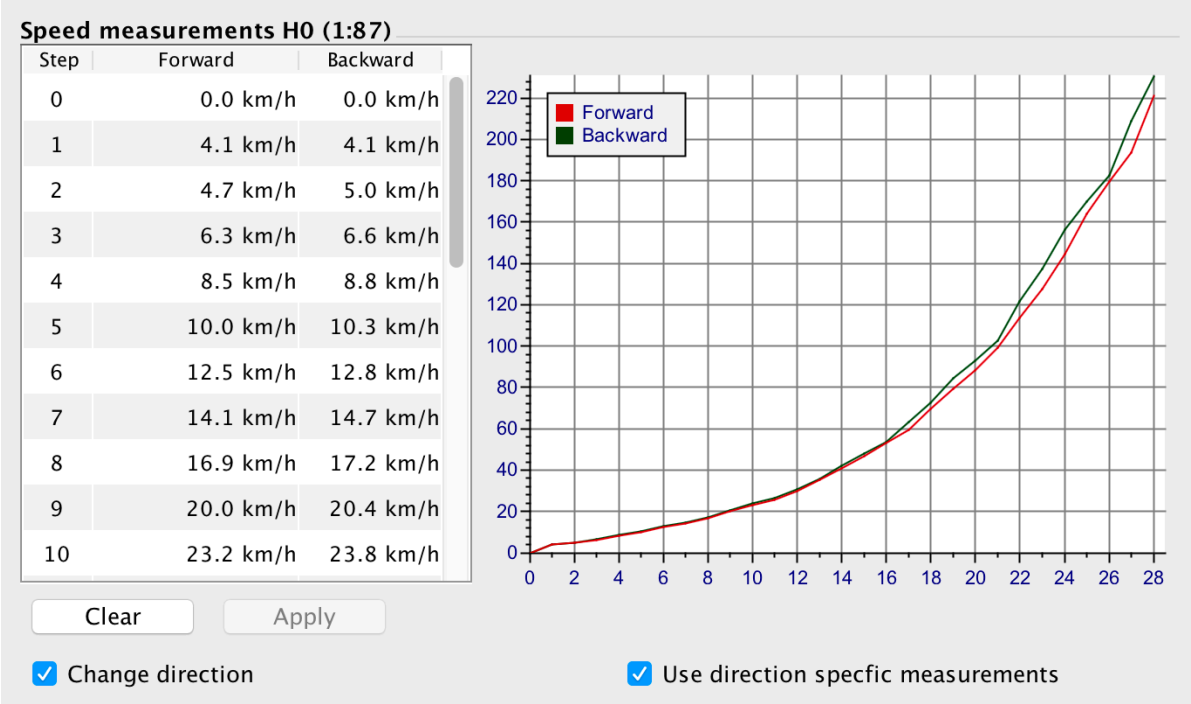


To do the actual speed measurements, select one or more rows with speed steps in the measurements table and press ‘Start’. All buttons will be disabled except for the ‘Stop’ button until all selected steps have been measured. A clock icon will appear in front of the value that will be measured. When the first feedback is activated the icon will change into a green arrow to indicate that measurement of this value has actually started. When the second feedback is activated the individual step measurement has been done and the icon will change into a ‘checked’ icon. The loc will drive on until the second feedback has been released so it is in position for a next measurement. When multiple rows have been selected the next value will be measured until all the values have been measured.

Note: The order of selection is important when selecting multiple rows and determines if the steps are measured from low to high or high to low.

The box ‘Change direction’ indicates that the direction of the loc will be changed between two step measurements. Only uncheck this if you are testing with an oval kind of track in which the loc will be running in the same direction for all measurements.

If a loc has substantial different speed measurements in both directions, it is also possible to use direction specific measurements. In that case an extra column will be added to the ‘Speed measurements’ table to separate the forward and backward speed measurements and there will be two lines available in the graph (see the legend).



Note: The measurements are first done in the window and are only applied to a loc if the 'Applied' button is pressed. Use the 'Clear' button to clean the table before doing measurements if the speed settings of the decoder have changed.

At any time during the measurements you can press 'Stop' to abort them. The 'Direction' button changes the direction of the loc in case that is necessary before starting a measurement. When doing the speed measurements one at a time, you can use the 'Next' or 'Previous' button to measure the speed for the next or previous decoder step. The direction is automatically changed and the speed of the loc will be set correctly.

Note: Cells can also be edited manually. To clean a value, you have to select a row and press the 'Delete' key.

It is not always necessary to measure all steps, and values can be left empty or zero. Always measure the first step for which the loc starts to move. This is the minimum step. All decoder steps before the minimum will not be used by automatic control and are considered idle steps. The last step with a value greater than zero is considered to be the maximum step. All steps that are left empty or zero between the minimum and maximum speed will be interpolated as soon as the 'Apply' button is pressed.

Note: All speed measurements should increase with every step to make them useful in controlling the loc. You can check this easily in the graph. If the values are not increasing, they will be sorted after pressing 'Apply' and after the optional interpolations.

Note: The loc function 'Direct control' will be activated when doing speed measurements to disable the inertia simulation in the decoder. The loc will get to the measured speed faster and it will brake faster giving a more precise measurement and taking up less track space for high speed measurements. Take care that the 'Direct control' function doesn't reduce the speed. In that case change the type of the function to 'Slow'.

Switchboard

The switchboard is intended to control all the switches in your layout and to see what is happening with every train. As every layout is different in size, the switchboard has some ways to handle a large layout on the screen:

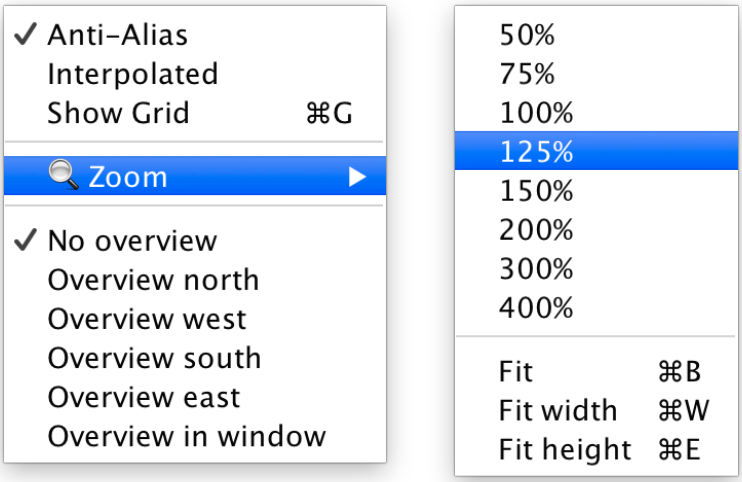
1. Multiple tabs to show different parts of your layout directly.
2. Separate zoom control per tab.
3. An optional overview per tab.
4. The option to use a 'Wide' layout view.

Before the details about creating a switchboard are explained, we first explain how to use a switchboard. You can try it with the demo layout (`demo.tcdz`).

Zoom

To zoom in on the switchboard, there are several options:

1. Use the zoom box or the zoom buttons on the toolbar (on the upper right corner).
2. Use the scroll wheel of the mouse in combination with the Control or Command key to zoom in or out at the location of the mouse pointer in the grid.



3. Use the popup menu via the right mouse button (or use Control + click) and go to the 'Zoom' sub menu.

The grid on the switchboard can be switched on and off via the popup menu (shortcut Command + G).

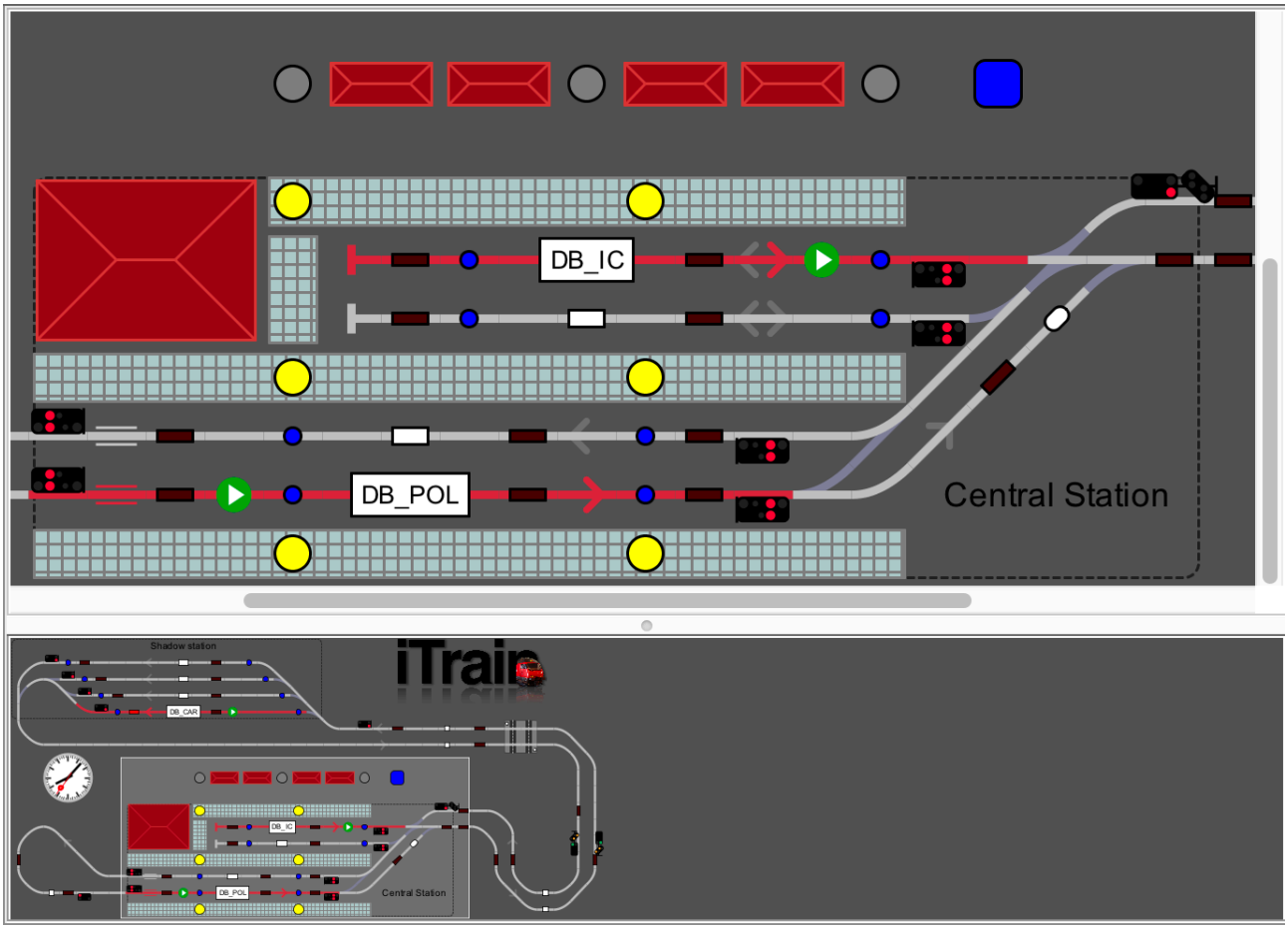
Quality

The option 'Anti-Alias' should always be switched on unless your graphics processor is quite old and slow. It will make all drawing more smooth. It is checked by default.

The option 'Interpolated' enhances the quality of imported images in the switchboard when being scaled. It will degrade the drawing performance and is only recommended on systems with very good graphics performance. It is unchecked by default.

Overview

This overview option in the popup menu of the switchboard adds an extra overview pane at one side of the switchboard (for example at the bottom with ‘Overview south’) or in a separate window.



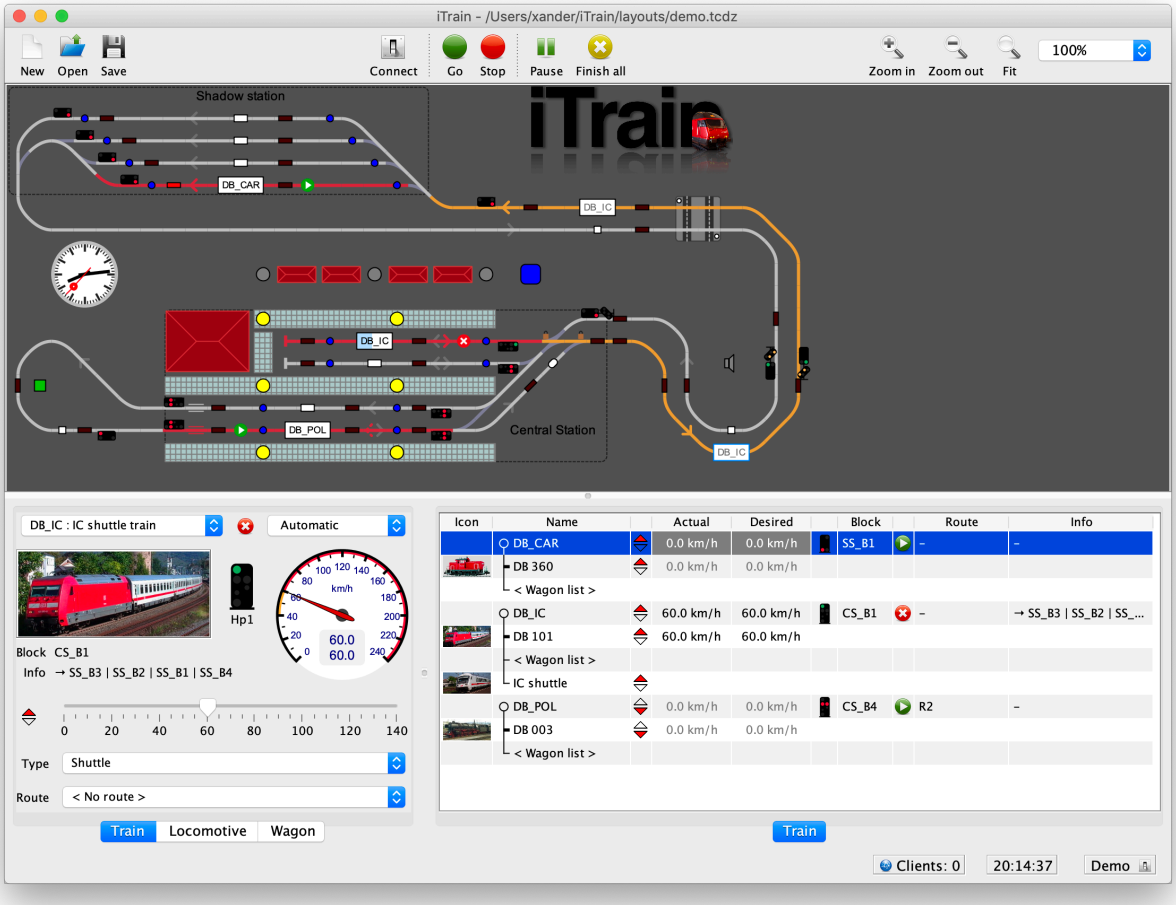
In the overview the whole switchboard is shown with a rectangular box that shows which part has been zoomed in. This is very useful if your layout is quite big and you want to control only a part of it, but at the same time keep an eye on the whole and be able to switch to another part very easily.

There are two ways to manipulate the visible part in the switchboard via the overview:

- You can drag the box in the overview to move the zoomed in area.
- You can select an area in the overview with the mouse (starting with a selection outside the current box, to prevent moving or using the ‘Shift’-key) to show what should be shown in the switchboard. After selecting an area it might change to keep the aspect ratio of the switchboard space.

Layout View

Next to the standard layout there is also a wide layout where the whole width of the window is used for the switchboard and the loc controls are put below the switchboard. This makes more space for all the columns in the ‘Loc overview’ and is also useful for layouts that are very wide but do not need the full height of the screen. It is available via the menu ‘View’ -> ‘Layout’ -> ‘Wide’.

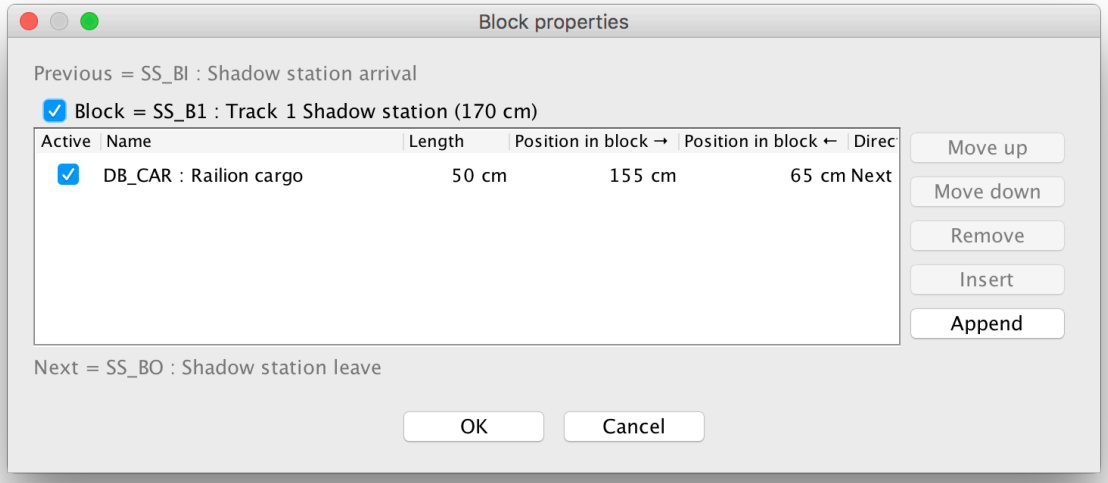


Controlling Switches

Every switch (an accessory like a turnout, a feedback or a track route) can be (de)selected or changed by single clicking it. For different objects the behavior might be slightly different:

- For turnouts it will change from `straight` to `branch` or the other way. A special case is a three way turnout that is actually considered as being two turnouts that cooperate. If the state of one of the two turnouts is `branch` it will go to `straight`. If the state of both turnouts is `straight` the change depends on which side you click and it will only change this side of the turnout.
  In case you want to select a state directly, just press down the mouse button on the turnout until a popup with all the states appears, and select one.

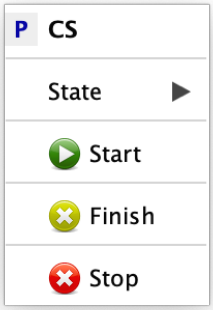
- Signals with two states just change their state. In case of more than two states, the signal will go always to red unless it is already red, then it will go to green or the third state (mostly yellow) when using the Shift key. Combined signals are actually two signals and depending on which part is clicked they will change accordingly.
  In case you want to select a state directly, just press down the mouse button on the signal until a popup with all the states appears, and select one.
- A relay will toggle its state `on/off` or `red/green`.
- Decouplers will be activated as long as the mouse button is held down and be deactivated when the mouse is up again (if the underlying system supports this and else they will be activated for the configured time period).
- A light will switch `on` and `off` when clicking. A light with a level can be dimmed by pressing and holding the mouse to change the level.
- A sound will `start` or `stop` when already playing.
- An aspect will loop forward through all states (`aspect0`, `aspect1`, etc.) or backwards when using the 'Shift' key.
- A crossing will toggle its state `open/closed`.
- A turntable will rotate to the clicked position. More details are described later on.
- A transfer table will shift to the clicked position. More details are described later on.
- A train magazine will shift to the clicked position. More details are described later on.
- Track routes (described later) are activated and deactivated indicated by their icon, but they can only be activated if they don't conflict with reservations already made on accessories in the track route.
- Feedbacks are toggled by clicking when the control system is offline, but if your control system is online it will set the feedback according to the state read from the interface and you cannot change it.
- Boosters will toggle its state `on/off`.



- Blocks cannot be switched. By double clicking the block a dialog pops up to select the trains that are in the block and to specify where they are in the block.
  One of the trains is called the 'Active' train, and for this train the properties (name, state,

direction) will be shown in the switchboard for this block
First the ‘Previous’ block is specified in gray, followed by the current block and its length and the trains in the block in order of the definition of the block (following the arrow from ‘Previous’ to ‘Next’) and finally the ‘Next’ block that follows after the block. For every train the length of the train is specified, followed by its position when the direction of the train is towards the ‘Next’ block. In addition, the position is specified when the direction of the train is towards the ‘Previous’ block and finally the current driving direction of the train (towards the ‘Next’ or ‘Previous’ block). You can choose to enter one of the positions and then the other will be calculated automatically.
You can deactivate the block by deselecting the checkbox before the block name so that it is not used in automatic route control, and it is visually disabled in the switchboard.

- Direction arrows indicate in which direction the active train in a block is going. When you click on it the train is turned and the arrow will be reversed. If the train direction on the track does not match the direction of the arrow, you can Shift + click on the arrow. By doing this the arrow will be reversed without changing the direction of the train itself so that they are synchronized again. It has the same effect as choosing another direction in the dialog above.



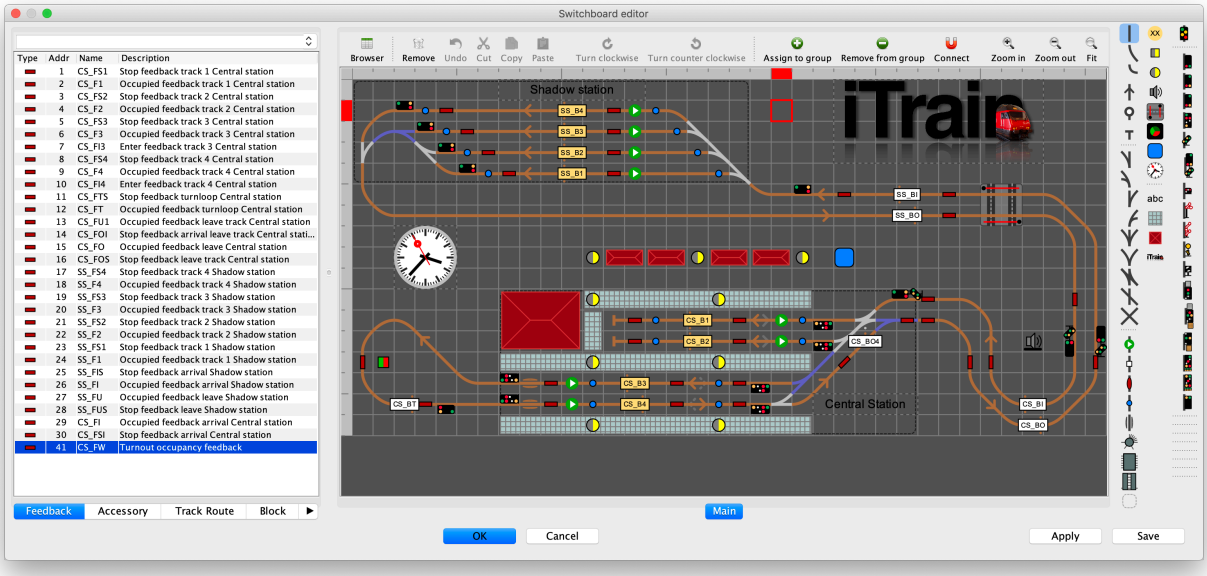
- A station is a collection of blocks that can hold trains. By pressing down the mouse button until a popup appears you can change the state or execute actions for all trains in the station. The actions have been described in the chapter ‘Trains’.

You can also attach a train to a block by drag & drop. You either select a train from the ‘Train control’ or ‘Train overview’, or you select a train from the switchboard itself and drop it onto a block. A popup menu will appear and you have to select ‘Set’. When you are holding Alt (or Control) while dropping the loc or train, it will be set directly without showing a popup menu. The direction of the train is set to the preferred direction and can be adjusted, if necessary, by clicking on the direction arrow of the block.

Tip: Per switchboard tab every switch can be assigned a key combination to simulate the mouse down/up with a key press/release (see key mappings later on).
Use the F4 key to give the switchboard focus so that key events are not absorbed by other parts of the application.

Create or edit the switchboard

To create a new switchboard or to edit a current one, select from the menu ‘Edit’ -> ‘Switchboard’ or press ‘Command’ + F4. If you do this for the first time you will be asked to enter a name for the first tab. You can modify it later, so if you don’t know or only need one tab, just enter ‘Main’.



You will now see the above screen. The switchboard is at the centre. At the top you see a toolbar with some actions. On the right side is the toolbar with three columns with all the elements you can put on the switchboard. On the left side there is an optional ‘Browser’ with all objects currently defined in iTrain that may be or are attached to the switchboard. In the grid you see the cursor shown as a red rectangle that you move with the cursor keys or by clicking in the grid with the mouse.

The ruler on the left side and upper side of the grid reflects the position of the cursor. This is useful if you cannot find the cursor directly in a large project. By dragging the end of a ruler, you can enlarge or shrink the grid in this direction. You can also use the ‘Alt’ key in combination with the arrow keys: ‘Left’ or ‘Up’ to decrease the grid size in horizontal and vertical direction, and ‘Right’ and ‘Down’ to increase the grid size in horizontal and vertical direction.

To draw elements in the switchboard the preferred way of working is to select an element on the right (by mouse or better with the keyboard), turn it into the right position and then go to the switchboard and press ‘Space’ to add the element. You can continue by selecting another cell with the arrow keys and pressing ‘Space’ again. Elements can be rotated in the drawing as well as on the element toolbar.

An alternative method is double clicking the element on the element toolbar to add an element to the switchboard. However, this slows down the drawing process when having to add multiple elements, because you continuously have to change the focus between toolbar and switchboard. Try to learn some of the key commands and you will be able to draw your layout very quickly. In Appendix A all the key commands available are described.

When using the keyboard to enter commands, the focus is important. The best practice is to keep the focus on the switchboard (highlighted cursor) and select elements from the toolbar via key combinations. When you are holding the ‘Control’ or ‘Command’ key you are navigating through the toolbar:

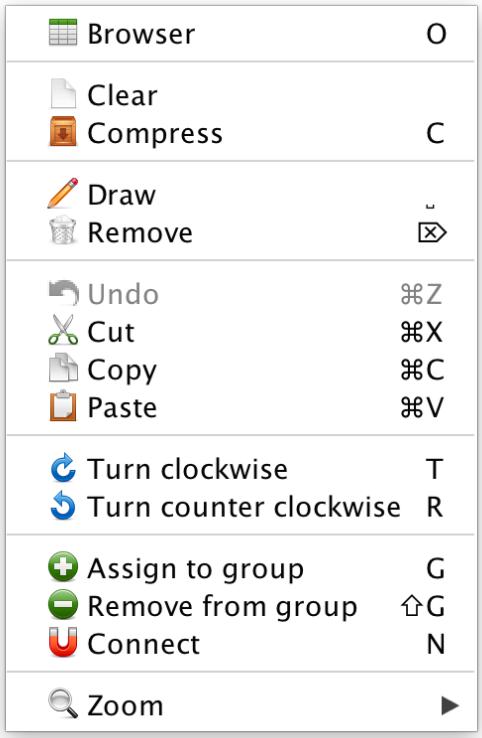
- Cursor Up/Down to select the items on the element toolbar.
- Cursor Left or ‘R’ key to turn the selected element on the toolbar counterclockwise in the preferred direction.

- Cursor Right or ‘T’ key to turn the selected element on the toolbar clockwise in the preferred direction.

From the switchboard you can now easily edit the grid:

- Cursor keys to move the cursor in the grid.
- ‘Space’ key to add an element from the toolbar to the switchboard.
- ‘R’ or ‘T’ key to turn an element in the switchboard counterclockwise or clockwise.
- ‘Alt’ + cursor keys to change the size of the grid.
- ‘Shift’ + cursor keys to select an area of the switchboard.
- ‘Shift’ + ‘Alt’ + cursor keys to change the size of elements that can cover multiple cells.
- ‘Shift’ + ‘Control’ or ‘Command’ + cursor keys to move a selected element or area over the switchboard.
- ‘Delete’ or ‘Backspace’ key to delete an element. Only the element on top will be deleted if one cell is selected. If the selection contains multiple cells, all layers will be deleted.
- ‘Shift’ + Delete or Backspace key to delete an element. Only the element below will be deleted if one cell is selected and two elements are drawn on top of each other. If the selection contains multiple cells all layers will be deleted.
- ‘Command’ + ‘X’ to cut the current selection and copy it to the clipboard so that it can be pasted later.
- ‘Command’ + ‘C’ to copy the current selection to the clipboard so that it can be pasted later.
- ‘Command’ + ‘V’ to paste the elements on the clipboard at the cursor. If the elements do not appear where they should do, you can move them again with Shift + Command key + cursor keys or delete them with the Delete or Backspace key to undo the paste operation.
- ‘Command’ + ‘Z’ to undo a move, a cut or a delete. Once the selection changes, you cannot undo it anymore.

Once an area (more than one cell) of the switchboard has been selected, you cannot use the curve and turnout elements from the toolbar anymore. Select a single cell to be able to use all elements again.



Tip: The popup menu (click right mouse button) of the switchboard in edit mode contains some of these commands (with their key equivalents). The same zoom functionality is available in edit mode, only without the overview function.

Toolbar

The toolbar on the right may contain more elements than your screen can display. The elements are grouped. Via the popup menu (right click) you can select which groups are visible.

As the space is limited, by default only the German signals are visible, as these ones are mostly used on model railways. But you can change that by selecting another signal group and deselecting the ones you don't need.

Once the focus is on an element in the toolbar, the following key commands can be used:

- Cursor Up/Down to select the elements.[34]
- Cursor Left or 'R' key to turn the selected element counterclockwise in the preferred direction.
- Cursor Right or 'T' key to turn the selected element clockwise in the preferred direction.

[34] With respect to navigation the three columns should be regarded as one long column with the second column coming below the first column and the third column at the bottom.

The track elements are to draw the track. The first element is the straight track and the second and third are both a curved track. The fourth element is the arrow that indicates the direction the train is driving. Add at least one per block if possible. The fifth element is a link element, in case a track is continued on another position or tab. The sixth element is for a dead end track.

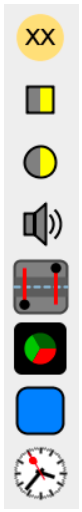
Note: If the train is allowed to drive in both directions, just select one direction or select the preferred direction, but do not put two arrows in the opposite direction. In the block properties (explained later), you can define which directions are allowed in the block and if necessary the element will change into a double arrow automatically.

The turnout elements are a preselection of most of the possible types of turnouts. You can change the type later on by double clicking the turnout and selecting a type (so single slip is also possible). You cannot change the angle of the cross elements afterwards, so only for 90 degrees crosses select the last element.

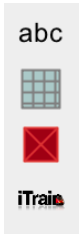
The ‘Others track’ are other elements that are drawn unto a rail element. The first element is a route button to control routes within the switchboard. It will only be visible if there is a train in the block. The second element represents the block itself. This block element can be used to change block properties and it is used as the display of the block to show a name, speed and waiting time and it may cover more than one cell. In edit mode it will show the name of the block if the zoom level is 125% or more. The third is a feedback and can be a reed contact, occupancy feedback or manual button on your layout or control panel. The fourth is related to the track route and can activate and deactivate a manual track route. The fifth element is the decoupler track to split trains manually. The sixth, seventh and eighth element (with a lot of track connections) represent a turntable, a transfer table (or traverser) and a moveable train magazine. The last element represents a station that is modeled as a rounded selection rectangle covering the elements that belong to a station.

Tip: The block element grows dynamically and may cover more than one cell when a name appears into it. Take this into account when drawing your layout, and do not put other non-track elements next to it that may be covered and will not visible anymore at that time.

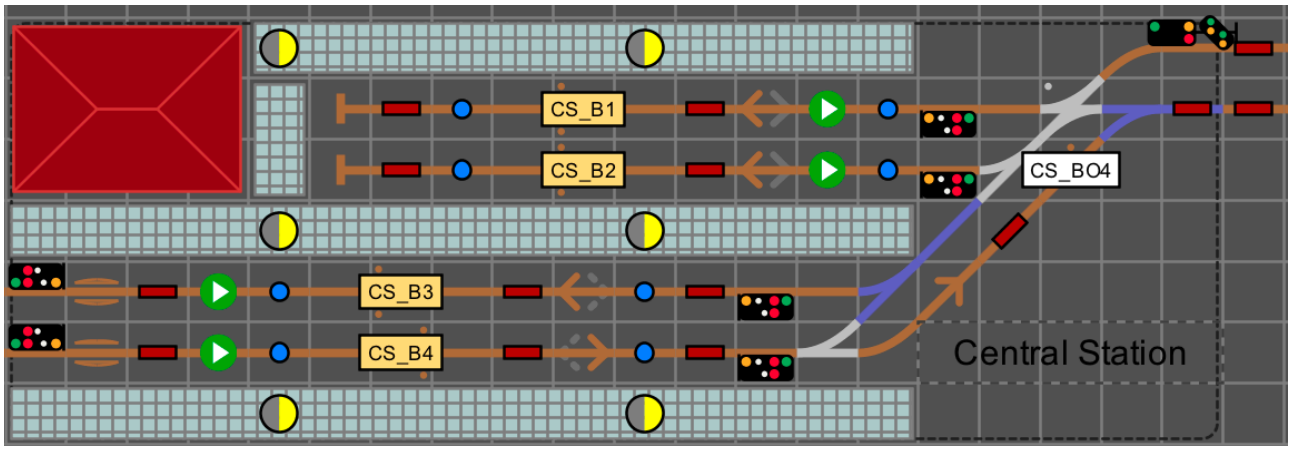
The ‘Control elements’ group are other elements that are not drawn with a track element. The first element represents an aspect with up to 32 states that displays the value of the aspect (0-31) in the element. The second element represents a relay with two states: `On` (=yellow) or `Off` (=gray), or `State A` (=green) or `State B` (=red), depending on the type. The third element represents a light with a level (on/off or dimmable) and optionally a color. The fourth element represents a sound on the computer or a sound decoder in the layout. The fifth element is a crossing for railroads or a bridge that can be `open` or `closed`. The sixth element represents a booster that can have three states: `off`, `on`, `alarm`. The seventh element represents an event action and the last element is an analog model clock.



The ‘Draw elements’ group contains elements to enhance the switchboard. They can cover multiple cells in both directions and the best way to put them on the switchboard is by first selecting an area and then adding the element.
The first is for adding text to the switchboard. The second is to draw a platform next to a track, for example in a station. The third element can be used to draw buildings or houses (schematically).
The fourth element is for inserting external images. Images will impact the drawing performance of the switchboard so it is recommended to keep them small. Do not use megapixel images of your photo camera, but downscale them first with another program. When no external image is selected (empty file name) the iTrain logo will be shown.

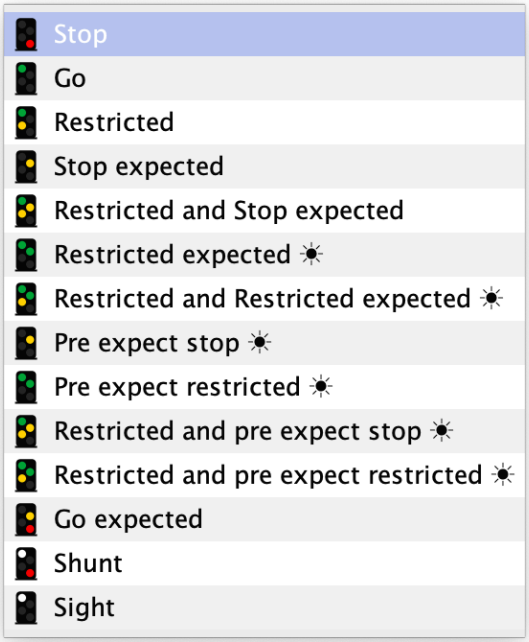


Tip: To change the size of ‘Draw elements’ in the switchboard, you have to put the cursor inside the element and use Shift + Alt + cursor keys to change the width and height. To always stay within the element when making an element smaller, it is best to put the cursor in the left upper corner of the element.



Signal elements
There are multiple element groups for the signals from different countries, but at first there is a ‘General’ signal which does not represent any national railway system. It supports all states that are used internally by iTrain and can be used if no other specific signal matches your needs. However the number of states can be overwhelming so you normally do not

want to enable all states.[35] The symbol ☀ means that some lamps are blinking.



The supported countries and signal types are:

- German - the H/V form and light signals of the DB, the Hl light signals of the DR and the new Ks signals including the ones with a display
- Swiss - the L type lights signals and the N type light signals with display of the SBB
- Austrian - the signals from 1954 and 1980 of the ÖBB
- Dutch - the light signals of the NS including the ones with a display
- Belgian - the light signal of the NMBS including the ones with a display
- French - the light signals of the SNCF
- British - the light signals
- American - the search lights as a combination of the BNSF and the Union Pacific
- Canadian - the search lights

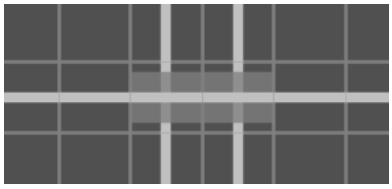
Just as with the turnout elements, the shown signal elements in the toolbar are just a preselection and more types are available by putting the signal on the switchboard and changing the type. The signals are drawn on or drawn next to a straight track depending on the country and signal type.

Note: Signals belong to the block in which the train will stop and not to the block they are protecting. This is a more practical decision to make it easier to define a situation where blocks are connected by a turnout and the signal is placed before the turnout, but is protecting the block after the turnout.

[35] Especially the states with 'Pre' in it look two blocks ahead and are normally not necessary. 'Go expected' is only for cars.

Layers

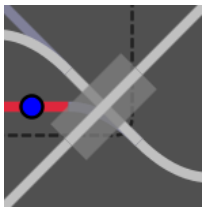
There are no specific elements for tunnels and bridges, because you can draw elements on top of each other. When you draw a new element over another element in the same direction, then it will replace the current element. If you draw it in a different direction, then it will be drawn on top of another element with a kind of transparent gray bridge element.



If only a single cell has been selected, then actions like delete, move, copy will act on the element on top. If more than one cell has been selected, the actions will act on all layers.

Tip: In case you see the gray layer drawn on some elements, but there is no purpose for two layers, you probably have put two elements on top of each other, but afterwards turned the element on top in the same direction as the one below. This is an error and may give unwanted results later on. You can correct it by selecting the cell and press Shift + Delete to remove the element below. Now the gray layer should disappear.

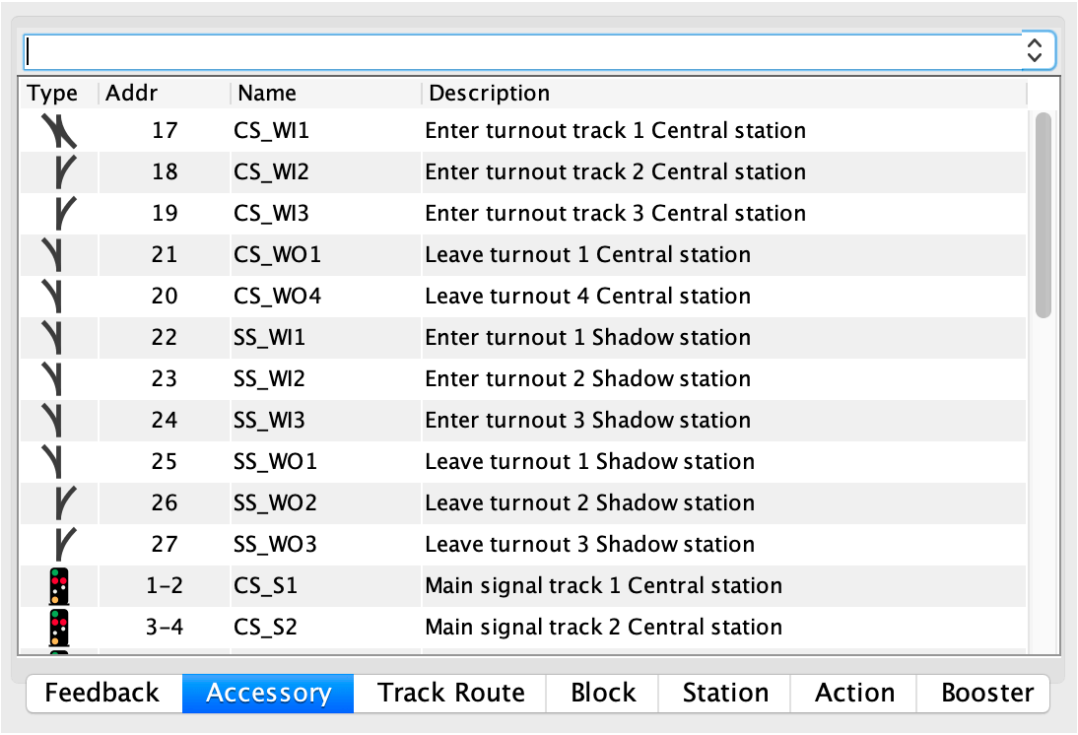
When you control the layout, it is no problem to switch an element (for example a turnout) that is below a normal track element. Only the top most switchable element is switched and so non-switchable elements on top of switchable elements are not a problem, but prevent two switchable elements on top of each other if you want to be able to control them manually.



Browser

The browser on the left gives on overview of all the control objects defined in iTrain that can be or are attached to elements in the switchboard. You can assign control objects to a switchboard element by drag & drop. Normally, you immediately create a control object together with the switchboard element, but sometimes control objects are already available, but not attached to switchboard elements, for example, when they are already retrieved from your command station (ECoS or CS1).

Tip: The browser can be hidden by using the 'Browser' button on the toolbar to get more space for the switchboard. Hitting the same button again will bring up the 'Browser' again.

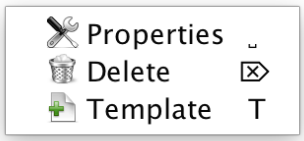


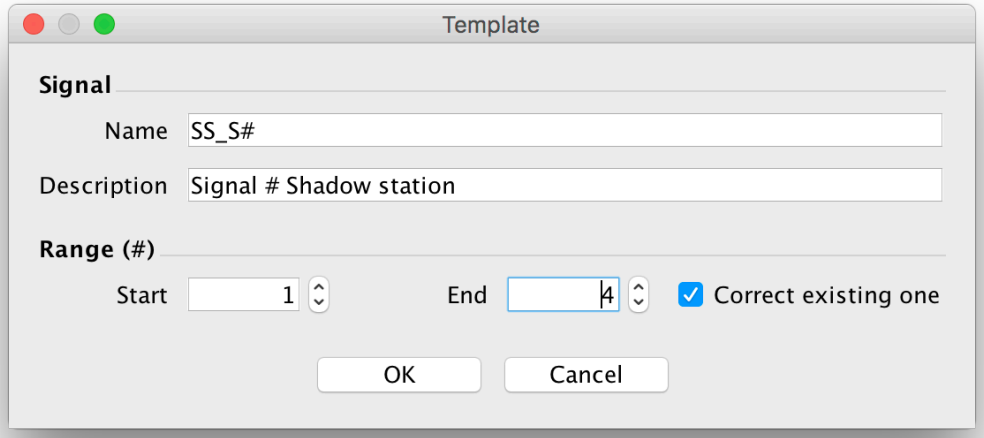
In the browser you can rearrange control objects by drag & drop within the browser or you sort the objects by name by double-clicking on the column header. You can also remove control objects that are of no use anymore, but be careful with this, because once deleted you cannot restore them anymore. Removal of an object can be done via the popup menu (right click) or via the ‘Delete’ or ‘Backspace’ key.

To select an object in the browser via the switchboard, you have to use Command + double-click on the element in the switchboard, or use Command + ‘Enter’. To find and select an associated element in the switchboard that is attached to an object in the browser, you can just double-click it or select it and press ‘Enter’ in the browser.

Template

You can also create new control objects based on other control objects via the ‘Template’ action in the popup menu of the browser (right click). This is useful if you have to create multiple objects (for example turnouts or feedbacks) with almost the same name and description, but only with another number in it.





You first select a control object from the browser that is used a base definition. Then you select 'Template' or you press 'T'. In the presented dialog, you put a '#' character where the number should be inserted and in the 'Range' section you define the numbers that will be generated. In the example, 4 signals will be created with the names SS_S1, SS_S2, SS_S3 and SS_S4 and with a description with the same number in it. Other properties of the base object will also be copied except for the ones that are naturally different, such as address, comment and other things that are unique for every object, such as length.

If the 'Correct existing one' flag is checked, then also existing control objects will be changed so that they now have the same properties as the reference.

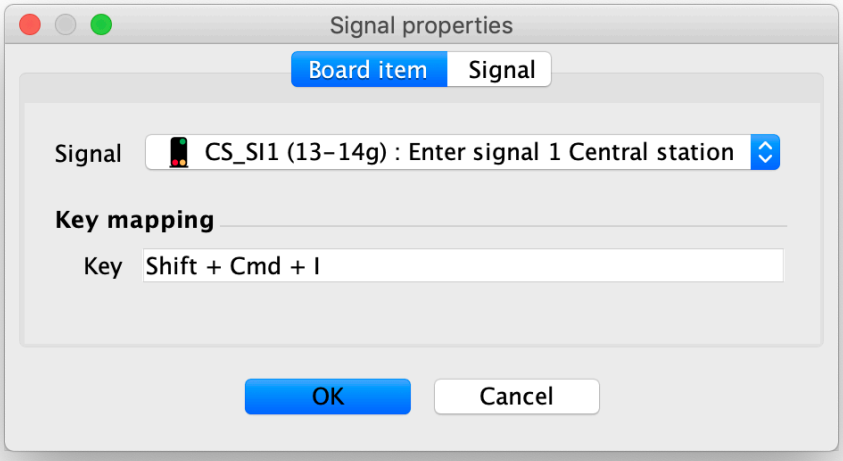
Tip: The 'Template' action can be used to create many turnouts, signals and feedbacks that are almost the same, such as in a shadow station. You only have to add the address manually. Even if you have already created these more or less equal objects manually, you can use 'Template' to assure that.

Assign control objects

In the switch board you can draw turnouts, signals, feedbacks and so on, but these drawing elements must be attached to their underlying control objects. For every control object there is only one object identified by its name, but it can be drawn multiple times on the switchboards, for example in different tabs.

You can create and/or attach the control object to its viewing element in the switchboard by double-clicking the element. A dialog with two tabs will be shown:

- The first tab shows the properties of the board element or viewing element. You can select a control object from a list with already defined objects that you want to attach to it, or select the first entry starting with '<No...' if you want to define a new control object on the fly. If you change the selection it will automatically select the second tab to edit its properties.
- On the second tab you can change the properties of the selected control object or add the data of a new control object.



For example, when you double click a three state signal element, the 'Signal properties' dialog shows up. You have to select one from the drop down box (for some interfaces a list may already be available from your command station) or go to the second tab and enter a new signal.

Key mappings

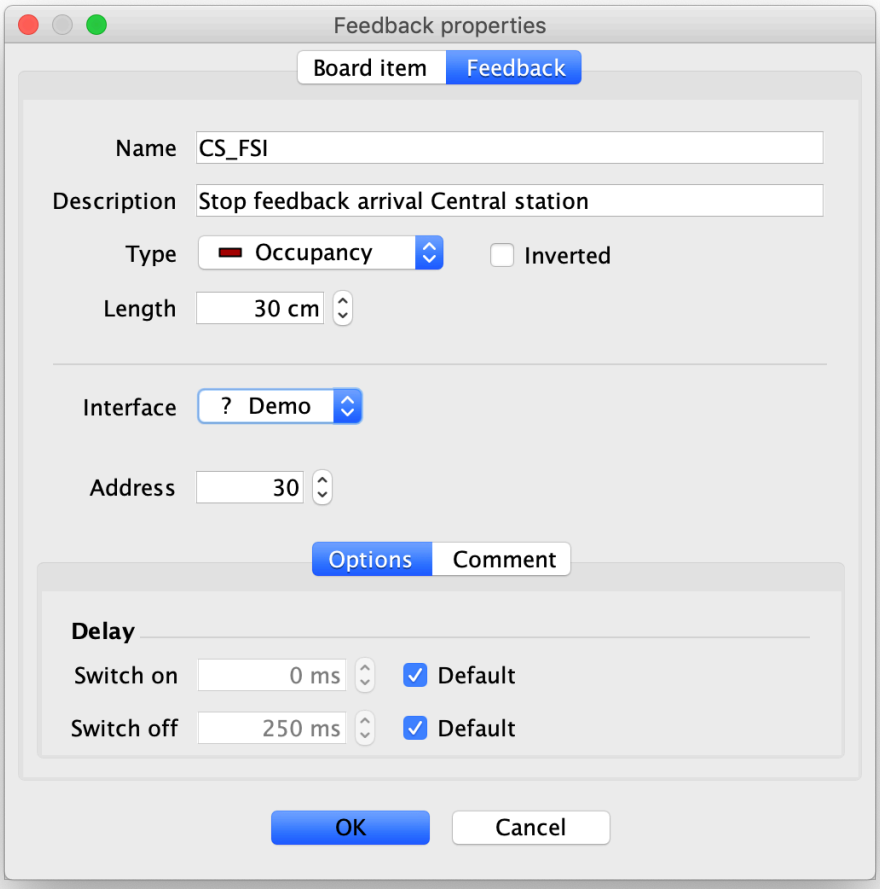
To assign a key combination to an element, you can use the key mapping feature. Just select the 'Key' box and use the key combination you want to use, including modifier keys like 'Shift', 'Control', 'Alt' and 'Command'. The text starting with 'Code' should change to represent the pressed key combination. This key combination is only valid on the tab of the switchboard where you assigned it, and doesn't belong to the control object itself. This way you can reuse key combinations on different tabs, for example for different stations.

Tip: Make sure you do not override an already existing system or program key combination, as you might get unexpected results. The 'Escape' key is to remove an existing key combination and cannot be used in key combinations, because it is reserved to 'Stop' the system.

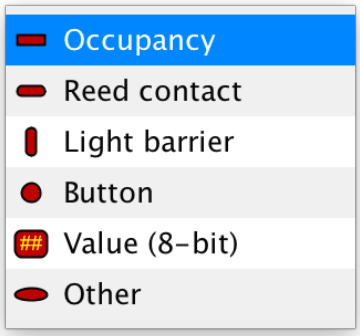
When you press 'Ok' on the dialog, the changes on the control object are stored in the object even if you later cancel the drawing in the switchboard editor.

Feedbacks

Feedbacks or sensors are the eyes of the system. They notify which part of the track is occupied or what point on the track was passed. In general, they can cannot see who was responsible for it, but only that something happened. Also see Appendix B for more background information on feedbacks.

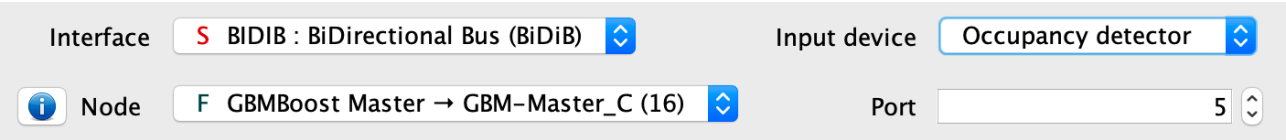


Feedbacks have an interface, an address or node, a type and depending on the type a length. It is recommended to specify the type especially when it is of type 'Occupancy', because it will affect the behavior in blocks.



For a feedback of type 'Occupancy' you can enter a length of the feedback. This length is used to calculate the positions of the feedbacks in the block and will be used to release previous blocks earlier. The feedback type 'Value (8-bit)' shows the actual value (range 0-255) that is only available on some specific input devices such as an OC32.

The 'Inverted' box allows to interpret the feedback the other way so that '0' is occupied and '1' is free. This is the default setting for a feedback of type 'Light barrier'. When a feedback is inverted you see the inverted value in the switchboard. In the feedback monitor (mentioned later in this manual) you see the original input from the interface on the border/edge of the element and the inverted value at the inside.



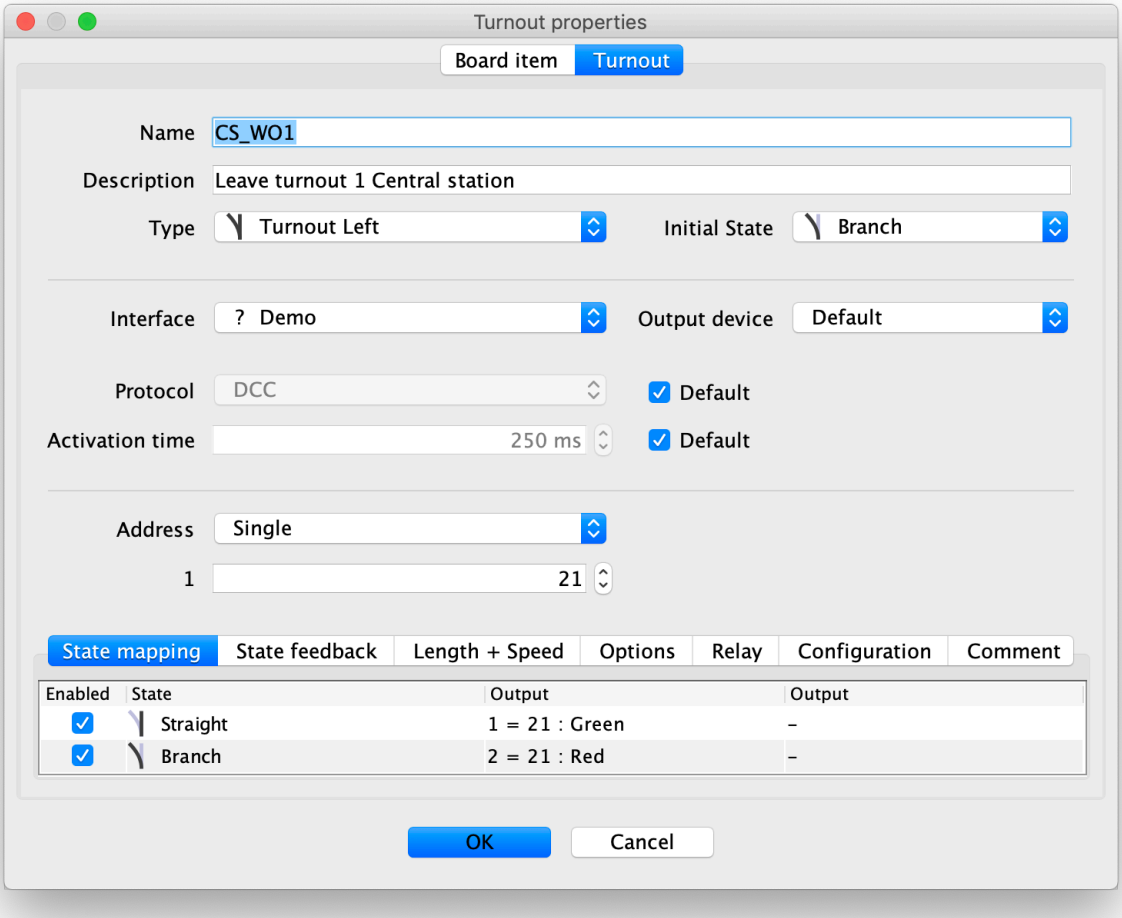
Depending on the interface, a feedback will have an address or in case of BiDiB a node, an input device and port number. The node indicates the module on the bus, the input device is the type of detector[36] and the port is the local port on this module starting at zero. The node can be selected from a list of available nodes. This list will be updated every time you connect to a BiDiB system. You can use the Identify-button to make a LED blink on the module when the system is online.

The 'Delay' of the feedback when switching them on or off is normally specified per interface, but you can override the values here for a specific feedback. When 'Default' is checked, you will see the default value in gray.

Tip: If the switchboard editor is zoomed in at least 200%, the feedback elements will show their name inside the red box. This is useful to check all your names in a big station on consistency.

[36] BiDiB support an 'Occupancy detector' (GBM), an 'Input' or a 'Position' for the FeedCar in OpenCar.

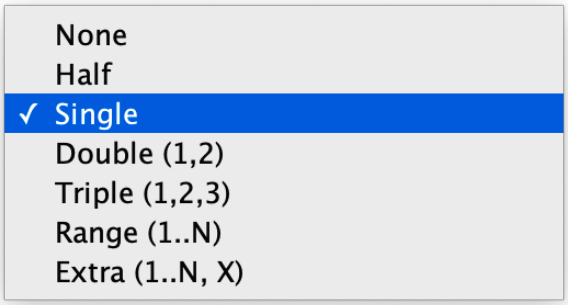
Accessories



An accessory is something with a state that switches something on the layout. In iTrain different types of accessories are available such as turnouts, signals, relays, lights, sounds, decouplers, crossings, turntables and transfer tables.

Depending on the interface, an accessory will have an address or in case of BiDiB it will have a node and port number or a node and an address.[37] If the accessory, for example a signal, does not really exist in your layout, but you want to add it to the switchboard (for example for hidden tracks) to show the internal state of the block, you do not assign it an address.

[37] You always have to select a 'Node' that sends a signal to the accessory, but depending on the output device you can select a DCC address or a local port on the node.



Depending on the accessory and the number of states, you might need one or more addresses. In the field 'Address' you can select how you want to specify these addresses:

- None - do not use an address.
- Half - use one output of an address.
- Single - use one address and both outputs.
- Double (1,2) - use two addresses. For the second you can specify if one or both outputs will be used.
- Triple (1,2,3) - use three addresses. For the third you can specify if one or both outputs will be used.
- Range (1..N) - use N consecutive addresses. You specify the first and the last address.
- Extra (1..N, X) - use N consecutive addresses plus an extra address. You specify the first and the last address of the range and additionally the extra address with one or both outputs.

In some cases a 'Connected to' box is added with option 'Both', 'Green' or 'Red', because the last address can be used half in case of an odd number of states.

For example, a three way turnout is one accessory with two addresses. The first address is for the right turnout and the second address for the left turnout. For 'Connected to' you select 'Both'.

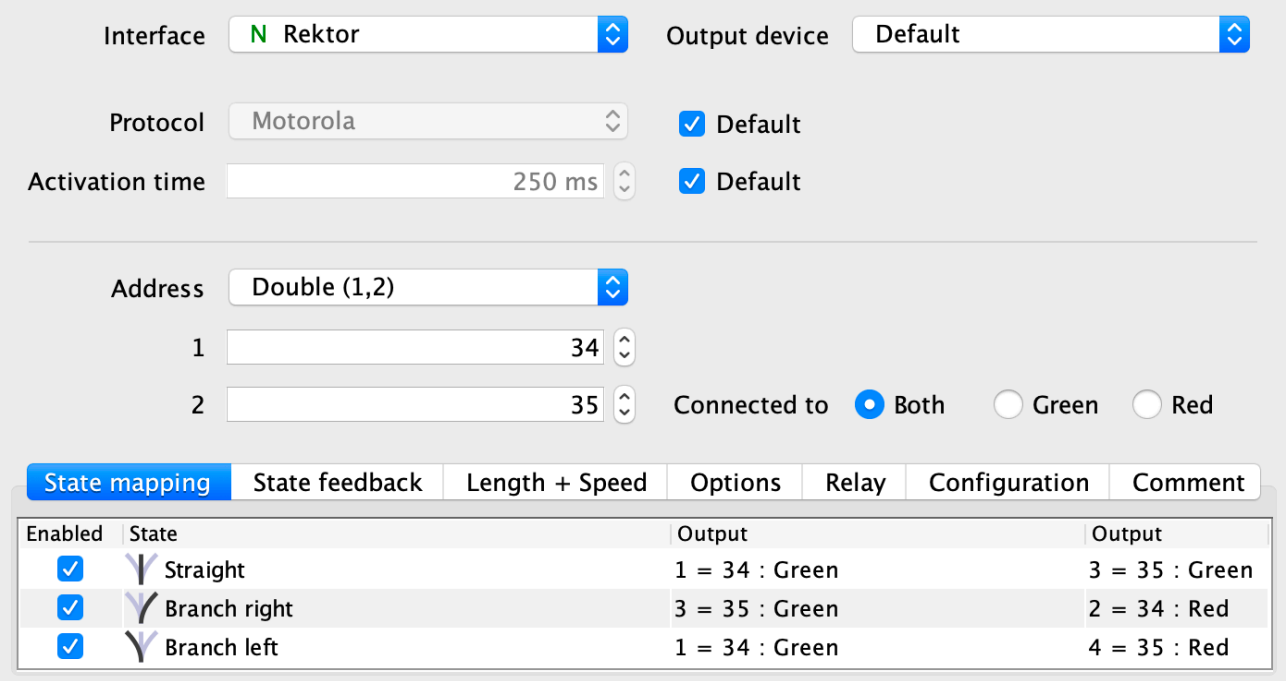
Note: In case the interface does not have an address, the field is still visible but it is called 'Pseudo address'. This pseudo address is only there to control this accessory manually from an address based remote control or a throttle on a virtual interface.[38]

The 'Activation time' box specifies the time between the activation of the accessory and the deactivation and must normally not be filled in. 'Default' must be selected to use the default of the interface as specified in the Interface editor. In some cases you might want to divert from the default settings and you can fill it in here.

State mapping

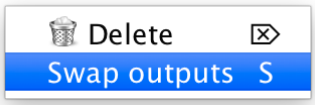
The state mapping tab allows you to exactly specify the outputs on the decoder that will be activated for every state. For every accessory and turnout they will have a good default value so that you don't have to change it in most cases, but you can if you want to. You can switch multiple outputs for every state change (for example, for a three way turnout you need two to set both sides). You can select from the addresses specified earlier. This way the mapping order is independent from the address and when you change the address the mapping will change accordingly.

[38] A virtual interface will be discussed in Appendix D in the section LocoNet and Selectrix.



The ‘Enabled’ column makes it possible to disable a state when it is no longer possible to select that state due to a mechanical or electrical defect (for example the coil doesn’t work or a wire is broken) or when the state is not supported by your accessory. Just keep the outputs settings so that you can continue when the defect has been fixed. Disabling a state will have effect when changing the state in the switchboard manually, but also when using train routes. Train routes will not consider this state an option anymore and will choose another option if available (in case of alternative blocks).

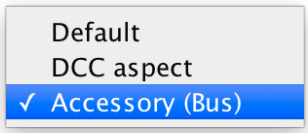
In most cases you will use the ‘State mapping’ to change the outputs of a turnout that was connected the other way around. You can select the two states for which the outputs need to be swapped, and press ‘S’ or use the popup menu. In case you only have two states, you don’t have to select them first.

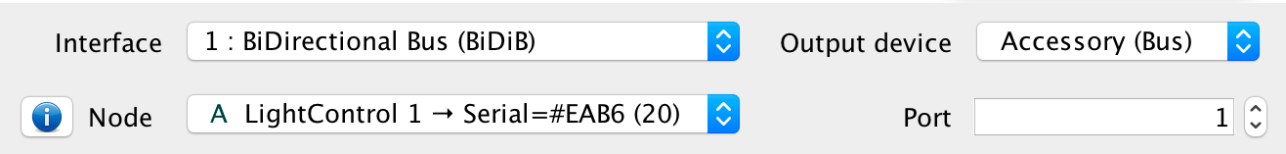


Output Device

Depending on the interface, users will see an extra ‘Output Device’ box to select how the accessory will be controlled. The choice ‘Default’ uses a normal accessory address with two outputs, but some interfaces also support ‘DCC Aspects’ so that a single address can have 32 different states so that one address will normally be enough. This is also called ‘Extended Accessory’ in the DCC specifications. You will need a command station and decoders that understand extended accessories to be able to use ‘DCC aspects’.

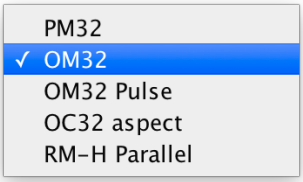
BiDiB users can control accessories via DCC (both default and via aspects), but they can also control accessories that are directly connected to BiDiB via the bus, and this option is called ‘Accessory (Bus)’.



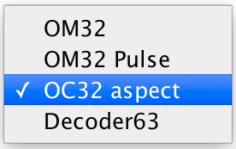


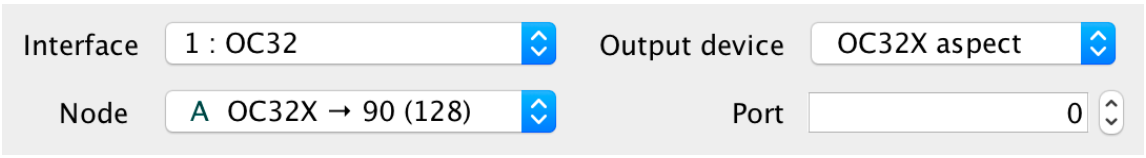
In case of the option ‘Accessory (Bus)’ you cannot enter an address, but you have to select a node. The node indicates the module on the bus and the port is the local port on this module starting at zero. The node can be selected from a list of available nodes. This list will be updated every time you connect to a BiDiB system. You can use the Identify-button to make a LED blink on the module when the system is online.

Dinamo users will have specific options to select if the accessory is controlled by the ‘PM32’, by an ‘OM32’ command, an ‘OM32 Pulse’, an ‘OC32 aspect’ or in case of an old RM-H, by the parallel outputs.

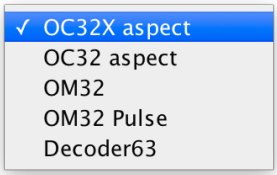


Direct OM32 or OC32 users will choose the “OM32” or “OC32” [39] as an interface and will see the same ‘Output Device’ box. Depending on the interface chosen, there are different options such as ‘OM32’, ‘OM32 Pulse’, ‘OC32 aspect’ and ‘Decoder63’[40]. When selecting ‘OM32’, ‘OM32 Pulse’, only one address box will appear that refers to the first output used by the accessory. This address starts with value one for the first output, because zero means no address used.





In case of the aspects of the OC32 there are two options. The classic ‘OC32 aspect’ with one or more addresses starting at one for older firmware, and the new ‘OC32X aspect’[41] that does not use addresses, but works by selecting a node and the local port or pin number starting at zero[42].



So when an accessory is connected to an OC32 you can select to use either the OM32 commands generated by iTrain (easy, but less flexible) or the preprogrammed ‘Aspects’ in the OC32 that have been created with the `OC32config` tool. In case the accessory has been connected to an OM32, always select ‘OM32’ or ‘OM32 Pulse’.

Tip: The ‘OM32 Pulse’ option generates only short pulses on the output like an ordinary accessory decoder or a PM32. You can use left over outputs on the OM32/OC32 for some accessories that need this, instead of buying a separate decoder.

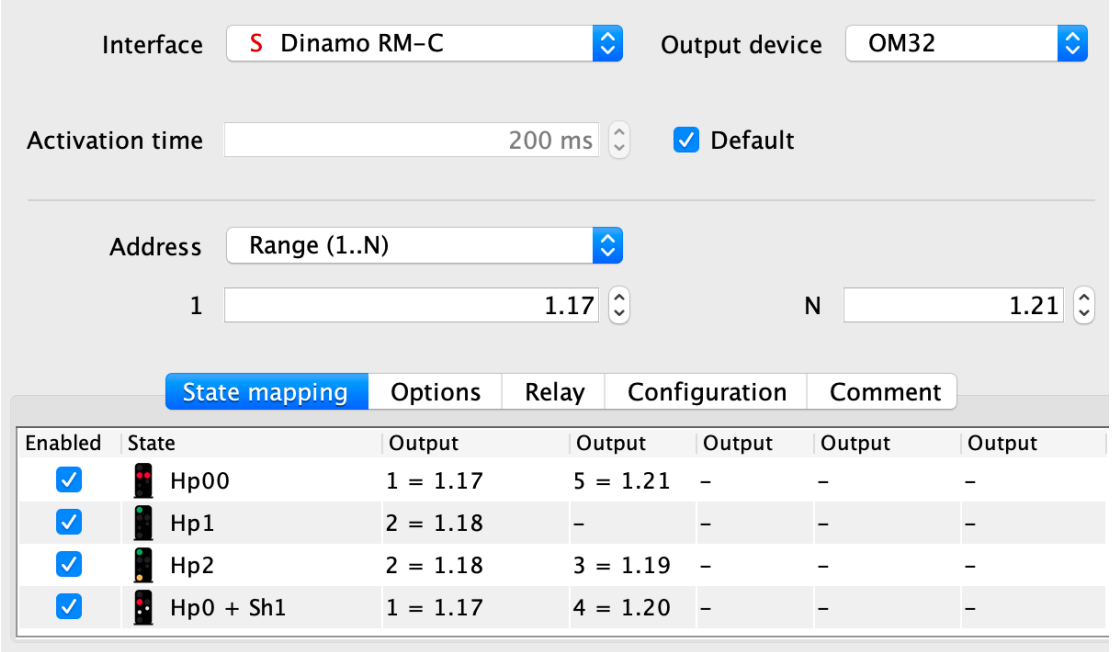
---

[39] You should only choose OC32 if you use the USB-RS485 converter and all devices on the RS485 bus are OC32 devices.

[40] A special turnout decoder for 63 turnouts that can be controlled by 8 outputs of the OM32 or OC32. The address ranges will be 1.1 - 1.63, 2.1 - 2.63, 3.1 - 3.63 and 4.1 - 4.63 for the first OM32/OC32, so one address is skipped every 64 positions (*.64) to better align them as can be seen in the Keyboard.

[41] OC32X aspect only works with OC32 firmware version 3.0 and higher.

[42] The numbering is now exactly as on the OC32 print, and also virtual ports/pins with values 0-127 are allowed.

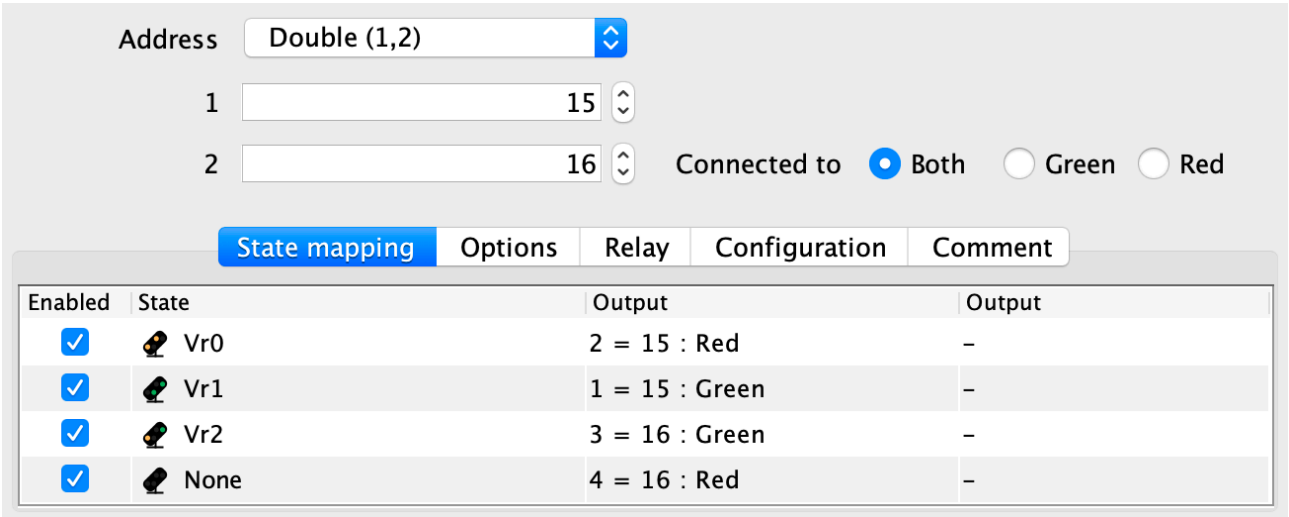


The number of outputs used in the OM32 depends on the type of accessory. For signals it is determined by the outputs used in the 'State mapping' table. In the example 5 outputs are used for the LEDs in the signal. Red left = 1, Green = 2, Yellow = 3, White = 4, Red right = 5. For turnouts, relays, lights, sounds and decouplers only one output is used that is switched like a standard relay (on or off).

Warning signals

A warning signal shows the state of the following home signal. It is only placed earlier on the track. In case the warning signal has the same address as the home signal, you should not define a signal object for the warning signal, but just use the home signal object. This can be achieved by selecting the home signal from the drop-down box on the 'Board item' tab. You can also do it in the switchboard, by dragging the home signal element and while holding the 'Control' key (or Alt) drop it on the warning signal element. Internally, both elements on the switchboard will now have one underlying signal object.

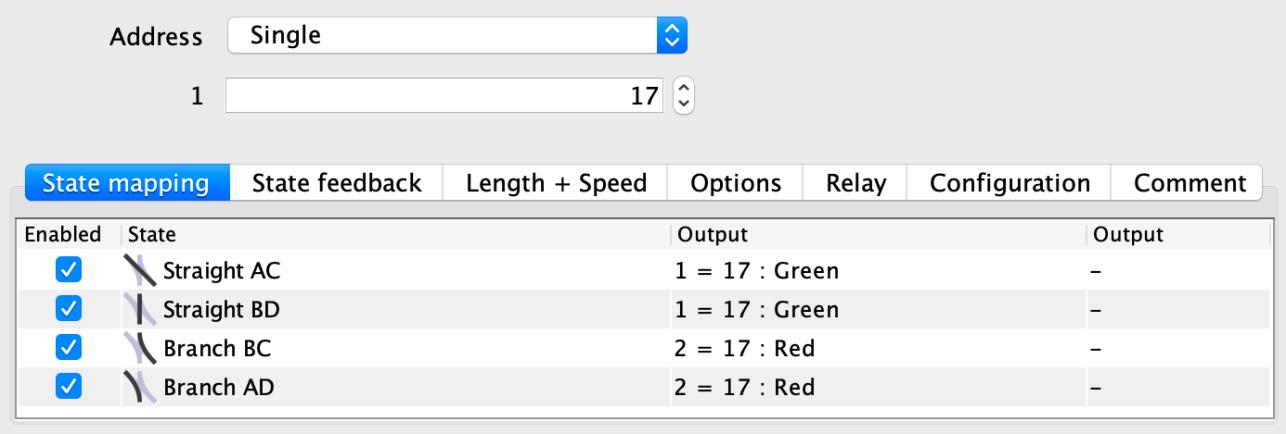
If the warning signal has its own address, it should have its own warning signal object. In the block definitions later on, you can attach this kind of warning signal to a block to automatically switch it to the correct state.



Combined home and warning signal elements represent two signal objects in one visual element. Depending on where you click, you will have access to the home or the warning signal. They should be treated as if you added two signal objects. In many signaling systems it is common that when the state of the home signal is stop (red), the warning signal will not show its state. This effect can be achieved for a warning signal with its own address by enabling the 'None' state on the 'State mapping'. If the 'None' state is available, then it will be used when the main signal is stop (red).

Cross turnouts

The cross is a special object that normally has no address (leave it empty or zero), but it is considered a turnout, because it can have two states on the screen (but only one on the layout). iTrain needs to know which direction is supposed to be used so that it can set the signals accordingly.

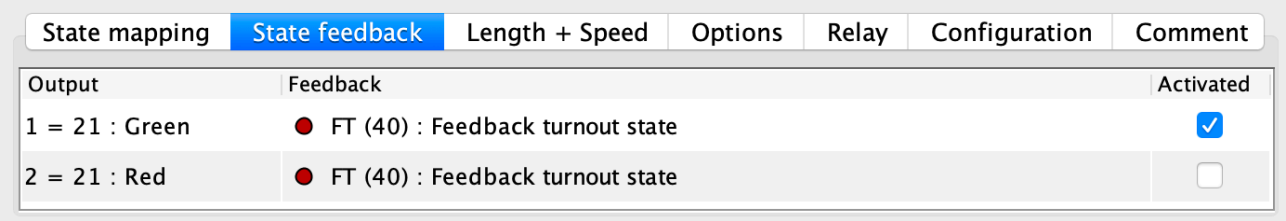


The normal cross turnout with two states cannot be selected anymore in this version and it is recommended using the double slip with four states that can have two addresses, even if you have a basic cross turnout with one address on the layout. In that case select 'Single' and fill in one address. The state mappings will be adapted automatically so that both 'Straight' states map to the same output and both branch states map to the same output (see picture). Now you can select four directions in the screen, but in the layout it uses two states. The advantage is again that the signals can be set correctly depending on the direction that is selected in the screen.

The sides of a cross are called A, B, C, D (clockwise starting in the upper left corner) so the possible directions are 'Straight AC', 'Straight BD', 'Branch BC' and 'Branch AD'. In edit mode a dot is drawn close to the A-side on the switchboard to identify the rotation.

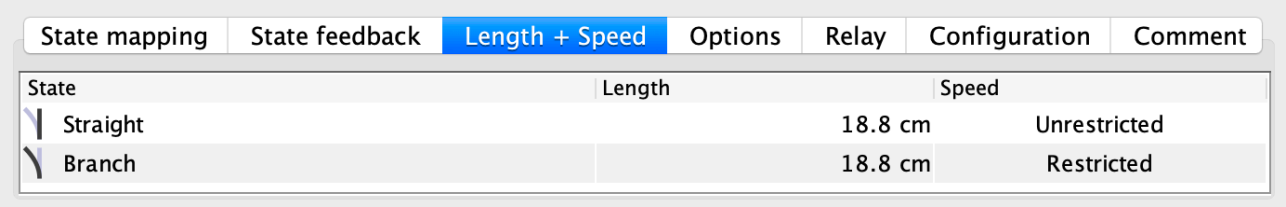
Turnout State feedback

By default a turnout is switched, and after the activation time it is considered to be in the preferred state. If you want to guarantee that the turnout has been switched correctly, you can connect feedbacks to the turnout. To be able to support, for example, feedback for a three way turnout, the feedbacks are not attached to the states itself, but to the outputs. This way it is possible to detect which turnout has changed. So there is a relation between 'State mapping' and 'State feedback', to be able to find back the turnout states.



The ‘Activated’ flag in the third column determines if the feedback will be activated or deactivated when the turnout reaches its end state. Normally every output would have its own feedback. It is however possible to use one feedback for a turnout and attach it to both outputs, but one time when its activated and one time when its deactivated.

Turnout Length + Speed

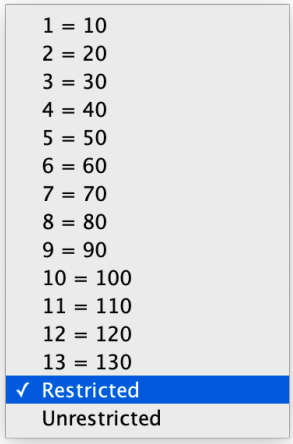


Turnouts have a length, but this length is dependent on the state of the turnout. In the tab ‘Length + Speed’ you can specify the length for every state.

In case tracks between two turnouts do not belong to a block, you have to assign the length of these tracks to one of the turnouts or split the length up and assign half of the length to one turnout and the other half to the other turnout. It is important that when a route over the turnouts is calculated, the total length will be correct.

NB: It is always recommended to fill in the lengths of the turnouts as they are used for many purposes, among them the release of turnouts and blocks.

The last column, ‘Speed’, affects the signal state before the turnout.[43] The value ‘Unrestricted’ means you can pass the turnout without any speed restriction (for example Hp1). The value ‘Restricted’ means there is a speed restriction and the signal will show a restricted state (for example Hp2).

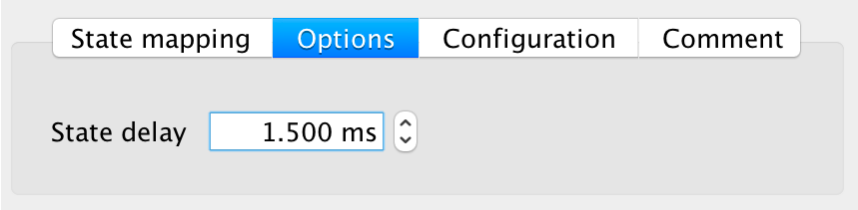


In addition, there are restricted states with a specific number (1-13) that relate to specific speeds that are allowed on this turnout. This specific number will also be used to select the correct restricted speed of a signal, in case multiple restricted states are available, or to show the correct digit(s) on the display of a signal.

Options

The tab ‘Options’ contains some other settings for an accessory.

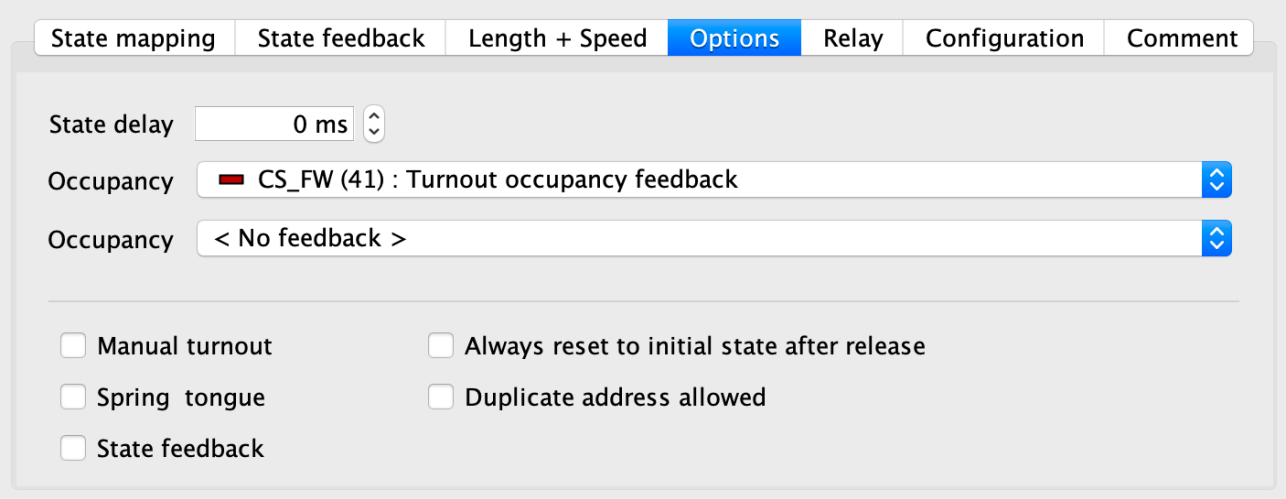
[43] This column replaces the ‘Speed type’ in previous versions of iTrain.



The option ‘State delay’ specifies the extra time that should elapse before an accessory is considered to have changed state after the command has been given. So it is added to the activation time. If there is no activation time (for example with an OM32/OC32), it is the time it takes to change the state of the accessory.

Tip: Servos normally take more time to change their state than the actual command to the interface. To prevent trains driving over turnouts that are not completely set, you can specify the ‘State delay’.

Turnout options



The tab ‘Options’ contains a few additional fields for turnouts. The ‘Occupancy’ feedback notifies if part of the train is still on turnout. In case this feedback is ‘On’, the turnouts will not be released. One feedback can be assigned to multiple turnouts so that when the turnouts are interconnected you don’t need a feedback for every turnout.

The first option, ‘Manual turnout’, indicates that the turnout will have a physical fixed state that can only be changed manually outside of the program. An orange cross is displayed next to the turnout to indicate that clicking on the symbol does not change the turnout.

A manual turnout may normally only be used in the displayed state. If the option ‘Spring tongue’ is used in addition then it is also allowed to use the turnout from the blunt side to the sharp side (in the image from left to right) with the other state(s), because it will spring back after the train has passed the turnout.[44]

Note: A turnout without an address normally indicates a manual turnout, so please select the right options here.

Tip: To synchronize the physical state with the state in iTrain, you can hold the Alt-key and click on the turnout. This changes the initial state of the turnout.

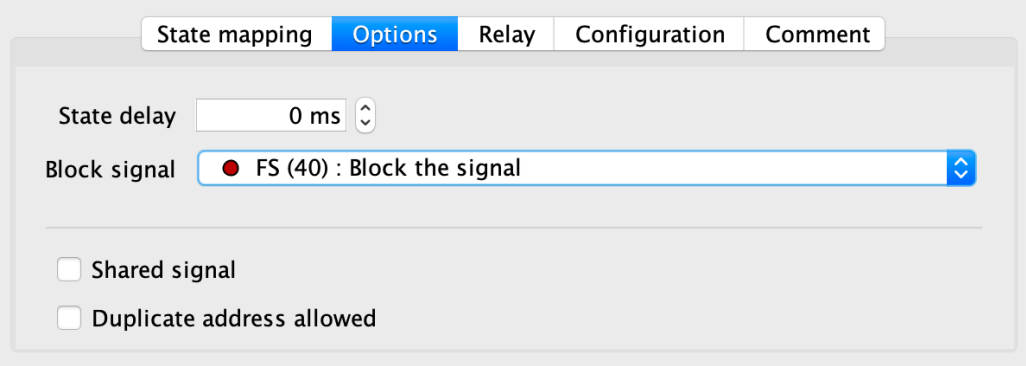
[44] All Märklin turnouts have a spring tongue.

The option, ‘State feedback’, indicates that the turnout has direct state feedback and no external feedbacks specified in the tab ‘State feedback’. It will not directly show its new state, but it will wait for a message from the interface.

The option, ‘Always reset to initial state after release’, will put the turnout back to its initial state after releasing its reservation. When this option is checked, it will overrule the option in the general settings. If it is unchecked, the turnout will follow the general setting.

The option ‘Duplicate address allowed’ is used to indicate that the address of the turnout may be used by another turnout as well and will give no error in the diagnosis tool.

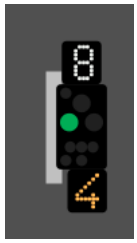
Signal options



There are additional options for signals. You can specify a feedback to hold the signal at Stop. Holding a signal means it will change to red as soon as possible and then it cannot be changed until the feedback causing the hold has been released.

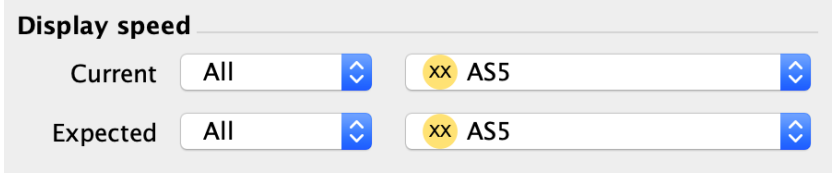
The first option, ‘Shared signal’, indicates that a signal is shared by more than one block and will give no error in the diagnosis tool. This is normally used to allow sharing of one main signal in special cases.

The second option, ‘Duplicate address allowed’, is to indicate that the address of the signal may be used by another signal as well and will give no error in the diagnosis tool.



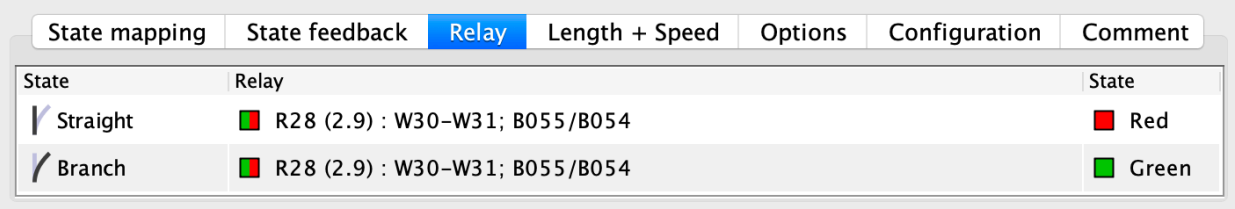
In case the signal type allows one or more displays to indicate the allowed speed, an extra section ‘Display speed’ appears at the bottom.

You can specify the values that are visible on the signal displays in the switchboard in case only a fixed value per display is supported by the signal itself. If multiple values per display are supported, you have to select the value ‘All’ and choose an additional object ‘Aspect’ that controls the display of the signal.



In case the signal has only one display, showing the current or expected speed dependent on the state of the signal, you have to select the same object 'Aspect' twice.

Turnout Relay



For turnouts there is an extra tab 'Relay' to couple relays with a turnout. For every turnout state you can specify the state of a relay. You can use this in the following cases:

- Polarization - Some track parts in the turnout need to receive + or – depending on the state. A relay will feed the track part with either one of them.
- Dinamo - Turnouts get their track power via neighboring block cards. In some cases this can be from multiple block cards via one or more relays depending on the state of the turnout.

Note: A relay of type `on/off` *specified for one turnout state will always be switched off in the other case. So you can use two different relays of type* `on/off` *with a turnout with only two states. Relays of type* `A/B` *only change state as specified, so normally only one relay will be used per turnout.*

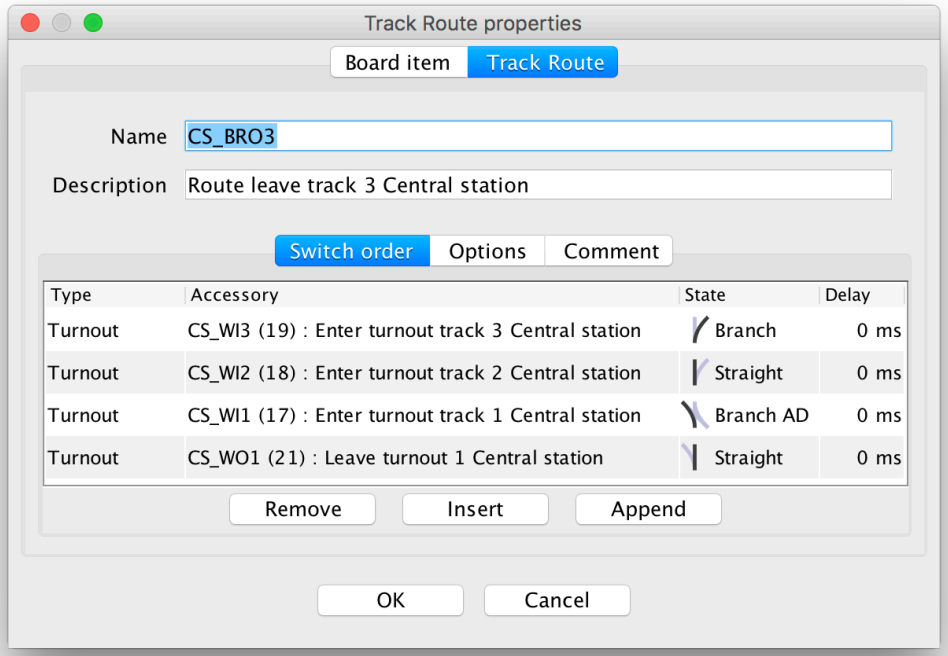
Tip: In case you really need more than one relay per state, you can double click on the column header 'Relay' in the table to get one, two or three rows per state.

Signal Relay

For signals there is also an extra tab 'Relay' to couple relays with a signal. For every signal state you can specify the state of a relay just as with the turnouts.

Track routes

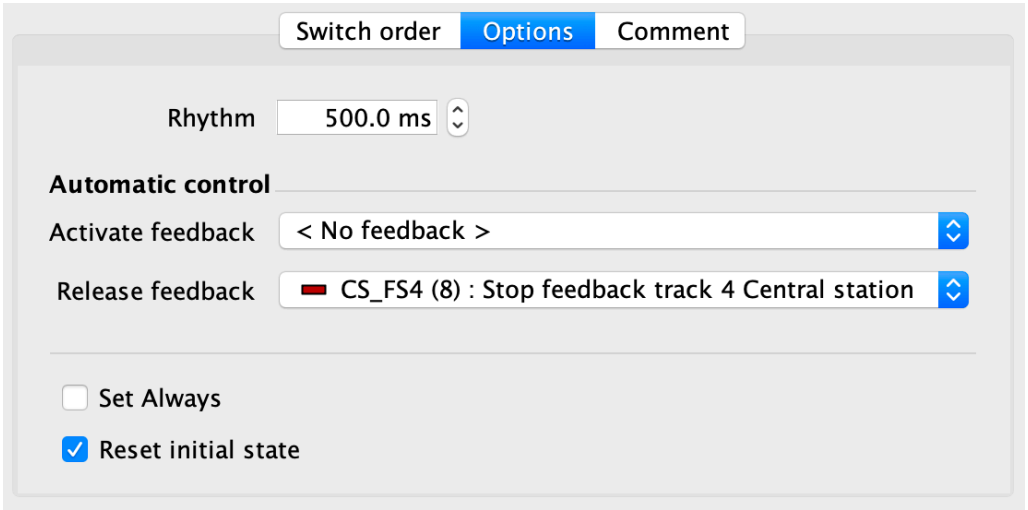
Track routes behave like the Memory module. You can set a list of turnouts or signals in order to a specific state and they will be set and reserved as long as the track route is active. That means that they cannot be set manually anymore by clicking or by other track routes, and they are not used by train routes.



On the ‘Switch order’ tab you can create the list of accessory states. The three buttons allow you to remove a row, insert a row at the selected position or append a row at the end. When filling in a row, first select the ‘Type’ of accessory (turnout, signal, relay or crossing) in the first column, then select an accessory in the second column from the drop down box and finally set the desired state in the third column.

Tip: First assign all your accessories and feedbacks before creating your track routes.

The last column ‘Delay’ allows you to set an individual delay per accessory before they will be switched. This makes it possible to make a timed sequence in which, for example, light can be switched on and later on the same sequence be switched off again. The specified delay will always be used, even when the accessory is already in the right position. A value of zero means no specific delay.



On the ‘Options’ tab, some extra settings on a track route are available.

The ‘Rhythm’ box specifies the time between switching the individual accessories, but only between the ones that are actually switched. By setting it to zero all accessories will be set

as quickly as possible, but in many cases it is useful to keep some space so that you are sure all turnouts are set before you set the signal. If an individual 'Delay' greater than zero has been specified per accessory, this delay will be used instead of the 'Rhythm' value.

Track routes can also be activated or released by feedbacks (see 'Automatic control') that have been defined before. This way you can at least always release a track route when the train passes a specific sensor or feedback.

But more advanced examples are possible, because the feedbacks will react the same way as the Märklin[45] Memory (6043):

- The 'Activate feedback' will not activate the track route if the 'Release feedback' is still selected.
- When the 'Release feedback' is deselected (freed) and at the same time the 'Activate feedback' is still selected, it will activate the track route.

If the 'Set Always' box is selected, it is guaranteed that all the accessories will be switched by the interface, even when the program thinks they are already in the correct state. This costs more switching time with long lists, but prevents errors by manual changes outside the program.

To reset the accessories back to their initial state after they have been released, you can use the 'Reset initial state' box. It will only reset them to their initial state again if they are not reserved immediately by another track route, to prevent unnecessary switching.

Blocks

In reality a block system is used to prevent train collisions by assuring that only one train can be in a block at the same time. If a block is occupied, a train in a previous block must stop and wait until the block is free again before entering it. In iTrain this is not any different. Therefore you have to divide your layout into blocks taking into account that turnouts are not part of a block and are always a border of a block. Long tracks between turnouts can be split into two ore more blocks. Normally a block should be able to contain the longest train so this is the main point. Of course sometimes a block will be shorter, because it is between two turnouts, so you have no choice. If the space between two turnouts is very short, you could also consider this as no block and configure it as if the two turnouts were directly connected without a track in between.

Every block is connected to other blocks directly or indirectly via turnouts, so each block must know what its neighboring blocks are on both sides (one side may not be connected). In iTrain the sides of a block are called 'Previous' and 'Next' indicating the default direction of the block from 'Previous' to 'Next'. In case the block is used in two directions and there is no preferred direction, just pick one arbitrary direction. It is no problem to connect a 'Next' side of one block to the 'Next' side of another block if that would make sense for you.

To indicate a block in the switchboard you need to assign multiple track elements (with signals and feedbacks and maybe more) to a block. An element with the white rectangle (in edit mode with an arrow for the text direction or the block name) represents a block. It is used to define the block.

[45] Märklin is a trademark

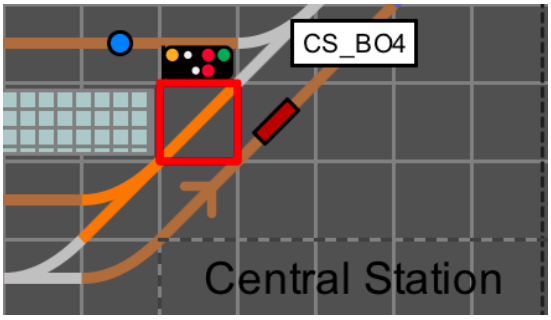
In edit mode you can assign other track elements to the block by first selecting the block element and pressing the ‘G’ key for grouping (see popup menu). Now all track elements that belong to that block are colored orange. You can now assign extra track elements to the block (not including turnouts) by selecting them and pressing ‘G’ again. You can also do it in one step by selecting all elements belonging to the block (including the block element) and pressing ‘G’. To remove already assigned track elements from a block, select them and use Shift + ‘G’. By selecting an empty element cell and pressing ‘G’, you turn off the highlighting of all elements belonging to the block.

Tip: The block names will be displayed in the block elements if the zoom level is 125% or more and you are in edit mode.

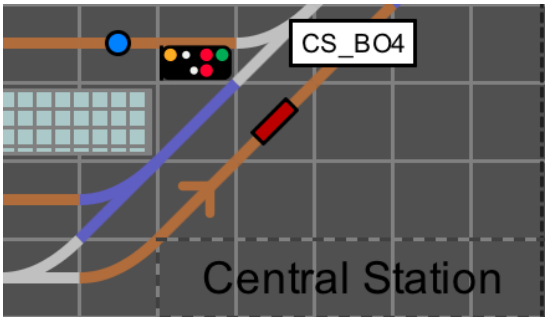
In switchboard edit mode all elements assigned to a block have a kind of brown color (color in between orange and black).

Turnout grouping

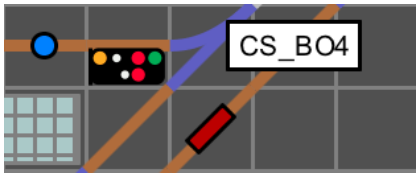
The tracks between turnouts that do not belong to a block can be assigned to a turnout by using the same ‘G’ key to group elements together. The advantage of grouping is that these track elements will get the same color as the turnout they belong to when controlling the layout. This makes it easier to see the exact path over the turnouts.



To group them, first select the turnout with ‘G’ and then the extra elements with the same ‘G’ key. In general it is better to add the track elements to the sharp side of the turnout, because the relation is then independent of the state of the turnout. If this is not possible you add them to another side of the turnout.



When no grouping is active on the turnout, it will show a lilac color to indicate it is directly attached to the turnout.

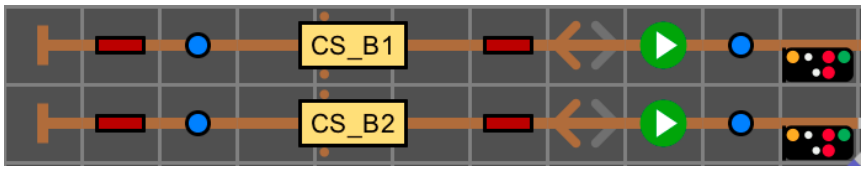


In case it is attached to a turnout, but only when a turnout has a specific state, it will have the same kind of brown color as an element attached to a block. You will only see this if you close the editor and open it again, because it will automatically assign the track element to the correct state of the turnout when you leave edit mode.

Note: It is only necessary to group one turnout with normal track elements that are not attached to a block. Never try to group two or more turnouts together as this is not allowed.

In a well defined switchboard in edit mode, all non-turnout elements will have a non-black color. Turnouts will be black or lilac when they have been attached to a track element.

Arrows

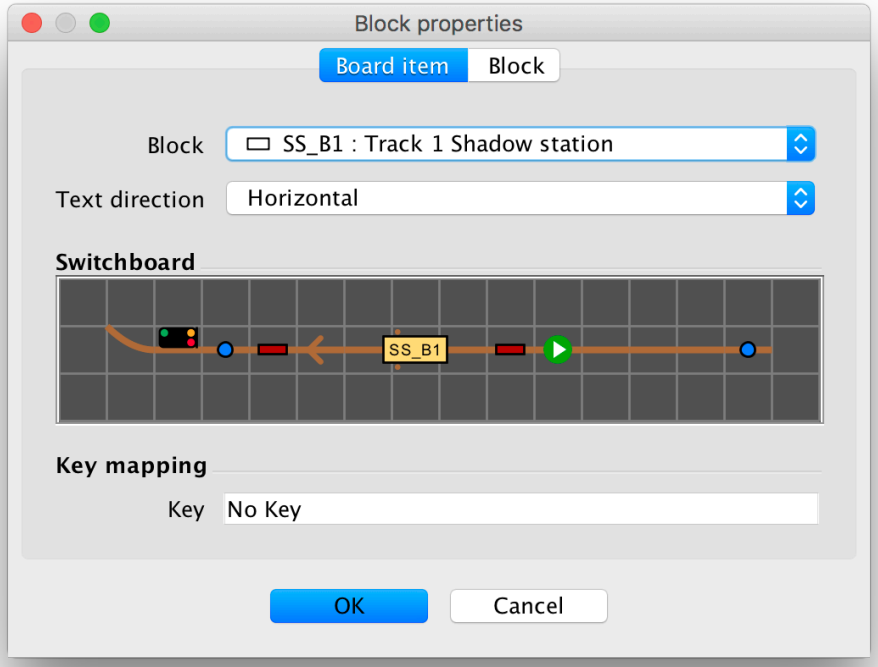


The arrow element shows the direction of the train in the block and is a recommended element in every block. In edit mode it will show the default direction of the block itself and you have to make sure that it points from the ‘Previous side’ to the ‘Next side’, not only to get good visual feedback about the direction of a train in the switchboard when controlling, but also to able to automatically fill in some of the properties of the block. In case both directions are allowed, the darkest of the two arrows should point in the default direction from the ‘Previous side’ to the ‘Next side’.

Note: In edit mode the block element also shows two dots on the ‘Next side’ . So it is always clear what the direction of the block is even if no arrow has been added.

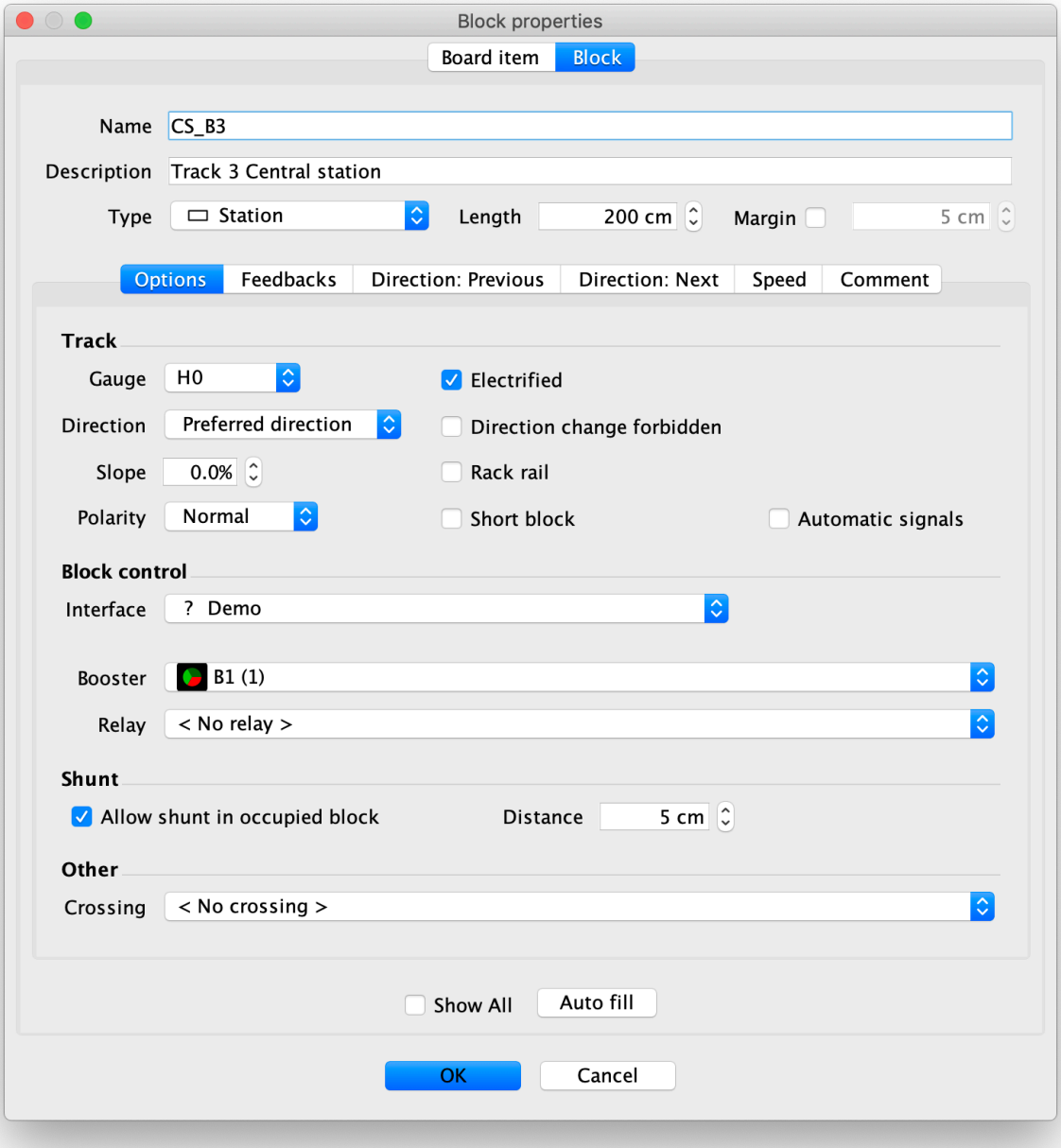
Edit

To edit the block properties, you double click on the block element or you select it and press the ‘Enter’ key. A dialog with two main tabs will appear. In the first tab you see a simplified version of the switchboard with only the elements assigned to this block. This can be used as a check to see if you did not forget to assign an element to the block. It is this definition of the block that is used later on to automatically fill in some of the properties of the block, such as the feedbacks, signals and block connections.



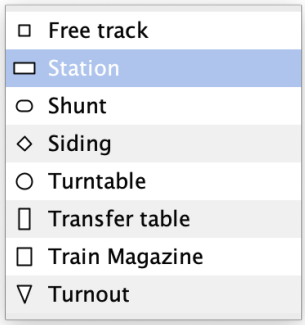
The text in a block element will by default be horizontal so from left to right, but when the block in the switchboard is drawn vertically you might want the text to be vertical as well. You can change the 'Text direction' to be 'Horizontal', 'Up' or 'Down'.

You can now select an already defined block definition from the 'Block' input or select '`<No block>`' and define a new one on the second tab.



Type

The type of the block affects how signals are treated and what the maximum speeds are. Justifications per individual block are possible via the 'Speed' tab, discussed later.



The type 'Shunt' is either meant for blocks between the entry signal and the station to make it possible to shunt, or for short blocks without home signals between turnouts. The type 'Siding' is for tracks with a dead end or between turnouts to park trains. The two types 'Turntable' and 'Transfer table' are special types that are automatically assigned and they should not be used outside of their context. The type 'Train Magazine' can be used to indicate that a block is part of a train magazine. The type 'Turnout' is a special type for block controlled systems only, like Dinamo, in which a block output is used to feed a turnout.[46]

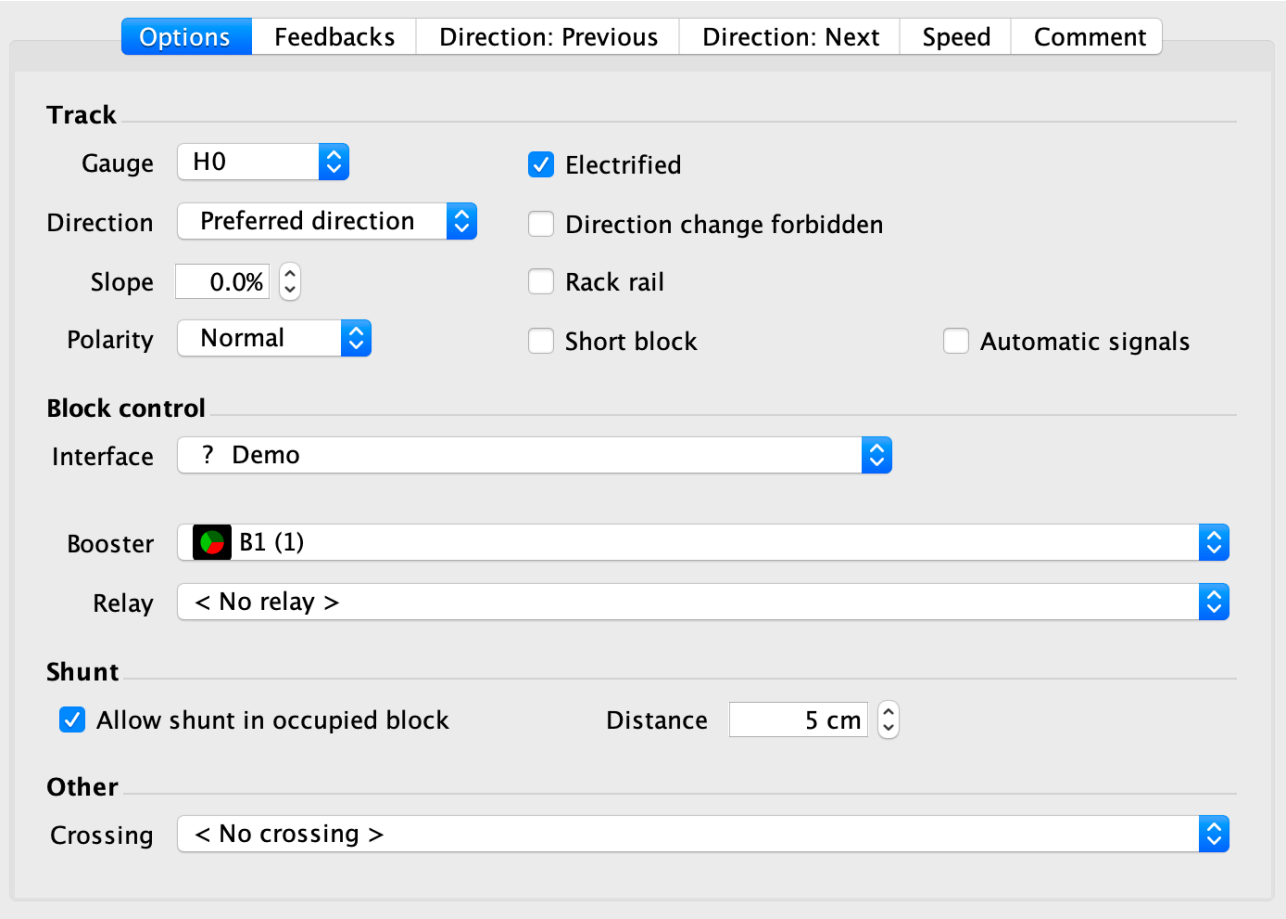
[46] A background block of the type 'Turnout' and the same name as the turnout can be used as an alternative to using relays to feed turnouts.

Length

The length specifies the total length of the block. The margin is the extra space necessary to compensate for differences in stop positions for different trains. There is a default value for the margin, based on the block type, but you can override it. The length minus the margin defines the usable length in a block. Trains that are longer are considered not to fit in the block and will not release previous turnouts and/or blocks when stopping in the block.

An empty length or length zero means unknown. In that case the program will assume that all trains will fit in the block. So especially for short blocks it is very important to measure the length. For very long blocks in which any train will fit, it is less important for safe train control, but it will be useful for better throughput in case you also specify the feedback positions.

Options

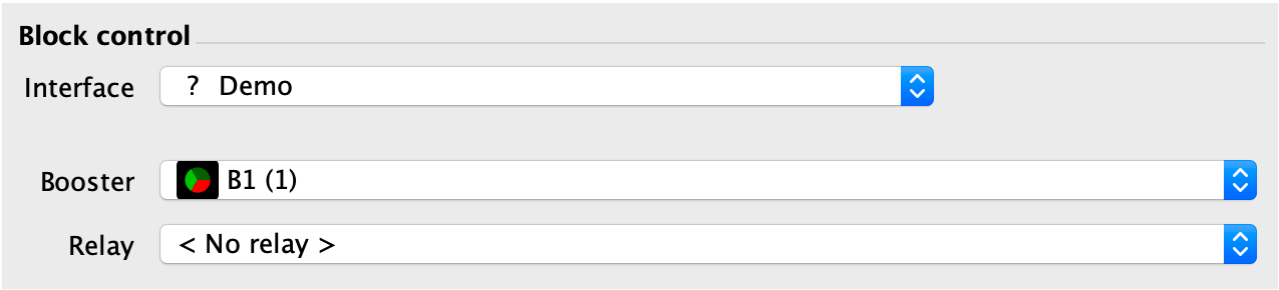


In the ‘Options’ tab you can specify some other settings of the block. It has been divided into three sections. In the section ‘Track’ some properties related to the tracks can be specified:

- The ‘Gauge’ or track width of the track specifies the allowed gauges. Only trains that meet this gauge are allowed in this block.
- The property ‘Direction’ of the block: A block can be allowed to be driven in both directions or in just one direction. Even if it is allowed to travel in both directions in some cases, in most cases it should be used in one direction. This is called ‘Preferred

direction'. The single or preferred direction is always from 'Previous side' to 'Next side'. This 'Direction' setting directly effects the way the arrow element in the switchboard is drawn and it is also used by routes (described later) to determine what the preferred direction is.

- The property 'Slope' specifies the incline of the block in percentage from the 'Previous' side to the 'Next' side and uses negative values for a decline. It is meant for speed corrections for locs with no cruise control in future versions, but it is currently not used.
- The property 'Polarity' is only for 2-rail systems and specifies how the two rails are wired. This is only used in case you have a RailCom detector in the block that uses the polarity to notify the direction of the detected loc, or when using a block controlled system like Dinamo.
- A block should be marked 'Electrified' if it has catenary, so that it can be used by trains with an electric main loc. If this is not marked, trains with an electric main loc will not be able to enter the block or start driving automatically from this block.
- The option 'Direction change forbidden' indicates that it is not allowed to change direction in this block when driving automatically. This prevents unwanted direction changes in this block in routing.
- If 'Rack rail' is checked, only trains with a main loc with this same option are allowed in this block.
- The option 'Short block' should only be checked if the block should be considered short for the signaling systems.[47]
- The option 'Automatic signals' specifies how signals attached to the block will react on neighboring blocks. If the box is unchecked or a turnout follows, the signal is always red, unless the following block has been reserved by the same train. If the box is selected and no turnout follows, the signal will be set to green automatically when the next block is free. By default this option is only selected for blocks of type 'Free'.

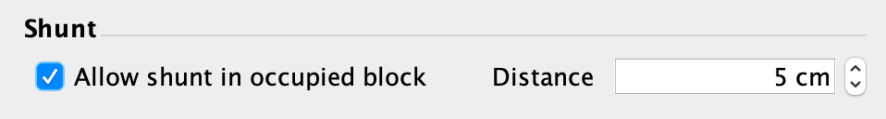


The section 'Block control' serves to indicate how the track section of the block is connected to the system.

- The property 'Interface' specifies the interface that is generating the rail signal for this block. You can leave it empty in case all blocks are controlled by the same interface and your system is not Dinamo. In this case the interface is determined by the interface of the vehicle in the block. If you specify an interface for the block, then this interface will be used by the vehicle when it comes in this block.

[47] Short block is currently only used for Dutch (NS) and Belgian (NMBS) signals.

- The property ‘Booster’ specifies the booster that is responsible for the track power in this block. In case you have defined so called ‘intelligent’ boosters, you have to select the booster here. If you have not defined any booster, just leave it to ‘`<No booster>`’.
- The property ‘Relay’ specifies a relay that can either switch off the power (`on/off` relay) or change the track signal of the blocks (`A/B` relay).

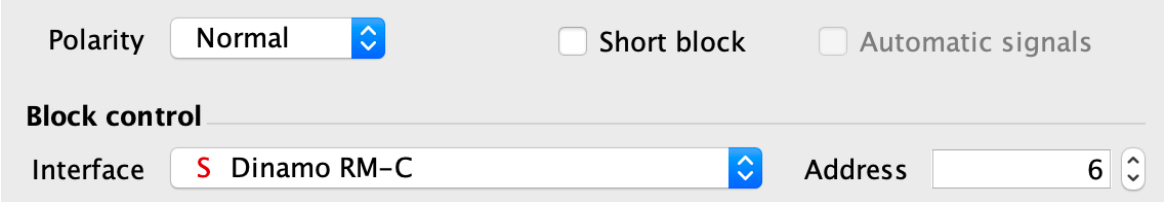


The section ‘Shunt’ indicates if it is allowed to shunt into an occupied block. If this is allowed, you have to enter the distance or space that should be kept between the trains. Some space between the trains should be left, so you can park multiple trains in a block, without accidents.

The crossing in section ‘Other’ attaches a crossing to a block. It will be discussed in the section about crossings.

Some of the properties are discussed here in more detail.

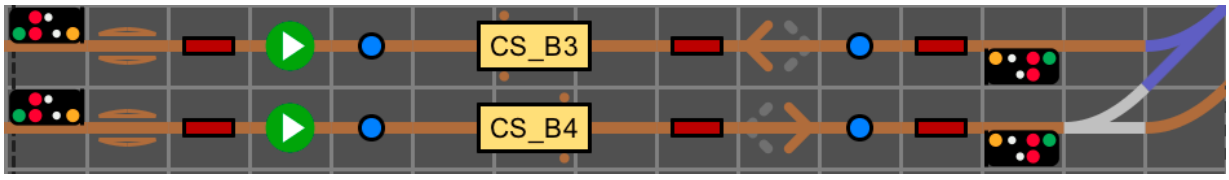
Dinamo



For block controlled systems like ‘Dinamo’, you have to specify the polarity and the address of the block. This is the address on the TM-44 or TM-H (starting at 1, because 0 means no address).

Polarity

For all 2-rail systems you have to decide which rail is the ‘+’ and which one the ‘-’. We call this the polarity. The polarity is a relative matter so we first have to define the reference. Let us assume that every track has a ‘red’ wire called ‘+’ and a ‘black’ wire called ‘-’. Now we look at the blocks in the direction from ‘Previous’ to ‘Next’. If the ‘+’ is on the right rail and the ‘-’ on the left rail we call that ‘Normal’ polarity. If in a block has the ‘+’ on the left rail we call that ‘Inverted’ polarity. You can also choose the ‘+’ and ‘-’ the other way around if that makes more sense for you, but stick to one definition.



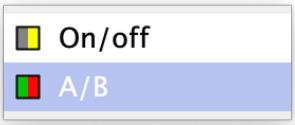
In case two neighboring blocks are wired the same (+ and -), but in the program they are defined in an opposite direction (see example above), then they have a different polarity in iTrain and one should be assigned the 'Inverted' polarity.

Tip: Try to use 'Normal' polarity for most blocks. When using 'Dinamo' choose the option 'Invert total polarity' on the 'Specific' tab of the 'Interface' editor (menu 'Edit' -> 'Interface') in case analog locs are going the wrong direction.

For normal digital 2-rail systems it is optional to fill in the polarity. It will probably be used in the future if relays that switch the polarity are coupled with blocks to automatically solve polarity conflicts, such as with turntables.

Relay

There are two types of relays that can be coupled with a block. They are both used to change the available track signal in the block.



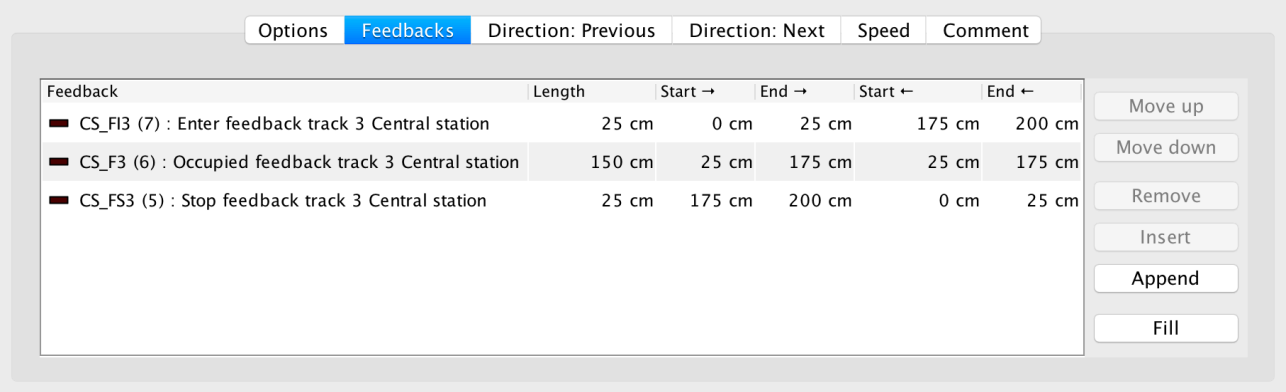
The type '`A/B`' is intended to change the track signal when the train is passing the block from one side to the other. One purpose is to change the polarity of the block (for reverse loops) and another to change the command station that is feeding the block. The relay will be put into the 'Block connections' with a different state on both sides. It is necessary that the train fits inside the block to be able to pass it. The train might slow down and even stop in the block until the relay with the previous state has been released.

The type '`on/off`' is intended to physically disconnect a block and its feedbacks from the system to park trains. The relay will be put into the 'Block connections' so that it will be reserved and switched '`On`' when creating a path to the block. The relay will not be released until the train has really stopped in or left the block, so that it will not stop abruptly because the power has been switched off too early.

In Dinamo systems relays attached to blocks can be used to reduce the number of block outputs (TM-H/TM-44) in shadow stations or sidings. This will only work properly if of all the relays attached to blocks with the same address, at most one will be active at a time. Internally iTrain will create relay groups to guarantee this. Attaching a relay to a block and using the button 'Auto fill' should be enough to make this work. The only requirement is that the relays are controlled by the Dinamo interface (possibly via an OM32/OC32) and not via a second interface.

Feedbacks

It is important to know where a train is to be able to guarantee that it will not enter an occupied block. Therefore, every block contains feedbacks or sensors to inform the block what is happening in the block. The tab 'Feedbacks' shows all the feedbacks in the block, together with the position of the feedback in the block. To fill the list of feedbacks, you will normally first use the button 'Auto fill' to get the list of feedbacks.



Per feedback, some extra columns are shown. At first the length of the feedback in case of an occupancy feedback. This is actually a property of the feedback, but it can be edited here as well for convenience. The next four columns show Start and End position of the feedback in the block in both directions. The first two in 'Direction: Next' and the last two in 'Direction: Previous' (see next section). Only the value in the first column 'Start' is the actual position and will be stored. The other values like the 'End' position and the 'Start' and 'End' position in the other direction are only shown here for convenience. You can enter a value for both 'Start' columns depending from what side of the block you are measuring, and the other values will be calculated.

Note: To fill in the position of the feedbacks in the block it is necessary that the length of the block has been specified first.

Entering the positions can be tedious. If the lengths of the feedbacks are filled in and you either have only one or two feedbacks, or the length of all the feedbacks together is equal to the block length, you can use the 'Fill' button to fill them in automatically. Before you do that it is important that the order of feedbacks is correct (from 'Previous' to 'Next'), and if necessary, you can adjust the order by using the buttons 'Move up' and 'Move down'.

Note: The feedback positions have become more and more important in iTrain over the years. The general idea is that the more information you give iTrain about the layout, the better it can make decisions about where the train is in the block and when to release a block. So please fill in as many lengths and positions as you know or still can measure. This will pay back in the future.

Direction tabs

A block has two sides so in theory it can be driven in two directions. To distinguish these two directions both driving directions need to have a name. They are called 'Direction: Previous' and 'Direction: Next'.

- Direction: Previous - is the direction from side 'Next' to side 'Previous' and in case the block has a preferred direction this is the non-preferred direction.
- Direction: Next - is the direction from side 'Previous' to side 'Next' and the preferred direction in case of a preferred direction and the only allowed direction in case of a single direction block.

Some important properties are specified per driving direction so there are two tabs with the same properties. It is necessary to specify them for both directions to correctly follow the train even when the block is only used in one direction.

Tip: The 'Auto fill' button can fill in most of these properties automatically. It is however required that all the feedbacks, signals, turnouts and neighboring blocks that will be chosen already have a name for correct results. So it is recommended to first assign names to all these control objects and then use the 'Auto fill' button to attach them to the block.

The first choice you have to make is called 'Use positions'. This choice makes a distinction between the two ways of controlling the block:

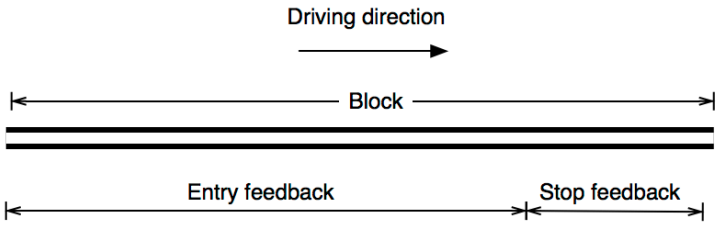
- 'Classic mode' - using entry, brake and stop feedbacks (optionally with shifts) to control where to stop. This requires at least two feedbacks per block to be able to stop. The stop position might differ between trains depending on its braking characteristics.
- 'Use positions' - using exact brake and stop positions that are the same for every train (optionally with a platform definition to stop a train centered around the middle of the platform). Works well with one feedback and even better with multiple feedbacks as long as the feedback positions have been entered correctly. It requires that speed measurements have been done for all active locomotives and all train lengths are known, because iTrain will make time/distance calculations to know where the train is in the block based on the speed measurements.

Depending on this choice the tab will show different fields. We first describe how it works in 'Classic mode' and later we describe what changes when using positions.

Tip: Even when you want to use positions exclusively, we recommend reading both parts. For some shorter blocks in which you don't want to stop you have to use the classic layout to indicate that.

Feedback

To detect that a train is in a block, at least one feedback is necessary. One feedback allows to follow a train so that the system knows in which block a train is. However with only one feedback it is not possible to accurately and nicely stop the train in front of a signal easily[48], because every loc has different characteristics. A stop section before a signal would solve this. If a train enters the block and needs to stop in this block, it can slow down in the main part of the block, but it will only really stop in the stop section at the end.



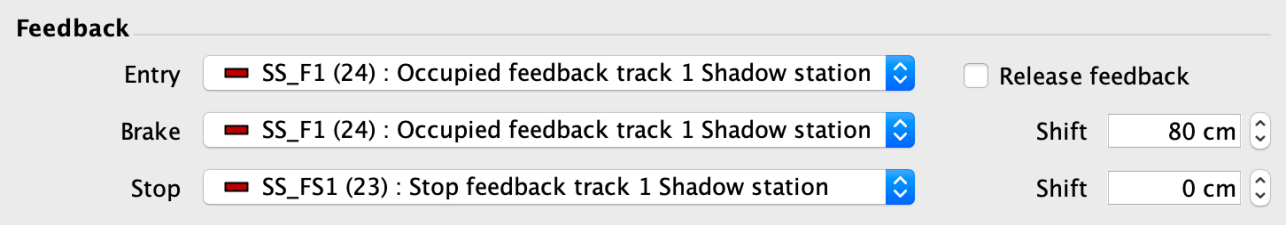
In this case we need an extra feedback at the side of the block where we want to stop (in HO about 20-35 centimeters, depending on the brake speed[49]). This feedback is called the 'Stop' feedback for this direction. In case you drive a block in two directions and want to be

[48] It is possible to stop accurately, but only with good decoders and more configuration in both the decoder and iTrain. For example, precise speed measurements have to be done for all locomotives.

[49] The brake speed is the speed of a train when it is entering the stop section. It is an option of the block.

able to stop in both directions, you need to create this stop section on both sides of the block resulting in a total of three feedbacks[50]. The feedbacks on the sides will have a different role depending on the driving direction. When driving in a block, the first feedback will be the 'Entry' feedback. The next will be the 'Brake' feedback and the last will be the 'Stop' feedback. For the other direction the same feedbacks will have another function. It is very common that the 'Entry' feedback is the same as the 'Brake' feedback and that the 'Entry' feedback in one direction is the same feedback as the 'Stop' feedback in the other direction.

First you have to specify the list of all feedbacks that are available in the block on the tab 'Feedbacks'. After that you can specify the feedback roles/functions per direction on the tabs 'Direction: Previous' and 'Direction: Next'. When starting with an empty form, it is always good to first use 'Auto fill' that makes a best guess based on how you have drawn the block in the switchboard. You only have to check if the result is what you intended and fine-tune it if necessary.



The first feedback we encounter entering from one direction is the 'Entry' feedback. You always have to fill in at least an entry feedback per direction. In some cases this 'Entry' feedback can be used as a 'Release feedback'[51] for the previous block or turnouts when going off, but only use this when you understand the precautions. With this option unchecked the block will also be released, only not on the 'Entry' feedback.

In case you want to be able to stop in this direction you have to fill in the 'Stop' feedback. Normally the 'Stop' feedback is the last feedback in the block. It is allowed to use one feedback multiple times, so in case you only have one feedback you can fill it in three times, but we recommend using positions in that case.

The 'Brake' feedback specifies the moment of braking from full speed to the braking speed (a low speed) that should be reached when the 'Stop' feedback is activated so you can stop decently. Many layouts do not have a separate 'Brake' feedback built-in per direction. Sometimes you can use the middle feedback when using three feedbacks, but in case you use fewer feedbacks it will normally be the same as the 'Entry' feedback.

It is possible to use the 'Shift' field to delay the moment of braking by specifying a distance to virtually shift the feedback specified for the 'Brake' feedback. In case the 'Brake' feedback is empty, the 'Shift' will be applied to braking on the 'Entry' feedback, but it is better to repeat the feedback in the 'Brake' feedback input so that it is clear what is happening.

---

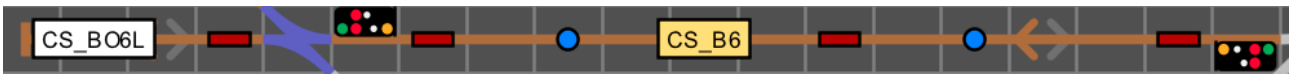
[50] Two feedbacks is also possible by not detecting the middle part and only using the side feedbacks.

[51] A release feedback will release the previous block and turnouts as soon as it goes off. For this to work properly the locs and trains need to be fully detectable and you probably want to use a 'Switch off delay' on the feedback(s) to prevent that the feedback goes off shortly, because of bad contact with the tracks.

Note: To use the 'Shift' option it is important that the Track Scale in menu 'Edit' -> 'Settings' is set correctly and that the loc speed measurements have been done to get accurate results.

It is also possible to use the 'Shift' field for the 'Stop' feedback to shift the stop point of the train, for example when the stop section is too long. When defining a stop section, it can better be somewhat too long, because you can correct the stop position in the program by using a shift. A stop section that is too short can only be corrected by lowering the brake speed, but this might give unwanted side effects.

Note: The 'Auto fill' button never fills in the 'Shift' option. This needs to be done by the user and requires some experimenting to get it right.

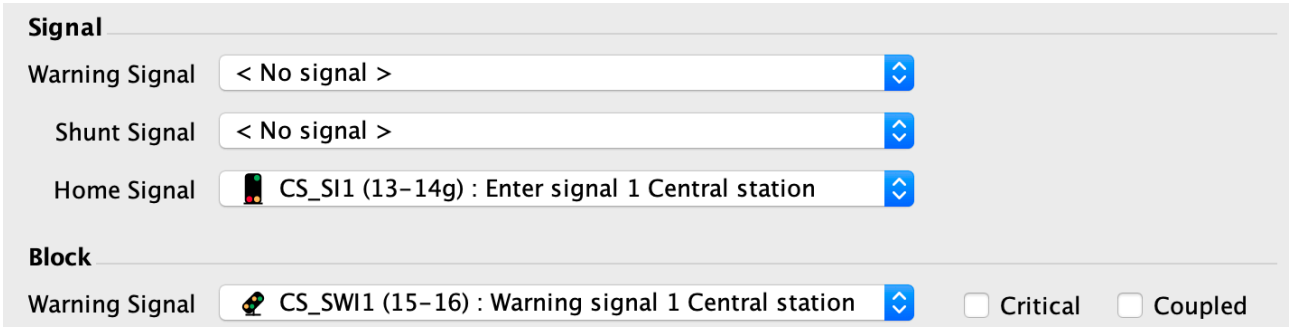


It is possible to stop in a block that only contains one feedback. This is, however, less precise than the option with a dedicated stop feedback, but in some cases there is no possibility for an extra feedback. You can use the same feedback for 'Entry', 'Brake' and 'Stop' with different 'Shift' values. If the block is too short to brake in, the last feedback of the previous block can be used to start the braking. To achieve this, leave the 'Brake' feedback empty and its 'Shift' zero. This is a natural option for short dead end blocks that are only used to park a loc or a very short train.

Note: A train will be stopped in a previous block when a block does not have a feedback assigned to the 'Stop' feedback input and a stop is necessary. This is direction dependent so you can make it possible to stop a train in a block in one direction, but do not allow it in the other direction.

Signal

If a block knows its neighbors, then it can also calculate the state of the block signal. In iTrain you can assign separate signals to each driving direction and then its state will be set automatically. It is important that you assign the signal to the block in which the train will stop and not to the block that the signal protects.



Next to the Home signal you can specify a Shunt signal that is placed in front of the home signal. The internal signal state of the block (in this direction) will then be represented by two signals. If you only have a Shunt signal you better fill it in in the Shunt field. A combined home and shunt signal should be filled in Home Signal, and the Shunt signal should be left empty.

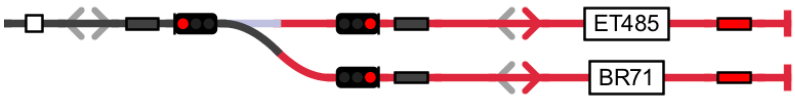
You can also assign warning signals. You should only do this if the warning signal has its own unique address (and object). We have to distinguish between two warning signals:

- A separate warning signal in the block that is related to and preceding the home signal in the same block. It should be assigned to the 'Warning Signal' box above the Home signal.
- A warning signal that is part of a combined home/warning signal and is related to a home signal in a following block. It should be assigned to the 'Warning Signal' box in the 'Block' section below the 'Signal' section.

Note: By default, only the signals that are attached to the block can be selected in the input boxes. This makes it easier to select the correct one, because it is never necessary to select one outside the block. In case the block is divided over two tabs, the selection may not be correct. The 'Show All' option will fill the input boxes with all the signals available.

Block

The section 'Block' relates to the neighboring block when driving in this direction. At first you can specify the 'Warning Signal' that informs you about the state of the signal in the block after the current one.

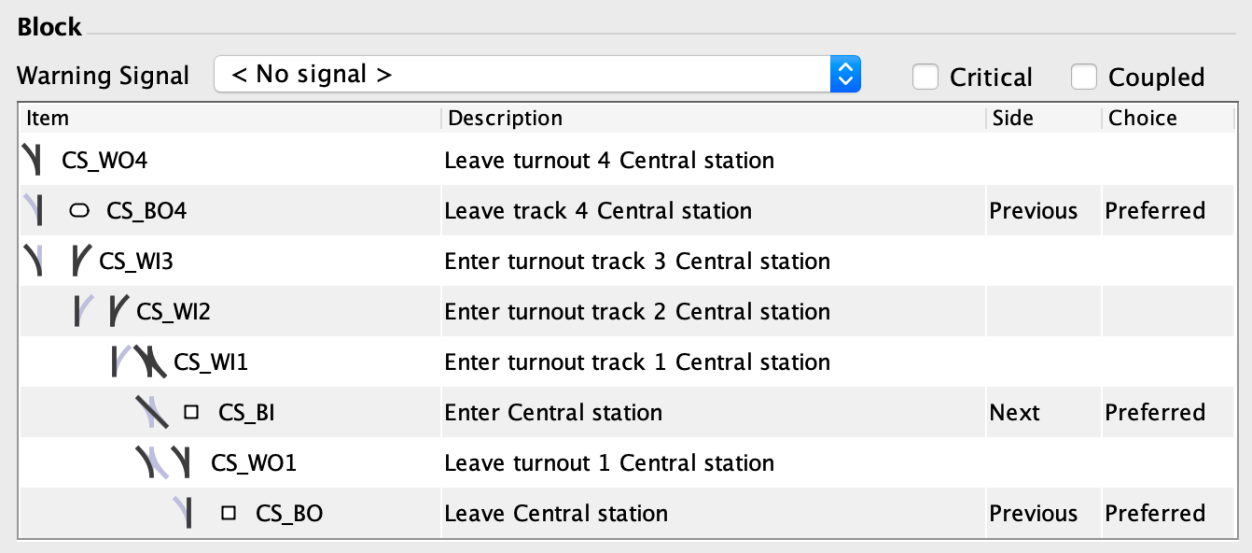


Secondly there is a small box 'Critical' with important functionality. If this block is marked 'Critical' it means that this block plays a critical role in the reservations to prevent deadlocks. The block will only be reserved if the block after this block can also be reserved in one action. It should, for example, be used on a single track in the direction of a terminus to prevent a train from reserving the single track before the station if it is not possible to enter the destination block(s) in the station. By reserving the single track before the station, the trains in the station would have no possibility to leave the station anymore, causing deadlock.

Note: In general, it is not necessary to use 'Critical' on a single track divided into two or more blocks that are bi-directional. The program will automatically detect this situation and reserve all the blocks until the end of the single track or until another reserved block in the same direction is found. Therefore it is not possible to do reservations on the single track in another direction anymore. However, it is still allowed to do reservations for other trains in the same direction that are behind so that multiple trains may follow each other on a single track and use the full capacity.

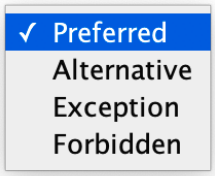
Connections

The program needs to know what the neighboring blocks are for every block, for example, to follow the trains over the layout, set the signals correctly and calculate routes. Starting in a block and following one of the two directions it will optionally pass some turnouts and come into other blocks. The path to these other blocks including all the turnouts are specified in the tree at the bottom.



In the block connection tree above, for every turnout only the states of that turnout that result in a connection to another block are shown. Next to the block name and description also the side of the other block connected to this block is shown in the column ‘Side’. All this information will be derived automatically from what you have drawn in the switchboard . It cannot be edited manually in the table, to prevent errors that are hard to find. If the data is incorrect, you can use ‘Auto fill’ or the button ‘Connect’ that will be explained at the end of this section.

The only column you can edit in this table is the column ‘Choice’. This only applies if there are multiple paths via turnouts from this block to neighboring blocks. Normally one is preferred in this direction and the others might be alternatives or even exceptions. In some cases you never want this path to be taken and you can select ‘Forbidden’.



The option ‘Coupled’ indicates that the block and the turnouts are coupled so that separate release of the turnouts and the block is not possible. This must be used in a Dinamo system if one or more turnouts get power from this block card.

To create the connections between all blocks in one switchboard[52], use the button ‘Connect’ - with the magnet - on top of the editor. It will show a dialog informing you about the changes made or a message that there were no connections to update. It is recommended to use this regularly after making changes that might affect the block connections.

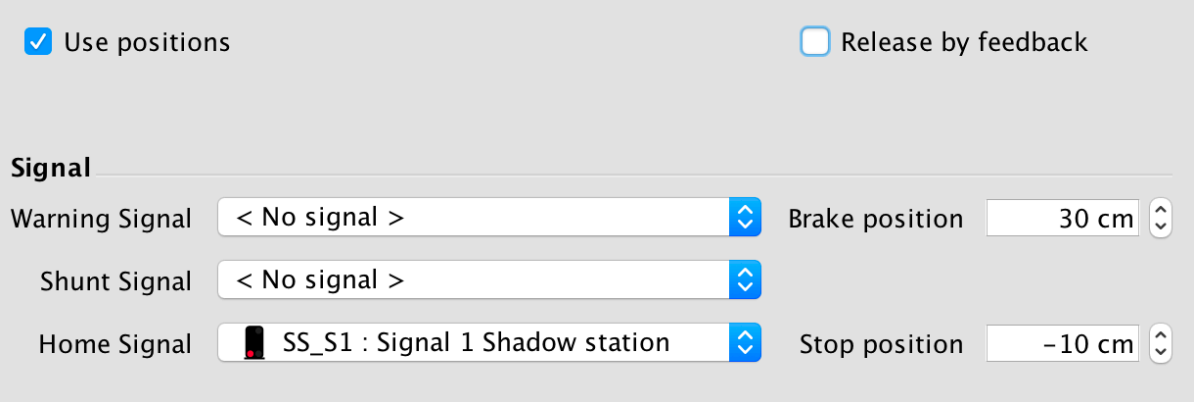
Note: In case the block connections change for a block, the ‘Choice’ column for that block will be reset to ‘Preferred’ for all rows. So first make the connections right and refine the choices when the block connections are stable.

Positions

With ‘Use positions’ you can stop at an absolute position in the block. When using positions, no feedbacks are displayed because their role will be calculated automatically

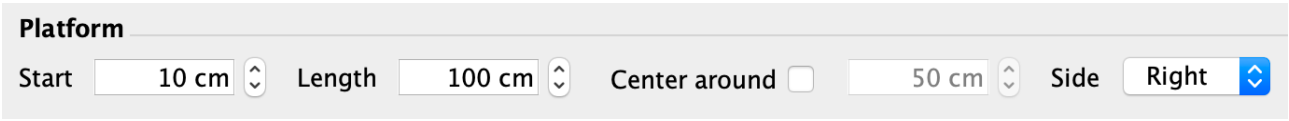
[52] The button ‘Connect’ only works on the selected tab of the switchboard editor.

now, based on the information on the tab ‘Feedbacks’. For example, all feedbacks after the actual stop position will act as emergency breaks.



Two extra fields appear in the section ‘Signal’ to indicate where to start braking and where to stop. The field ‘Brake position’ is located next to the warning signal, because that is where you normally start braking. The field ‘Stop position’ is located next to the home signal, because that is where you normally want the train to stop.

We have to distinguish two positions to stop. One because the signal is red and the train is not allowed to pass it. In this case the train should stop before the signal. In the box ‘Stop position’ you can enter the position at which the train should stop for a signal, using a positive value when measuring from the start of the block or a negative value from the end of the block.



The other position is a scheduled stop in a station independent of the signal and should preferably be along the platform. If the block is of type ‘Station’, a ‘Platform’ section will appear at the bottom to specify the platform. The ‘Start’ is the position of the platform from the start of the block and the ‘Length’ is the length of the platform itself. The ‘Center around’ box specifies the relative position of the exit on the platform. By default this is in the middle of the platform, but by checking the box before the value you can adjust it (positive value relative to the start of the platform and negative value relative to the end of the platform). A train will stop centered around this point, unless this would mean a part of the train would not be along the platform. In that case it will be shifted so that is always along the platform, but as close to the exit as possible. The field ‘Side’ indicates on which side the platform is when entering the block in this direction.[53]

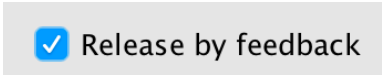
A train needs to brake in case the speed of the train should be reduced in the block or the train has to stop. Normally the driver knows this when he sees the warning signal. Instead of using a feedback you can specify a position in the field ‘Brake position’ after the field ‘Warning signal’ to indicate where all trains should start to brake. The new speed or stop should be reached at the

[53] The side of the platform will be used to open the correct doors in a train if there are different functions for both sides.

stop position. iTrain will calculate the deceleration profile necessary to smoothly decrease the speed to the desired level.

When using positions, you can even work with one feedback[54]. So the release will by default be done based on the calculated position of the train in the block. The previous block and turnouts will be released when the calculated distance travelled in the block is more than the total length of the train.

Note: There is a potential danger that the train might stop too early in the block because of dust or other bad contact with the tracks and still occupies a turnout or a previous block. The program thinks the train is still running and the calculated distance in the block might release previous blocks and/or turnouts after some time. So to use positions safely you have to guarantee proper and good contacts with the track.



In case you use multiple feedbacks per block, you can decide to only release after reaching a real feedback and not only release based on the calculated distance travelled in the block. In practice this means that there will only be released when a feedback has been reached that has a position larger than the total length of the train. This makes releases safer, but they will be delayed more. How much more depends on the number of feedbacks and their position in the block.

The only issue is when a train makes a normal stop at a position further in the block than the total train length, but the last activated feedback still did not release anything, because at that time the train was not completely in the block. In that case the release will be based on position again, but always with the condition that all the feedbacks before the stop position need to be passed.

To prevent that case, put a feedback just before the stop position of most trains in that block, or at least after the train length of most trains. This is actually similar to the classic approach with a dedicated ‘Stop’ feedback.

Note: You could consider switching from classic mode to using positions with all its benefits. If you use the option ‘Release by feedback’, then you will not lose any safety.

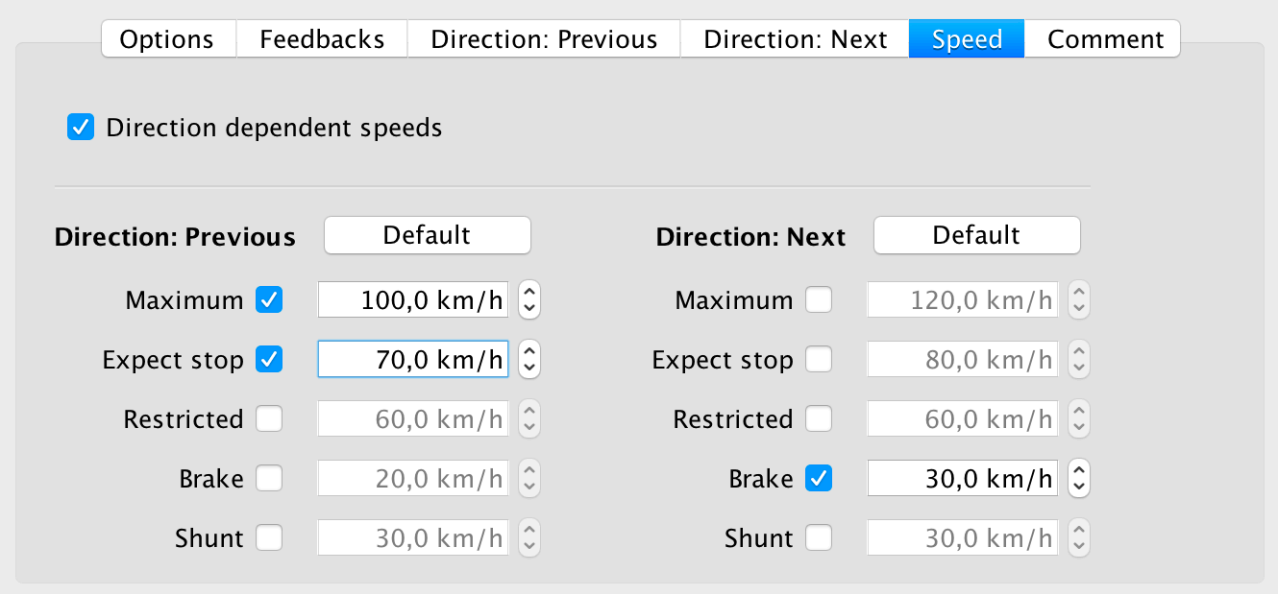
The calculations work best if your loc has cruise control and the mass simulation in the decoder is set as low as possible. In that case the train will stop pretty precisely at the same position in a block, but probably not immediately at the specified position. Without a correction in the calculation most locs would react too late and the train will run further than specified. Therefore you can set an individual ‘Reaction delay’ per loc (on the ‘Options’ tab of the loc editor). The higher the value the earlier the loc will stop. By default the value is set to 200 ms, but it should be determined by trial and error and will probably be higher in practice.

Speed

The default speeds in the blocks are specified in the general ‘Settings’, but these can be overridden per block on the tab ‘Speed’. First you have to choose if you want to be able to specify different speeds per direction in the block or not, via the checkbox ‘Direction dependent speeds’. In case ‘Direction dependent speeds’ are checked, two columns will appear with in the left column the speed in the direction ‘Previous’ and in the right column the speed in direction ‘Next’.

---

[54] You will benefit from using multiple feedbacks, because the positions will be corrected at every feedback.



Now you can override the speed per category by checking the box before the speed. An unchecked box will show the speed value that will be used based on the default speeds. By using the 'Default' button you can set all speeds to default at once.

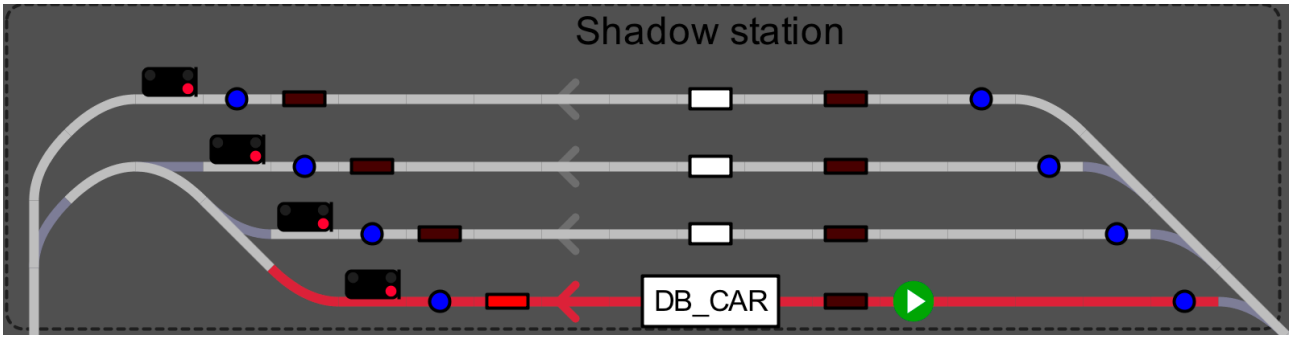
The different categories are:

- Maximum - the maximum speed in the block if there are no other restrictions from signals.
- Expect stop - the maximum speed when leaving the block if the train has to stop in the next block.
- Restricted - the maximum speed if the signal shows the restricted sign, for example when a turnout is branched in the path to the next block.
- Brake - the speed to which a train brakes in case of a red signal until it enters the stop feedback.
- Shunt - the maximum speed if the signal shows that only shunting is allowed.

Note: You can specify a speed per turnout state as well. If all turnout states in the path have a specified speed, only the minimum of these speeds will be taken. If at least one turnout state has the speed 'Restricted', then the restricted speed specified in the block is also used in the calculation of the minimum speed.

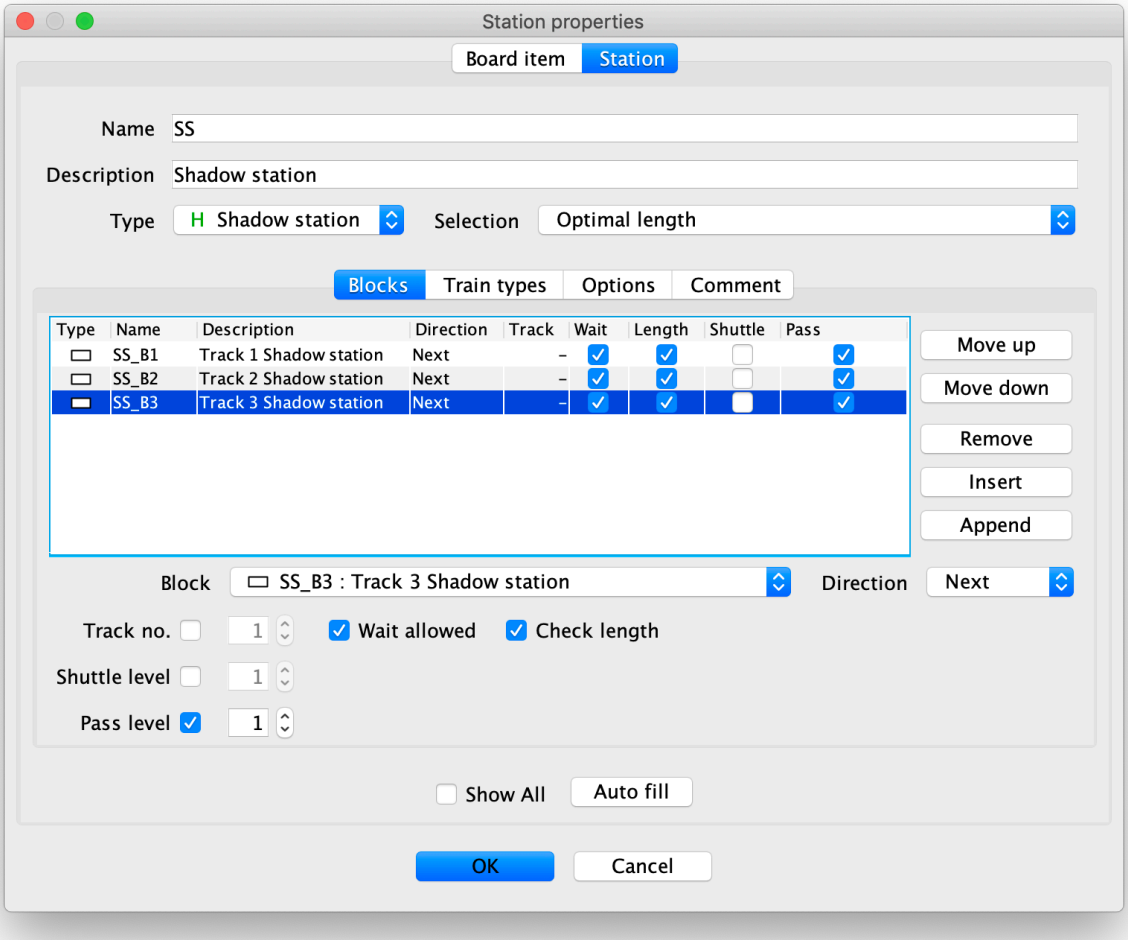
Stations

A station is a destination where trains can stop. It contains a set of blocks that belong together. It can be used in routes that are described later on.



To create a station, select multiple cells in the switchboard and double-click the station button in the toolbar or use the keys ‘Shift’ + ‘S’. This only creates the ‘Station’ element. To create the ‘Station’ object you have to double-click in the element so you can edit the station properties.

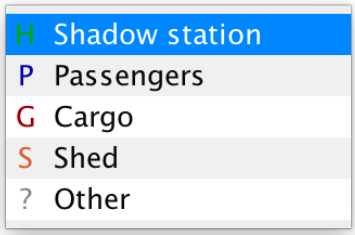
The station has a type that indicates what the purpose is of the station and what

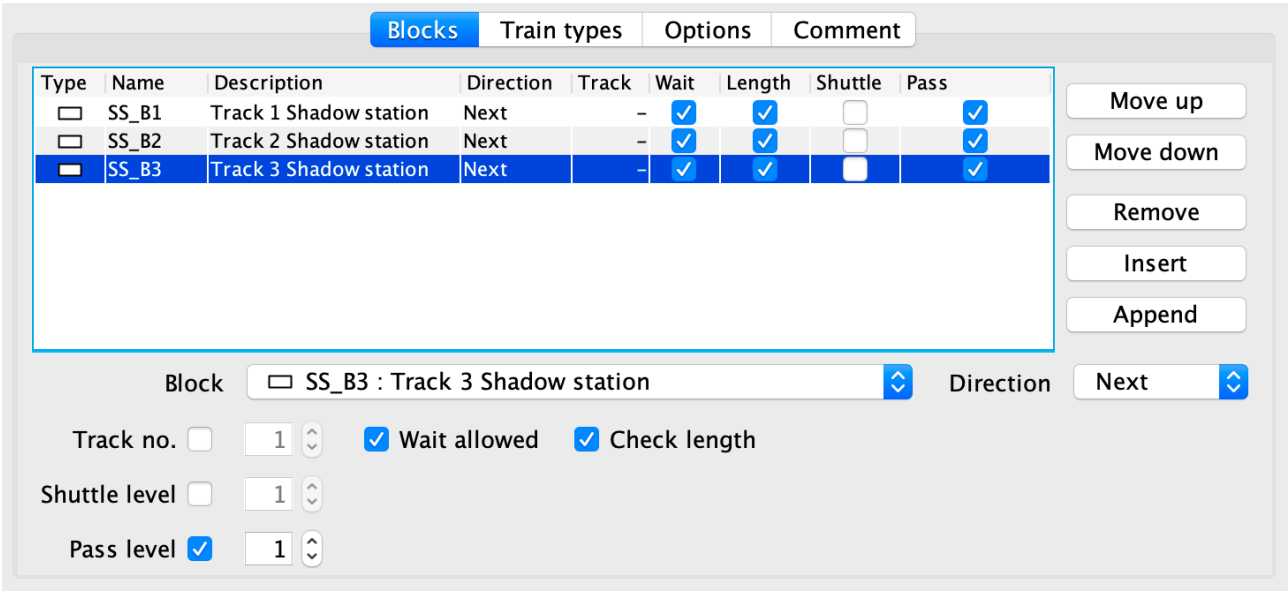


kind of blocks can belong to the station. The property ‘Selection’ is used when driving automatically without routes to specify the order of selection of the blocks in the station. The selection options are described in the chapter ‘Train routes’.

Blocks

The list of blocks can be filled automatically. In that case it will include all the blocks of a type that have a block element (white box) in the area covered by the station and matches the station type. A ‘Shadow station’ matches blocks of type ‘Station’ and ‘Siding’; ‘Passengers’ only matches blocks of type ‘Station’, and ‘Cargo’ and ‘Shed’ only match blocks of type ‘Siding’. It is also possible to add and remove blocks manually via the ‘Block’ input and the buttons on the right of the table.





Every block in a station has some additional properties to specify its purpose in the station, and some suggestions when to select a specific block. Some of the properties are directly visible in the table with blocks and others are only visible at the bottom of the table when a block has been selected.

The column ‘Track’ or the field ‘Track no.’ specifies the track number. This track number is optional and can be used to indicate that two blocks, for example B3A and B3B, belong to the same track ‘3’ in the station and could be used together for longer trains.

The column ‘Wait’ or checkbox ‘Wait allowed’ specifies if this block is meant as a destination for a train to stop and wait. However, if a block is meant to pass the station you can indicate that by selecting the column ‘Pass’ or the checkbox ‘Pass level’. The column ‘Shuttle’ is only a suggestion that this block is meant for shuttle trains, so that the shuttle trains will prefer to choose this block.

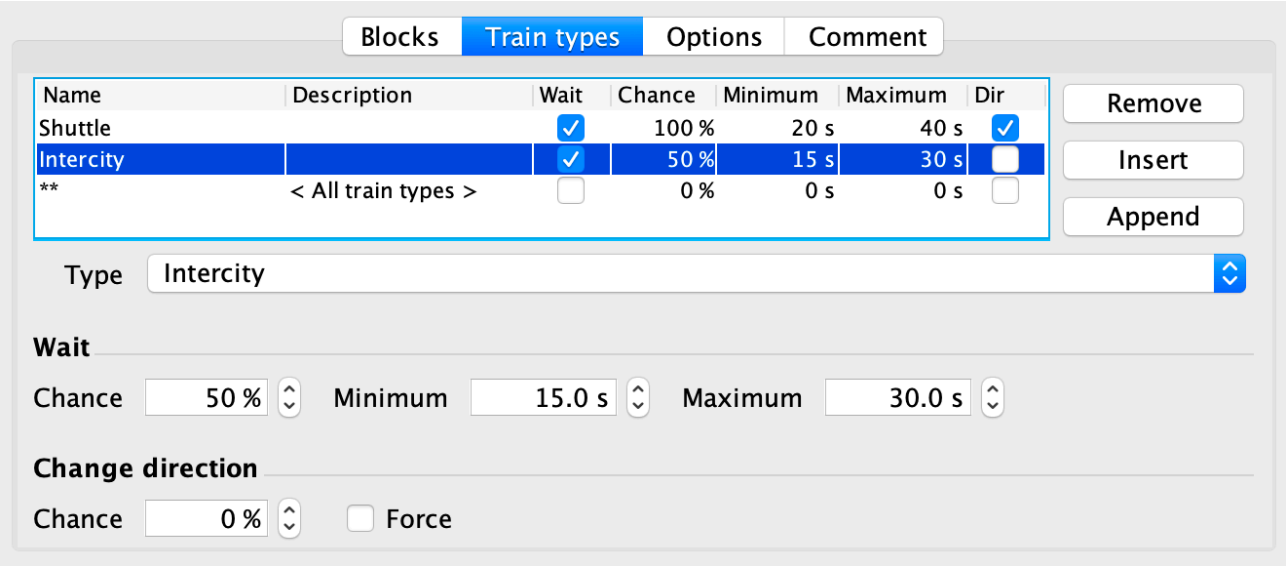
The column ‘Length’ or the field ‘Check length’ indicates if the train must fit in the block to allow it to stop here.

The fields ‘Shuttle level’ and ‘Pass level’ are directly related to the columns ‘Shuttle’ and ‘Pass’. You can select them, but when selected you can choose a level in addition to specifying the preference, the lower the value the more preferred (so 1 is the primary and 2 is the secondary level). For example, if multiple blocks are selected for shuttle trains, you can select which one should be selected first by giving it number 1. Equal numbers are also allowed, so if it doesn’t matter just use 1 for all levels.

Note: The options 'track number', 'pass level' and 'shuttle level' are meant for driving automatically without routes, because in a route you can specify exactly where to stop a train in the station by specifying the blocks.

Train types

A specific station is a destination for some trains, but for others it is not a destination at all or only sometimes. On the tab 'Train types' you can specify the relation between a train type and the station for trains that drive automatically without a route.



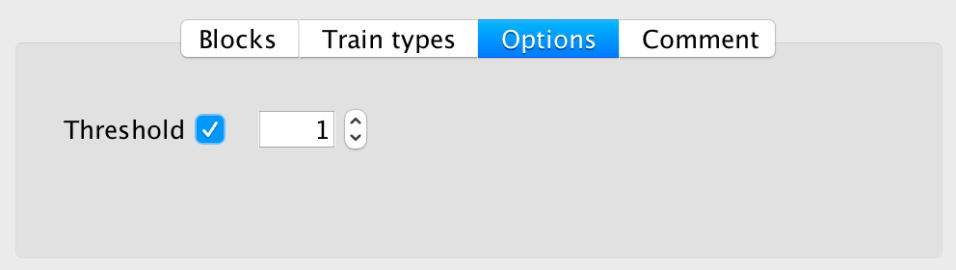
For every train type you can specify if it is allowed to wait in a station and what the chance will be that it will wait. In case it waits, you can specify the minimum and maximum time it will wait. This time will be chosen randomly every time the train decides to wait in the station.

In case the train waits in the station, you can also specify the chance that it will change its direction. By clicking the checkbox in the column 'Dir' you can toggle this value between 0 and 100%. Only trains that are allowed to change direction, because they have a locomotive or control car on either side, will be allowed to change direction. For other trains the change of direction is ignored unless you check 'Force' to allow any train to change direction here.

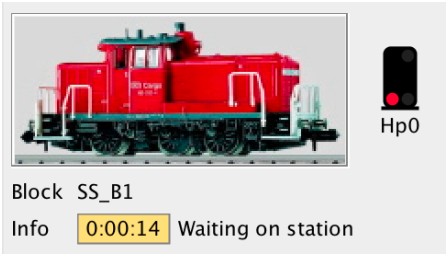
Note: If the train type is not specified at all, the train will avoid the station and will not attempt to pass through it.

Options

On the tab 'Options' there is just one extra option 'Threshold' to regulate the number of trains leaving the station. The threshold value specifies the number of free blocks that are allowed in all of the station blocks marked for 'Wait' (excluding the inactive blocks).



Using a value of 1 simulates the traditional shadow station in which a train can only leave when another train enters. The only added condition is that after entering the shadow station there are no empty (active) blocks anymore so that the leaving train will only create one empty block again. A higher value N for the threshold will allow more trains to leave the station to create N empty blocks as a maximum.



Only trains that are driving automatically will participate. After their normal waiting time in the station is over, they will continue to wait until the station allows them to leave. In this case the waiting time increases to distinguish it from the normal waiting time in the station.

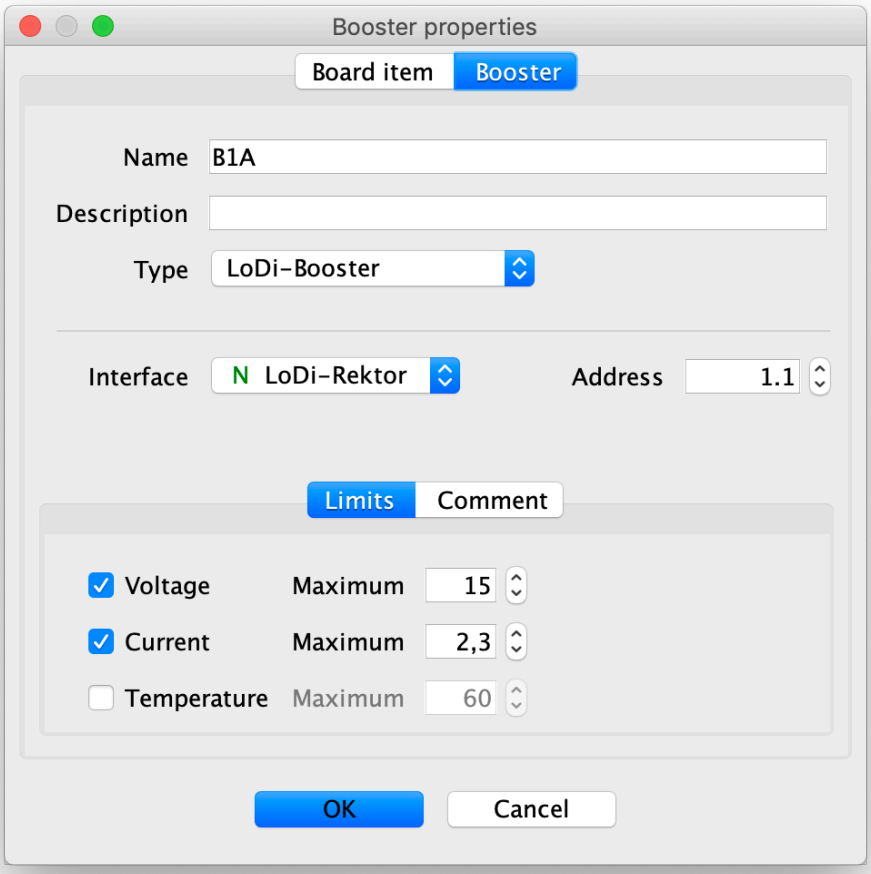
States
A station can be in one of the following states:

- Normal - indicated by a dashed black line. Normal operation as specified above. Double clicking in the station will restore this state.
- Finish - indicated by a solid black line. Trains visiting the station for a stop will not continue afterwards anymore. Trains already in the station will not leave anymore. You can select it by clicking in the station while holding the Alt key.
- Pass - indicated by a dashed gray line. Trains will not choose the station anymore to wait in it and will pass or skip it.[55] You can select it by clicking in the station while holding the Shift key.

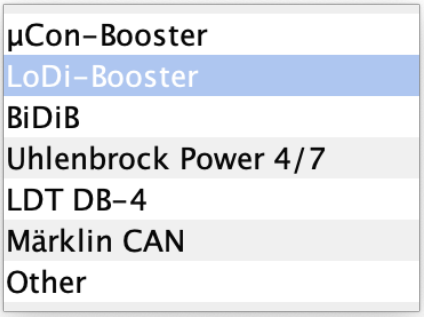
Boosters

A booster is an amplifier of the track signal created by the command station. Sometimes the command station has a built-in booster as well so you can start without additional external boosters. The booster objects in iTrain are only related to external boosters that can be individually switched on and off and report individually when they are in alarm state (short circuit or overloaded), now called intelligent boosters. If you don't have intelligent boosters, you don't have to define boosters in iTrain, because they will not be functional.

[55] Trains that have already taken off with this station as a destination will continue as they cannot suddenly change their plans while driving.



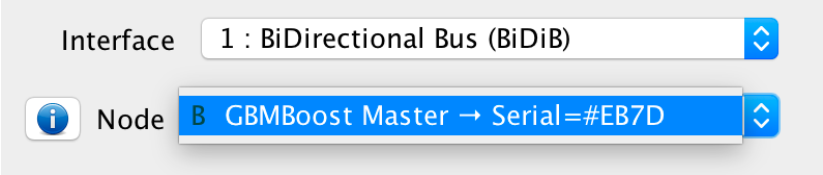
To create a booster object you need to specify the name and type. Depending on the type, you will get other options.



µCon-Booster
The *µ*Con-Booster is always used in combination with a *µ*Con-Manager or LoDi-Rektor. You need to specify the address of the booster and the interface of the Manager or Rektor. Because every *µ*Con-Booster contains two 2.5 A boosters that have to be addressed separately, the address can be specified as 1.1 for booster 1A and 1.2 for booster 1B.

LoDi-Booster
The LoDi-Booster is always used in combination with a LoDi-Rektor. You need to specify the address of the booster and the interface of the Rektor. Because every LoDi-Booster contains two 2.3 A boosters that have to be addressed separately, the address can be specified as 1.1 for booster 1A and 1.2 for booster 1B. The current LoDi-Boosters also support the temperature.

BiDiB
The BiDiB-Booster should be selected in BiDiB systems.

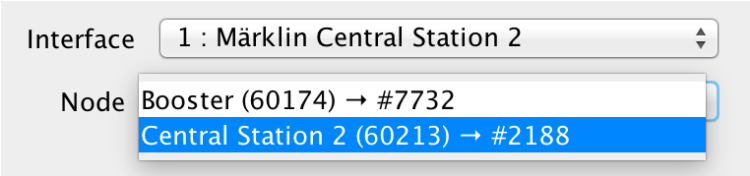


The node can be specified from a dropdown list with all active boosters. You can use the Identify-button to select the physical booster in BiDiB when the system is online.

Uhlenbrock Power 4/7
The Uhlenbrock Power 4 and 7 can be used as intelligent boosters when they are used with LocoNet and configured properly. You have to define a relay in iTrain that allows you to switch the booster on and off via an accessory address (LNCV 8 of the Power 4/7). In case of overload or short circuit the relay will be switched and so the booster object will follow. There is just one disadvantage that only two states are available, and depending on the type of the relay the booster will show `on` and `off` (for an on/off relay) or `on` and `alarm` (for an A/B relay).

LDT DB-4
The Littfinski DatenTechnik (LDT) DB-4 can be used as an intelligent booster. You need to create an `on/off` relay to switch the booster manually and a feedback to notify that the booster is in alarm state. Both the relay and the feedback have to be attached to the booster. Read the manual of the DB-4 for more details on how to configure the address of the relay and the feedback.

Märklin CAN
Both the internal booster of the Märklin CS2/CS3 and the Booster 60174/60175 can be used as an intelligent booster. This booster support is still a little bit experimental, because there is no official documentation how to implement this.



A booster can be selected from the list with available boosters called nodes. You can distinguish similar boosters from the serial number shown with a # that is equal to the last four digits of the serial number on the bottom of the device.

Diagnostics

For every booster you can specify which diagnostics are available together with their limits. For the voltage you have to specify the upper limit that is still a safe value in Volt. For the

current supply the maximum current it can deliver in Ampere. For the temperature, specify the value in Celsuis after which the device really gets too warm.

Blocks

After defining the booster itself you have to attach it to all the blocks that are powered by this booster in the section ‘Block control’ on the tab ‘Options’.

Switchboard

On the switchboard the booster elements can have three states: `off`, `on` and `alarm`. The red alarm is only shown when the booster reports an issue. You can switch manually between the `on` and `off` state by clicking the element.

When you enlarge the element with Shift + Alt + cursor keys so that it spans multiple cells in width, it will show the diagnostics as well on the right side with colors. The most left cell will still show the overall status and you can click the whole element to change the status.

The state of the booster will be visible via the border of the block element. If the booster is switched off it will be gray, and if it is in alarm it will be blinking red.

Control

Switching off a booster section or having an alarm situation in a booster will prevent blocks belonging to this section to be reserved. Existing reservations will remain, but signals in front of the booster section will turn red immediately to stop trains before the section if possible.

Aspect

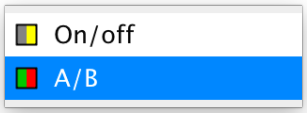
An aspect is a basic element to switch between 32 states. By default the states are called ‘Aspect 0’ until ‘Aspect 31’. The element in the switchboard will show the number. Aspects can be used as general accessories with many states, but their original purpose was to display numbers on the layout, for example in speed displays.

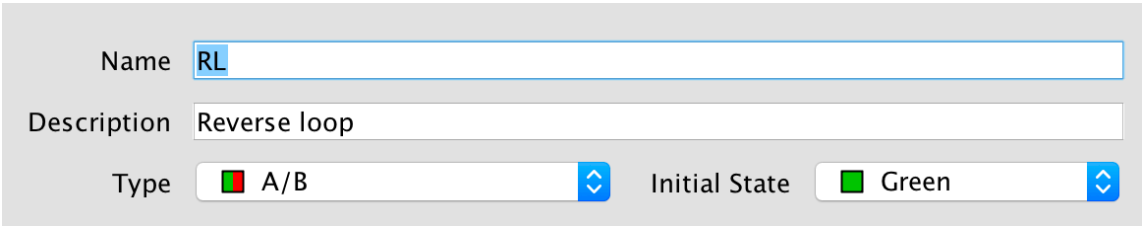
Note: Only use an aspect if there is not a more specific accessory type for your task.

Tip: You can edit the names for every aspect state in the state mapping table. This way it is easier to remember its purpose if the state does not represent a value.

Relay

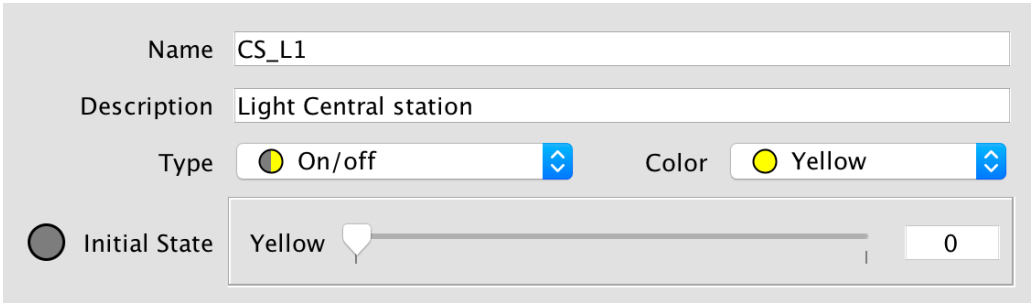
A relay is a basic element to switch between two states. States can be `on` and `off` or `A` and `B` depending on the type of relay. A relay is used if there is no more specific accessory available (such as Light, Sound, Decoupler or Crossing).





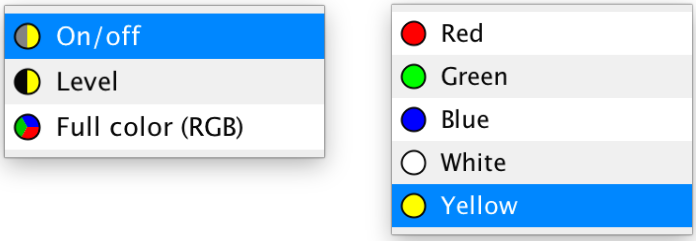
A special application for a relay is the reverse loop to change the polarity of the tracks for 2-rail layouts. This has been described in an earlier section, but can be achieved by attaching a relay of type `A/B` to a block.

Light



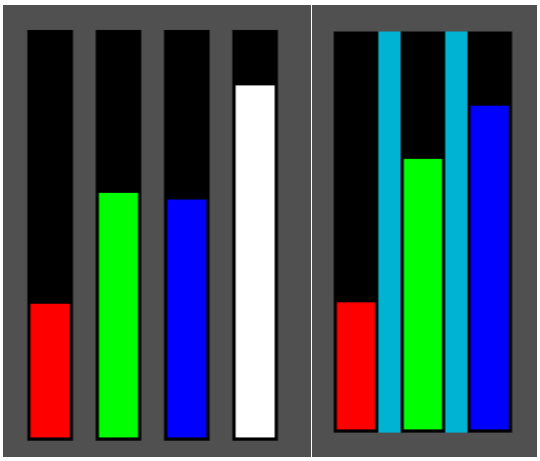
The light element represents the simple light switch that can turn the light `on` and `off`, but it also represents a dimmer that can set the `level` of the light (if supported by the hardware). In addition there is the full color (RGB) type that can only be used with specific hardware that supports at least three level outputs.[56]

[56] iTrain allows level outputs of 10 bits (0-1023), but this can currently only be achieved with the LoDi-Shift-Commander in combination with the module 4-C-LED. With DMX there is support for 8 bits so the lowest two bits are ignored.



For the first two types (`on/off` and `level`) of light you can specify one of 5 colors that are typically used for LEDs to distinguish them in the switchboard.

There is no state mapping available for light. For simple `on/off` light you can specify after the address if the on state is selected via `Green` or `Red`. When working with aspects, there is a default mapping with aspect 0 for `off` and aspect 1 for `on`. In case of type `Level` the aspect indicates the level. In all other cases the hardware is directly controlled and only specifying an address or port of a module is enough.



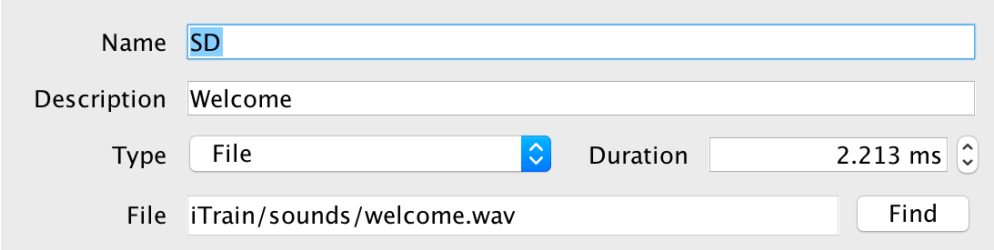
When working with light levels it is useful to enlarge the light element in the switchboard to show level bars that can be adjusted with the mouse. The type 'Full color (RGB)' will show three bars together with the resulting color between the bars.

Sound

The sound element is an abstraction to cover the two types of sounds available in iTrain:

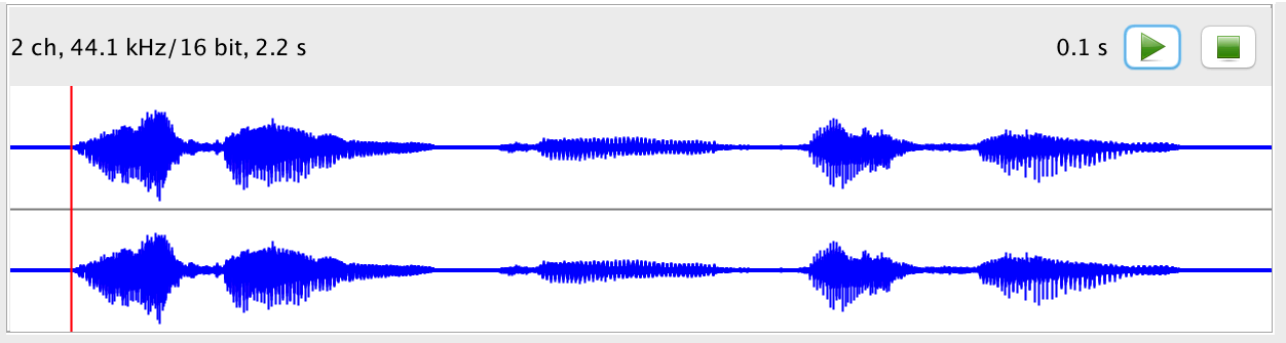
- File - play a file on the computer via speakers connected to the computer.
- Decoder - control a decoder that plays sound via a loudspeaker in the layout itself.

Both types are totally different in how they are controlled, but can be used identically in the rest of the application.

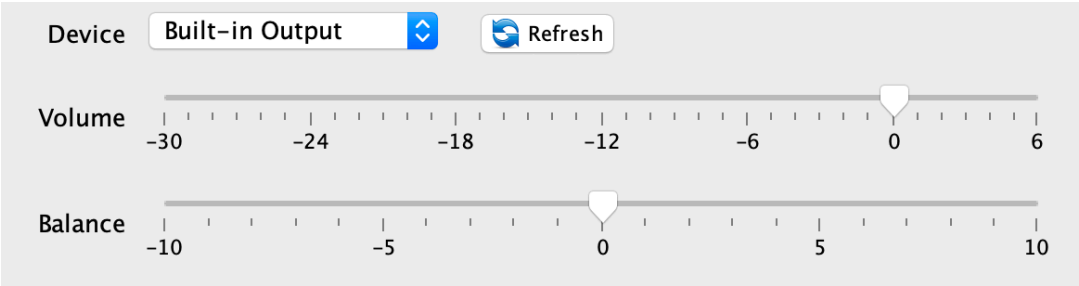


File

To create a sound that is played on the computer, you have to select a file. Use the 'Find' button to open the resource browser to select a file of type `.wav`[57]. The sound resource browser contains a preview with information about the sound file on the left (number of channels, sample frequency, bit depth and duration) and the waveform at the bottom.



You can listen to the sound by using the play and stop buttons on the right. A vertical red line indicates the current position. To find a specific spot you can click on to the waveform (even when playing). The corresponding time offset is shown to the left of the play button.



Three additional fields are available when working with sound files:

1. Device - to select a sound device connected to your computer to play the file.
2. Volume - to adjust the volume (in dB) of the sound file in relation to others.
3. Balance - to change the balance between the two channels (no effect on mono files[58]).

Decoder

In case of a decoder you have to select an address or port on a module to access the external sound device, and you can use the 'State mapping' to exactly define the outputs as with most accessories.

---

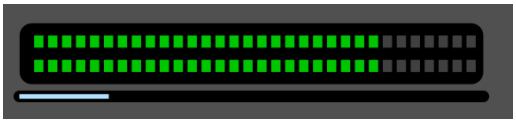
[57] Depending on the operating system some other formats may also be supported like .aiff and .au, but it is recommend to use only .wav and first convert other formats like mp3 to this format via other software.

[58] There is no panning for mono sound files. This is a limitation of the current Java sound implementation.

Tip: In case of a sound file it is still possible to select an address to simultaneously switch something when the sound is playing. This can be used to turn on an LED or other physical indicator.

Control

The sound element in the switchboard will show a thin time bar at the bottom to indicate the time progress in playing dependent on the total duration. You can click on the element to start and pause the sound, and double-click to stop the sound. A sound of type 'Decoder' cannot be paused and will stop directly.

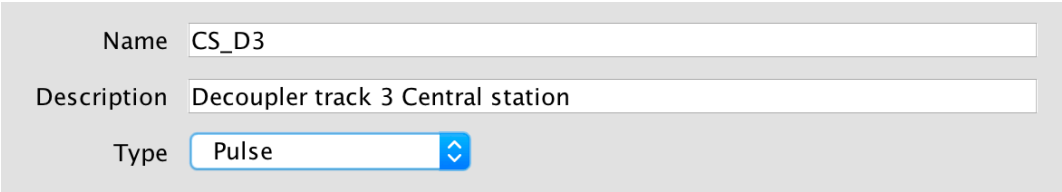


If the sound element is of type 'File' and it is enlarged in the switchboard, then it will show level meters (two for stereo). These level meters show an average of the sound level that is currently playing. If you click on or just below the time bar, it will change the position in the sound file immediately (so also while playing).

Decoupler

The decoupler is an element to separate vehicles (locomotives and wagons) from each other in a train. A decoupler has two states: `active` and `inactive`. There are two types of decouplers, determined by the way they are controlled by the interface:

- The type `On/Off` works as a relay, you either set the decoupler `active` or `inactive`.
- The type `Pulse` uses only one output of an address instead of two (also called 'Half' in the address usage) and originates from the Märklin decouplers with a decoupler attached to only `red` or `green`.

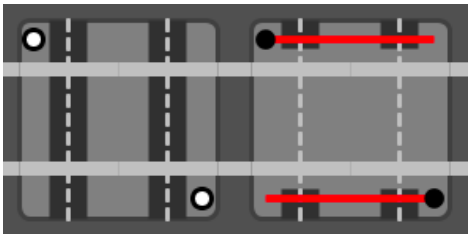


The type `On/Off` is preferred in automatic control. Other decouplers can be converted to this type by using a physical relay to switch between the states, so they are not using the switching current to keep them active.[59]

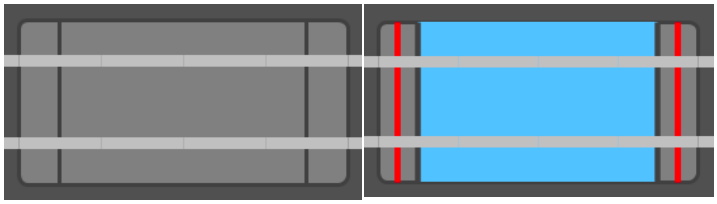
Crossing

The crossing is an element to indicate that the railroad crossed some other road type or a bridge. There are two types of crossings available. The first is the railroad crossing and the second one is the bridge. In both cases the states will be `open` or closed, but they differ in how they influence the train.

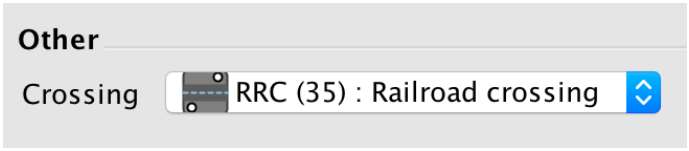
[59] On older systems activating a decoupler draws a lot of current and does not allow other accessories to be switched as long as the decoupler is active.



Normally a railroad crossing does not influence the train too much, but it only takes care it will be closed for cars when a train is expected to pass.

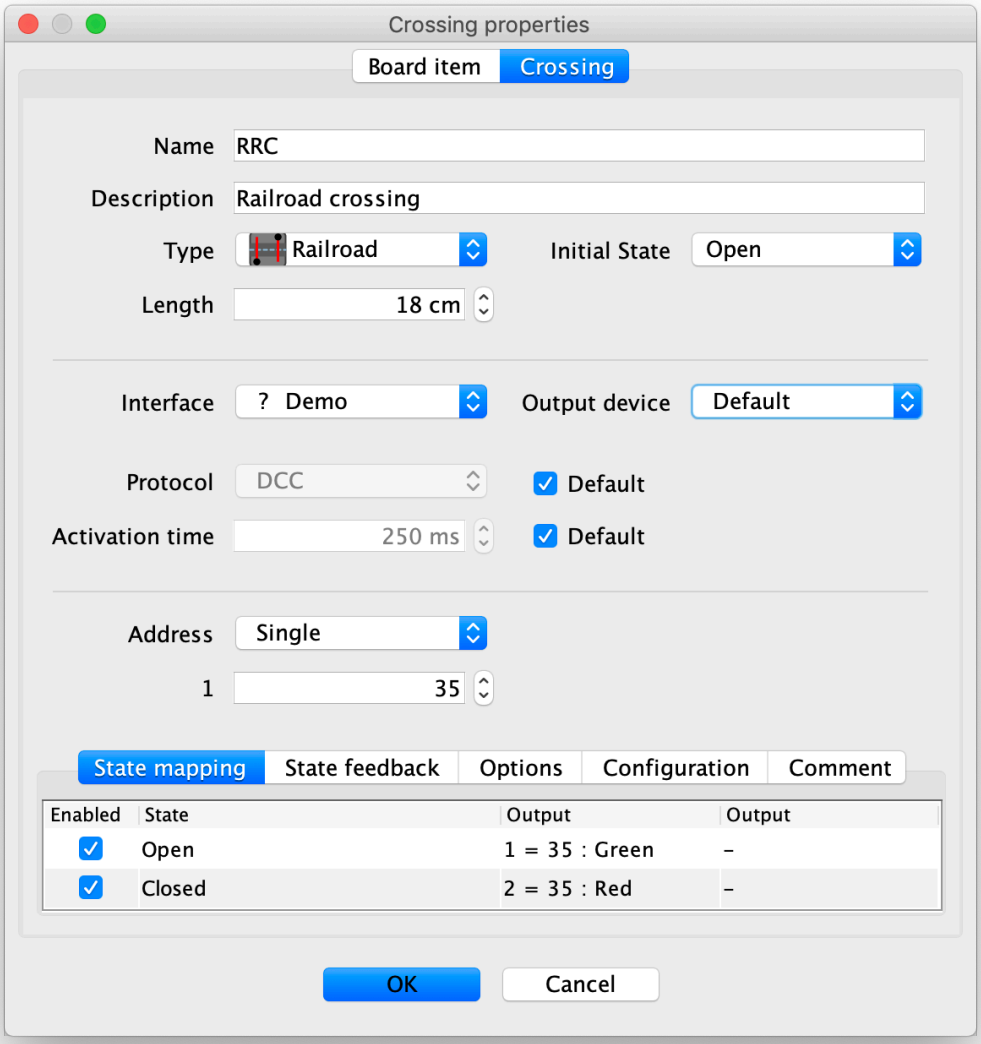


In case of a bridge the train is not allowed to pass as long as the bridge is open. It will show water in blue and a red line indicating it is not allowed to pass that line.

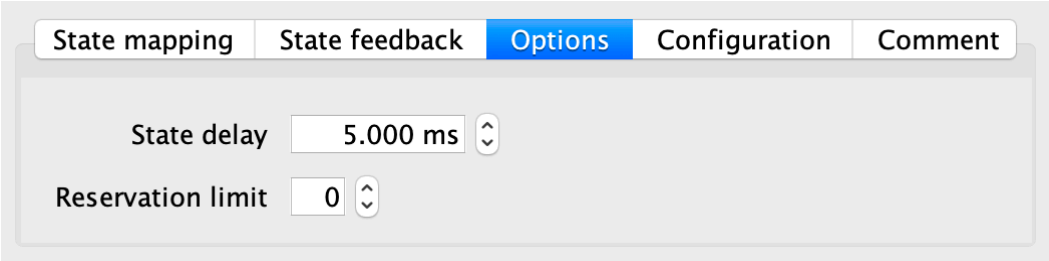


The crossing is a board element without track elements. You have to add the track elements yourself. This allows you to add feedback or block elements as well on top of the crossing. You can draw as many parallel tracks as you like on top of the crossing, but you have to specify the crossing in the tab ‘Options’ of all the blocks that are on the crossing to make it work, because this will not be done automatically.

Tip: It is recommend adding the railroad crossing to shorter blocks so that it will not close too early or open too late, because as long as one of the blocks is reserved or occupied it will be closed.



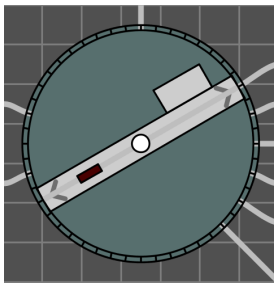
The length specifies how long the track on the crossing is. It is not used yet, but may be used in future versions. The other options are equal to the other accessories.



On the tab ‘Options’ you have to fill in the ‘State delay’, because it specifies the time it takes to go from one state to the other. In case of a crossing this normally takes seconds and an intermediate state is shown to indicate the crossing is in a transition.

The second input, ‘Reservation limit’, is only visible in case of a railroad crossing. It indicates the number of reservations in front of the train that will cause the railroad crossing to go the closed state. A number of zero means that the railroad crossing will only be closed when the block with the crossing has been occupied itself.

Turntable



- The turntable is an element to connect multiple tracks together by means of a turning bridge. This way it is possible to turn the loc. This is commonly used for steam locs. On the toolbar the turntable is just a small element, but it has been designed to be a much bigger element on the switchboard, covering multiple cells in both directions. Depending on the number of outputs used it can be useful to use 4x4 cells up to 12x12 cells for all outputs, but there is no limit.

The track connections have to end in the middle of a neighboring cell.
Just enlarge the turntable until it looks right by selecting the turntable and using Shift + Alt + cursor keys to size. It is also possible to turn the whole turntable in both directions with the ‘R’ and ‘T’ keys (counter clockwise and clockwise).

The turntable element in the switchboard does not just represent the accessory turntable, but also the block and the occupancy feedback. When you double-click on the white block element, you edit the block properties of the associated block with its own type ‘Turntable’. When you double-click on the feedback item, you edit the feedback like any other feedback. Clicking elsewhere on the turntable will edit the turntable accessory.

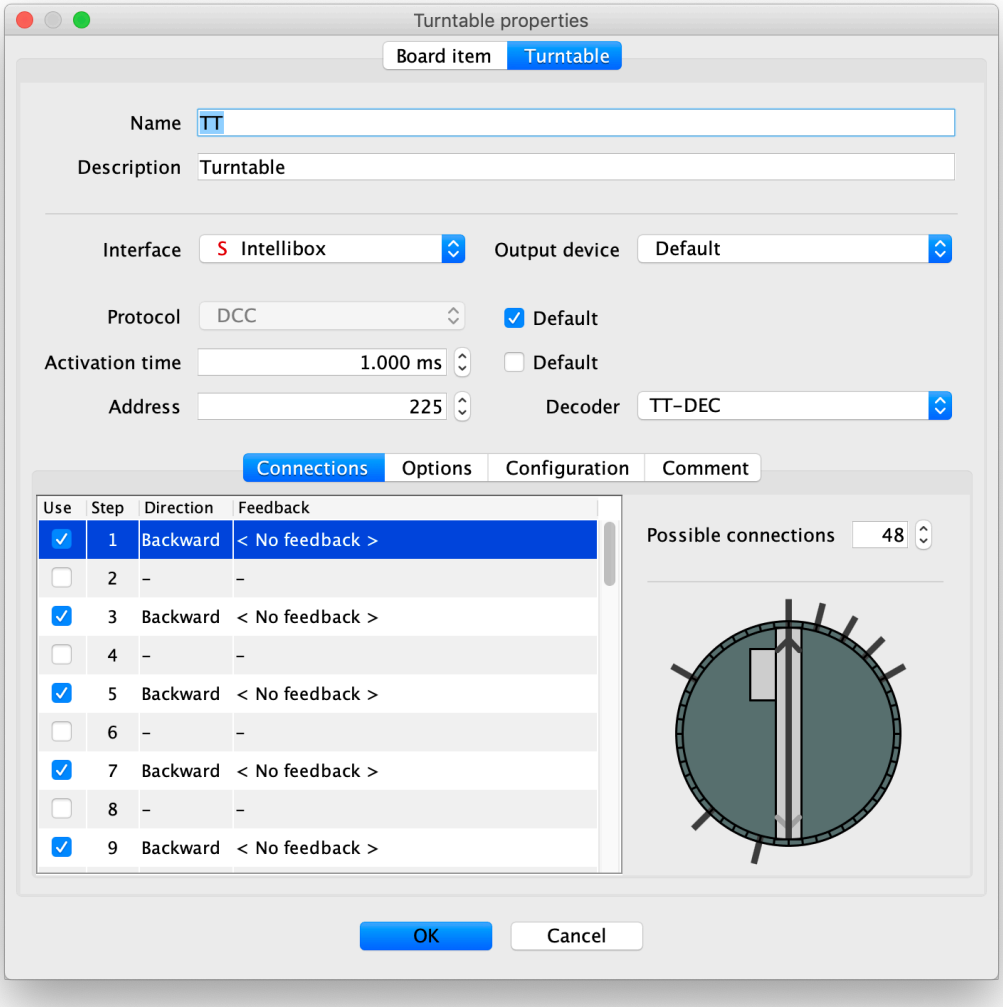
Decoder
There are different types of turntables, but what is most important is the decoder that controls the turntable. The following ones are supported:

1. Märklin 7687: This decoder presents itself as an accessory with 16 consecutive accessory addresses. By default the address range starts with 225, but you can change it. It does not support bridge position feedback, but is is possible to create an end state feedback.
2. TT-DEC from Littfinski DatenTechnik (LDT): This decoder is fully compatible with the Märklin 7687.
3. DSD from digital-bahn.de: This decoder is similar to the TT-DEC with some minor changes in address mapping.
4. Helljan 89121: This decoder presents itself as an accessory with 24 consecutive accessory addresses for a total of 48 positions. There is no position feedback.
5. Fleischmann 6915 Turn-Control: It normally reserves 100 addresses from 200-299. There is no position feedback.
6. Digikeijs DR5052: It normally reserves 50 addresses from 200-249. It has no direct bridge position feedback while moving, but it is possible to create an end state feedback.
7. Rautenhaus SLX 815 or SLX 819: It uses one Selectrix address that relates to 8 accessory addresses in iTrain. It has no direct bridge position feedback while moving, but knows when the end state has been reached.
8. DSM PIC from Stärz: This decoder is fully compatible with the SLX 815 protocol.
9. Müt: It uses two Selectrix addresses (control + feedback) that relate to 16 accessory addresses in iTrain, so there is direct position feedback while moving.
10. BiDiB StepControl: This BiDiB module reports intermediate bridge position feedbacks while turning.

11. DinaSys Turntable Control (DTC): This is a separate interface with integrated bridge position feedbackack and relays for sharing block outputs. You have to use address 1.
12. Draai15 from Kees Moermans: This is a simple turntable decoder that presents itself as an accessory with 4 consecutive accessory addresses. It does not support bridge position feedback.

Accessory

The turntable accessory is what is physically attached to your layout and has to be configured first. The properties in the upper half are comparable with the other accessories, but in addition the type of 'Decoder' used must be specified. In case the turntable has an address range, only the first address of the range has to be specified in the 'Address' field.



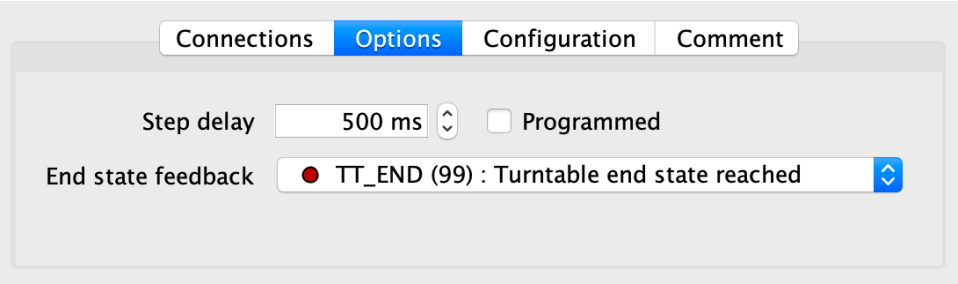
Now you have to check the number of possible connections. Most commercial turntables have 48 connections, but other values are also supported. The table on the left now lists the number of rows as specified, in our case 48. Now you have to specify which connections actually have a track connected ('Use') and in which direction a loc must leave the bridge on this connection. There are 4 options:

1. None - Choose the quickest way to turn to the connection, independent of the direction of the loc.

2. Keep - Keep the direction of the loc, so when a loc will enter at one side of the bridge, it will leave at the other side.
3. Forward - The loc always leaves the bridge in forward direction.
4. Backward - The loc always leaves the bridge in backward direction. Useful for parking a steam loc.

In the column 'Feedback' you can add feedbacks to report the actual state when the table is turning, but this is optional and is normally not used.

Other settings are available on the tab 'Options'.



The 'Step delay' is used for simulating the turning of the bridge in case there is no bridge position feedback or the interface controlling the turntable is offline. In that case the computer gives a command and the turntable will start to move, but the program is not informed where the bridge is during the movement. To know when it has reached its destination and to animate the movement on the screen, you have to know how long it takes, considering that the turning speed is constant. By manually turning the bridge 180˚ and measuring how long it takes, you can calculate the delay per step.

If the turntable decoder allows you to program it - so that only the connected tracks can be selected and not the intermediate ones -, the option 'Programmed' will be available. iTrain prefers a non-programmed decoder, but if you have programmed the decoder you have to check this option to prevent malfunctioning.

When the decoder has an output that the end state has been reached, you have to create a feedback in iTrain for this and select it in the field 'End state feedback'. The movement will still be simulated based on time, but the last position will only be shown when the feedback has been activated, to prevent the loc from driving off the bridge too early.

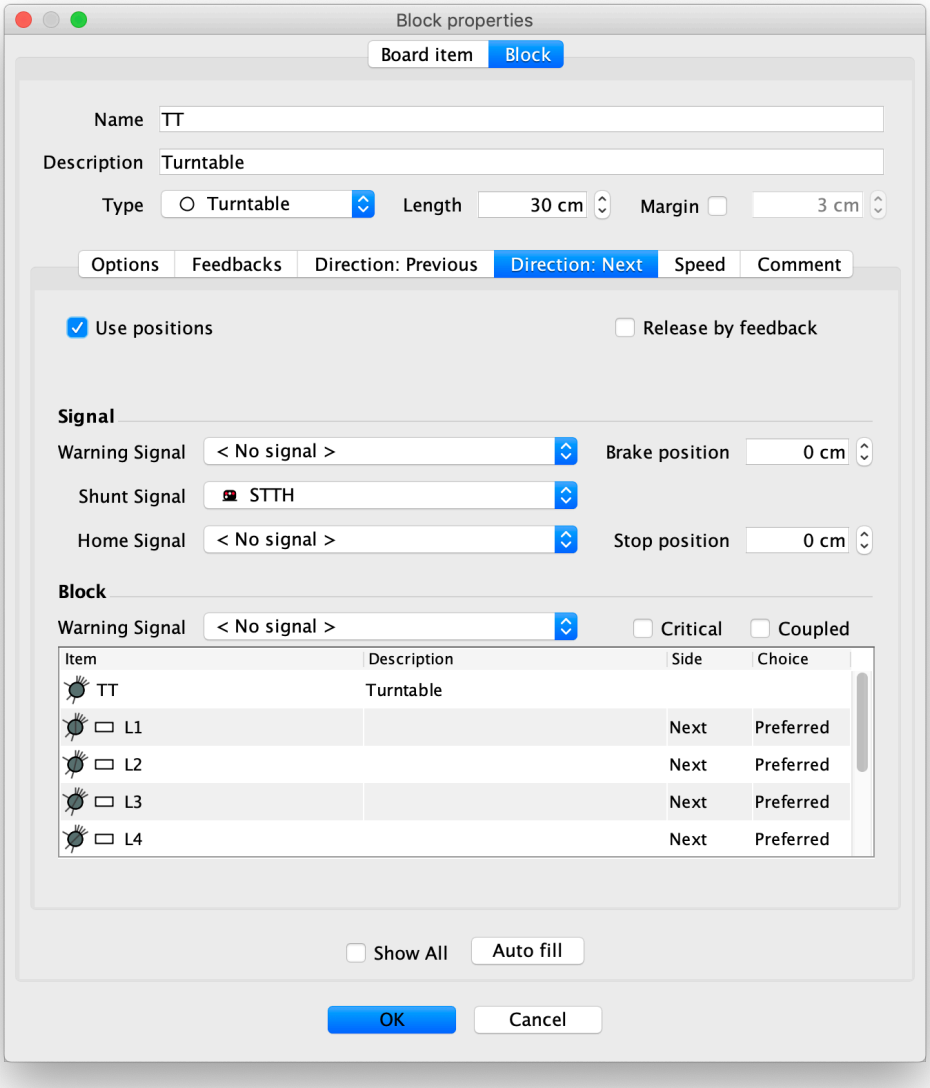
NB: This end state feedback is different from the occupancy feedback to indicate that the track on the turntable has been occupied. This occupancy feedback is entered in the block properties.

Feedback

The track of the turntable can be used as an occupancy feedback to detect the loc and it is necessary to have feedback on the bridge to follow and stop the loc. Create a feedback, possibly with the same name as the turntable accessory, by double-clicking on the feedback symbol within the turntable item.

Block

Next to being an accessory, the turntable is also a small block that only can contain a loc.



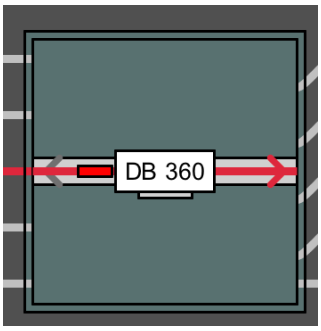
When you double-click on the block symbol within the turntable item, you can edit this block. The type is automatically set to ‘Turntable’. Because the turntable is turnable, there is normally no distinction between the two direction tabs, so they should be filled in equally. The easiest way is to use the ‘Auto fill’ button at the bottom. This guarantees the correct block connections and feedback settings. For the rest it is just an ordinary block, but probably with only one occupancy feedback. So you can benefit from using positions to automatically let the loc stop in the middle of the block when the ‘Stop position’ is 0.

Control

The bridge on the turntable has two sides. The side with the control cabin (drawn as an extra square) is called the ‘Next’ side and the other side the ‘Previous’ side, just as with a block. To control or turn the turntable to the correct position manually with the mouse, you can click outside the space of the bridge on a track, and the ‘Next’ side of the bridge will turn to that side via the shortest path. To turn the ‘Previous side’ to the clicked position use the Shift-key when clicking.

The feedback and block part of the turntable element will react like any other feedback and block element.

Transfer table



The transfer table is an element to connect multiple tracks together by means of a shifting bridge. That means that the connected tracks will always be connected to the same side of the bridge.

On the toolbar the transfer table is just a small element, but it has been designed to be a much bigger element on the switchboard, covering multiple cells in both directions. Depending on the number of outputs on each side used it is recommended to use at least 5x5 cells, for example for the Märklin 7294.

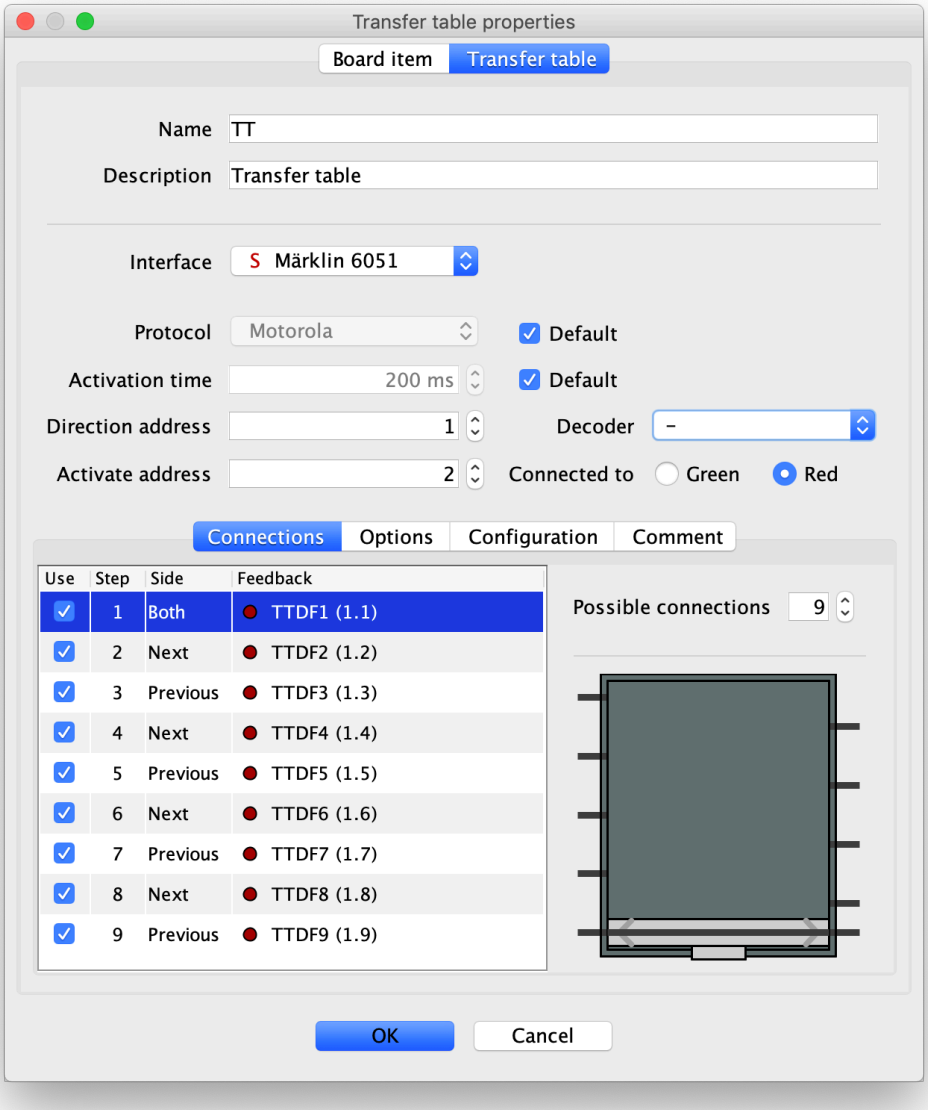
Normally the height should at least match the number of track connections on the side. Just enlarge the transfer table until it looks right by selecting the transfer table and using Shift + Alt + cursor keys to size. It is also possible to turn the whole transfer table in both directions with the ‘R’ and ‘T’ keys (counter clockwise and clockwise).

The transfer table element in the switchboard does not just represent the accessory transfer table, but also the block and the occupancy feedback. When you double-click on the white block element in the middle of the table, you edit the block properties of the associated block with its own type ‘Transfer table’. When you double-click on the feedback item, you edit the feedback like any other feedback. Clicking elsewhere on the transfer table will edit the transfer table accessory.

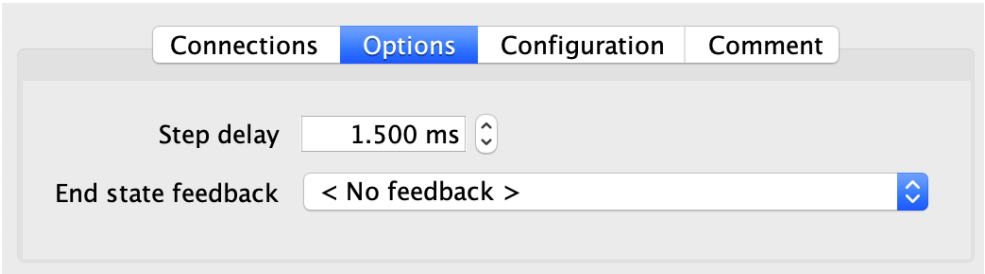
Accessory
The transfer table accessory is what is physically attached to your layout and has to be configured first. The properties in the upper half are comparable with the other accessories. First it is important to select the ‘Decoder’:

- No decoder - No specific decoder, but a general decoder with continuous outputs is used, for example a k84. Two addresses are needed: One to switch between the direction of the the bridge and another to activate the motor so that the bridge will start moving. If the last is switched off it will automatically run until the next connection. In this case you need to fill in the feedbacks[60] in the column ‘Feedbacks’ on the tab ‘Connections’, so that iTrain can switch off the motor in case the destination has been reached.
- Default decoder - An address based decoder is used that takes care that the bridge stops at the selected end position. The first address has two outputs (`red` and `green`) for the first two bridge positions and the next addresses are for the other bridge positions.
- Rautenhaus SLX 819 - A decoder for the Selectrix bus in combination with a step motor using the same protocol as the turntable decoder.

[60] In case of the Märklin 7294 the feedbacks can be attached to the catenary outputs of the transfer table.



Now you have to select the number of connections or positions the bridge can have. In case of the Märklin 7294 that is 5 on each side, but one is shared on both sides to that counts to a total of 9. For every position you have to specify on which side it is connected. Depending on the selected decoder, you have to select the feedback that detects that the position has been reached. In the position with the control cabin (drawn as an extra square) on the bridge at the bottom, the left side is the 'Previous' side and the right side is the 'Next' side.



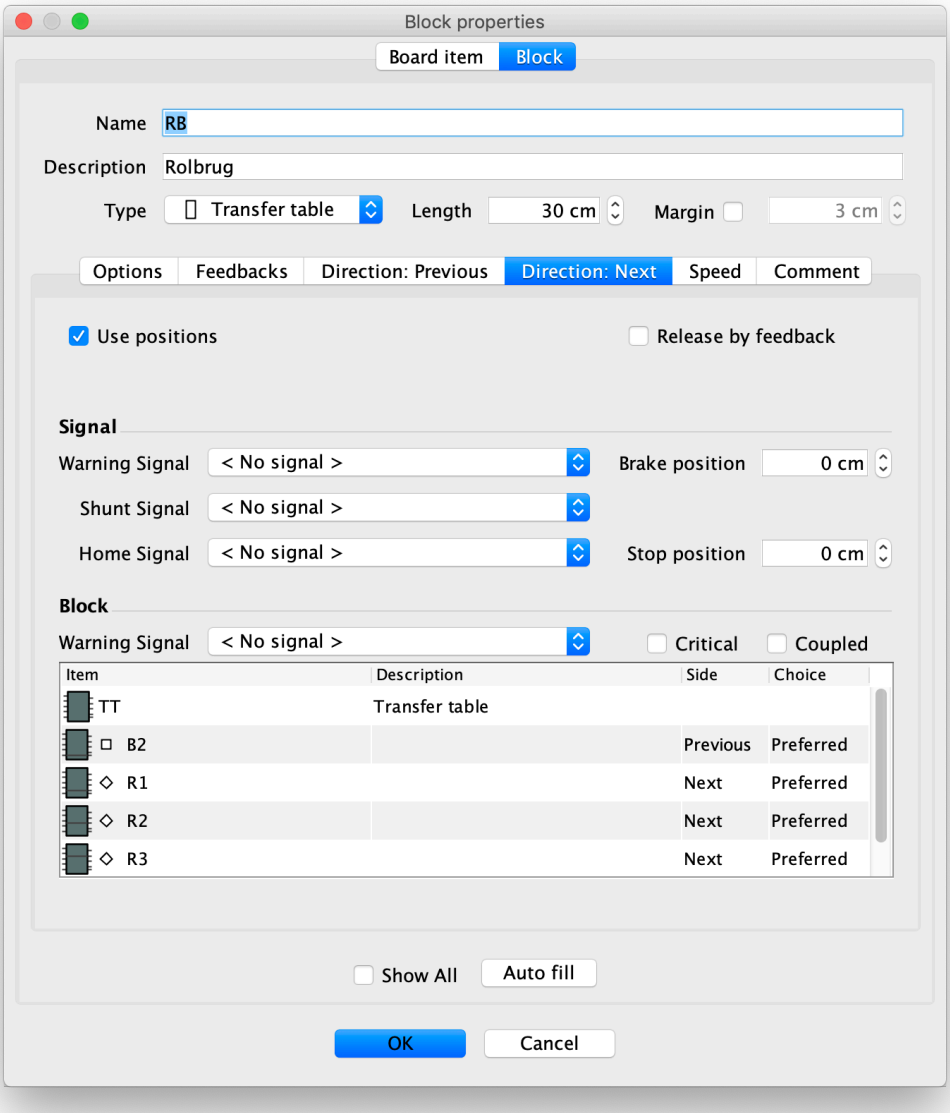
In case of the 'Default' decoder, you can fill in an 'End state feedback' on the tab 'Options' so that it is known when the end position has been reached. This can simply be that the motor stopped running.

Feedback

The track of the transfer table can be used as an occupancy feedback to detect the loc and it is necessary to have feedback on the bridge to follow and stop the loc. Create a feedback, possibly with the same name as the transfer table accessory, by double-clicking on the feedback symbol within the transfer table item.

Block

Next to being an accessory, the transfer table is also a small block that can only contain a

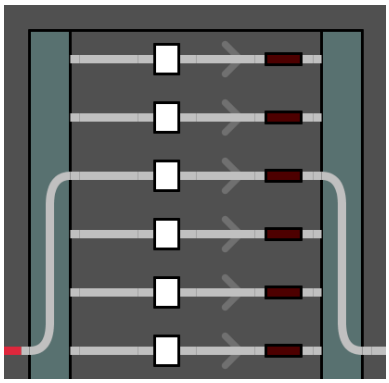


loc. When you double-click on the block symbol within the transfer table item, you can edit this block. The type is automatically set to ‘Transfer table’. For the rest, the easiest way is to use the ‘Auto fill’ button at the bottom. This guarantees the correct block connections and feedback settings. For the rest it is just an ordinary block, but probably with only one occupancy feedback. So you can benefit from using positions to automatically let the loc stop in the middle of the block when the ‘Stop position’ is 0.

Control

The bridge on the transfer table has two sides and you can click on any of the connections on either side and the bridge will move to to that connection. The feedback and block part of the transfer table element will react like any other feedback and block element.

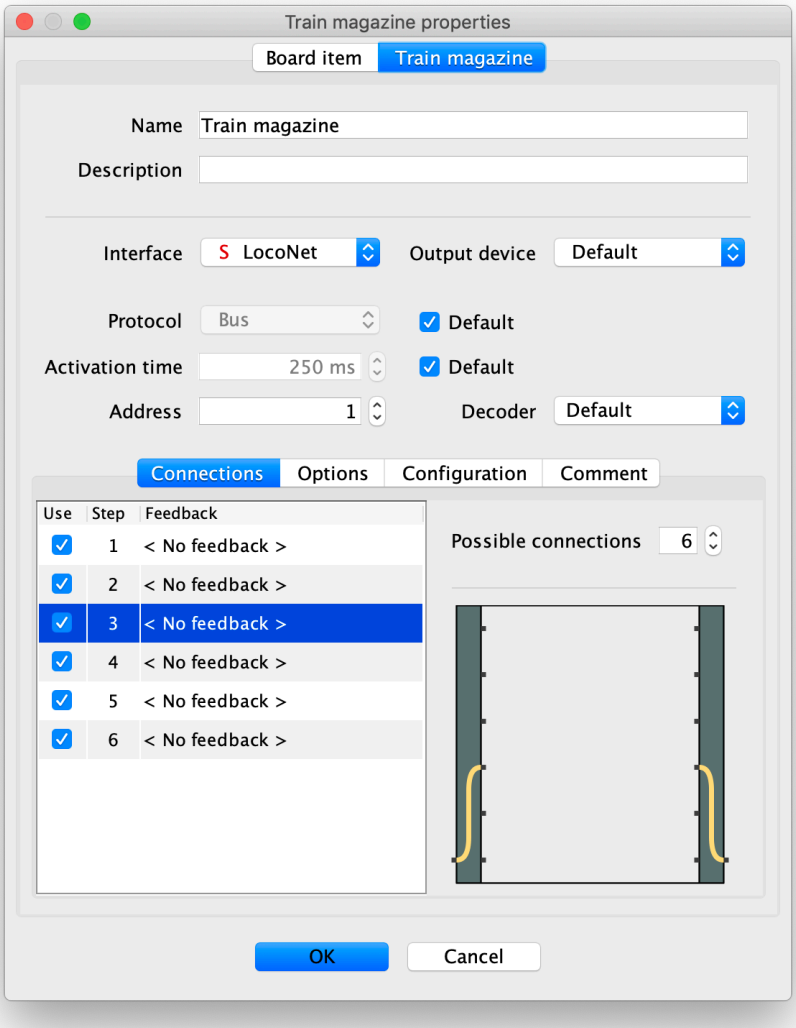
Train magazine



A train magazine is an element to connect multiple moving tracks on a shifting table or a lift to a fixed track on one or both sides. It is more or less the opposite of the transfer table in which one track is moving. The difference with the transfer table is that a train magazine is only an accessory, because you draw and define the moving tracks with blocks and feedbacks inside the transparent part of the train magazine.

On the toolbar the train magazine is just a small element, but it has been designed to be a much bigger element on the switchboard, covering multiple cells in both directions. There needs to be enough space to draw blocks inside of it, so a minimum width of 6 cells is recommended (4 for the block itself).

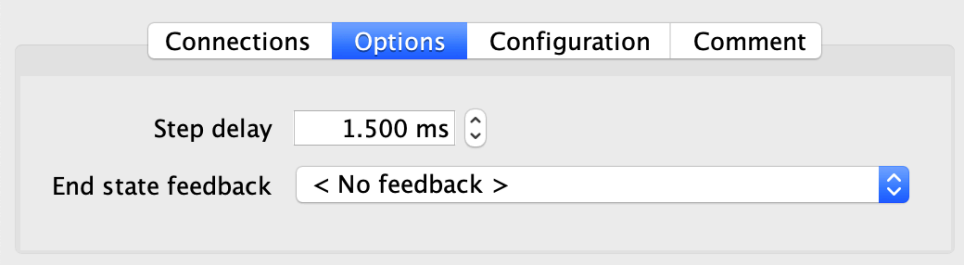
Accessory
The properties in the upper half are comparable with the other accessories.



Three decoder types are currently supported:

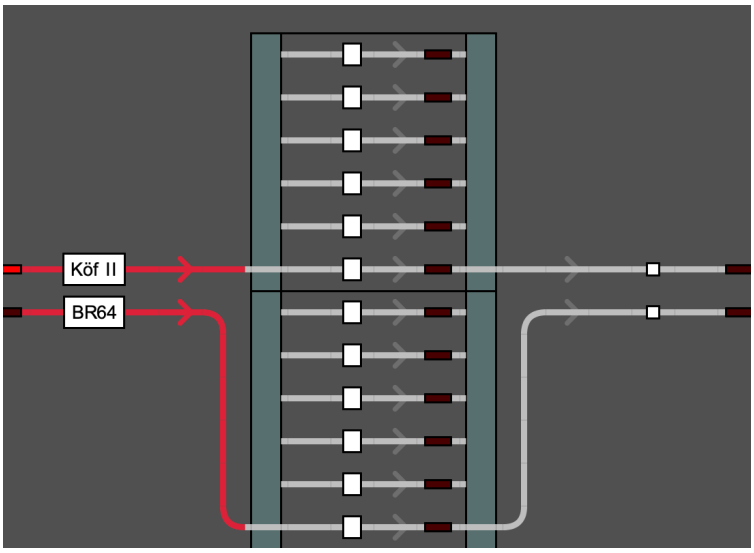
- Default decoder - A general address based decoder is used that takes care that the device stops at the selected end position. The first address has two outputs (`red` and `green`) for the first two positions and the next addresses are for the other positions.
- Rautenhaus SLX 819 - A decoder for the Selectrix bus in combination with a step motor using the same protocol as the turntable decoder.
- Müt Loklift - A special decoder for a Loklift for the Selectrix bus with actual position feedback.

On the tab ‘Connections’ you define the number of tracks in ‘Possible connections’. In the table you can specify an optional feedback per position to inform the software where it is.



In case of the ‘Default’ decoder, you can fill in an ‘End state feedback’ on the tab ‘Options’ so that it is known when the end position has been reached. This can simply be that the motor stopped running.

To control the train magazine, just click inside of it on the destination position and it will start to move.



Tip: The train magazine can be used for a loc or train lift. For a lift it is common to have multiple tracks next to each other. This can be achieved by drawing multiple train magazine elements below each other, but assigning them the same underlying train magazine accessory via the tab ‘Board item’.

Model clock

There is general clock available in iTrain that can be configured in the general settings for time. You can place this so called model clock on the switchboard as well. It is recommended using at least 3x3 cells to display it well. The seconds are only displayed if they are not moving too fast, so if the factor is not more than 20.

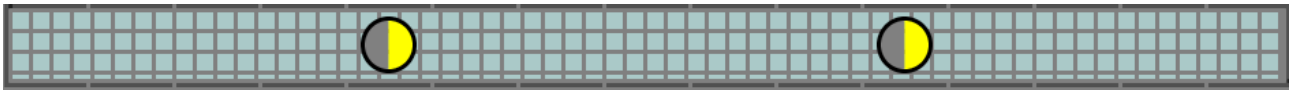
Double clicking on the clock sets it to the current time. Pressing 'Shift' + double click, sets the time to the start time of the day.

Deleting switchboard elements

When deleting an element from the switchboard that refers to a control object (for example a turnout, feedback, etc.) only the visual element is deleted and not the control object details. So if you accidentally delete a turnout element and then add a turnout element to the switchboard to replace it, you can still select the turnout object from the drop-down box. You should not define it again.
To completely delete the control object, you have to remove it from the 'Browser', because then all references to this control object (for example in blocks and track routes) will be removed. However it is always preferable to edit and reuse a control object when something needs to be changed rather than to delete it and create a new one.

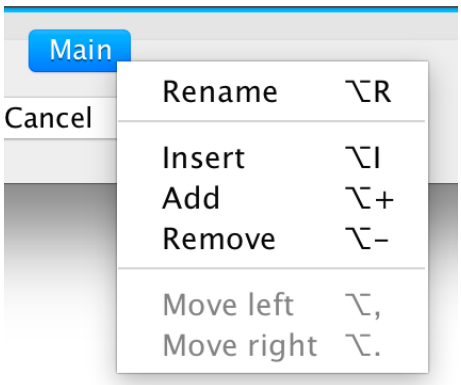
Reusing control objects

Because of the separation between the control objects and the switchboard elements, it is possible to have multiple switchboard elements with the same control object, for example, to assign a signal object to both the main and its warning signal or to assign one light object (relay object) to multiple elements on a platform.



A quick way to assign a control object already assigned to a switchboard element to another switchboard element is to start dragging an element, while at the same time holding the 'Control' or Alt key, and drop it on another element.

Adding/Modifying switchboard tabs



It is possible to add extra tabs or rename existing tabs by using the popup menu on the switchboard tab (right mouse click) in edit mode. The function always works on the active tab. The 'Insert' inserts at the selected tab, so before the current switchboard, and 'Add' appends a new tab at the end. If you want to change the order of the tabs later on, you can use 'Move left' or 'Move right' to move the current tab.

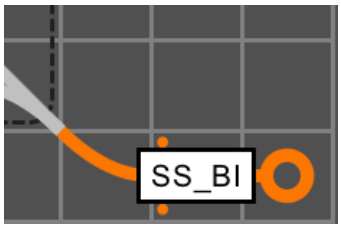
When using multiple tabs, the same accessory, feedback, track route and block elements might appear on different tabs. In that case it is necessary to define the objects only once and to attach them to the

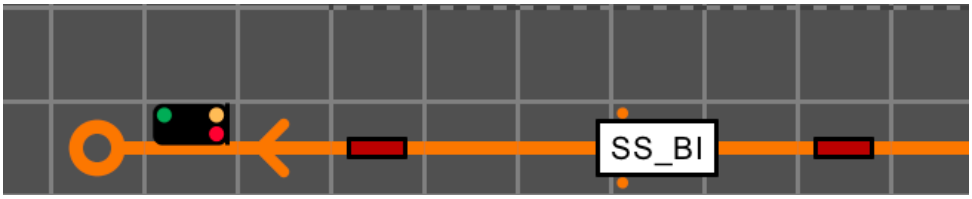
switchboard elements on different tabs.

Tip: Often parts of the layout of two tabs have some overlap. It is possible to select an area and use 'copy' on one tab and 'paste' on another tab to copy parts to another tab. In that case the visual elements are copied, but references to the same accessory and feedback objects are used.

Linking switchboard elements

When drawing your layout schematically in iTrain, sometimes a track has to continue on another part of the switchboard or even on another tab without a directly connected line. In that case it is recommended to use two 'Link' elements (on both sides of the split) to indicate that the track will continue somewhere else. Both link elements have to be assigned to the same block, so the split always needs to be inside a block.



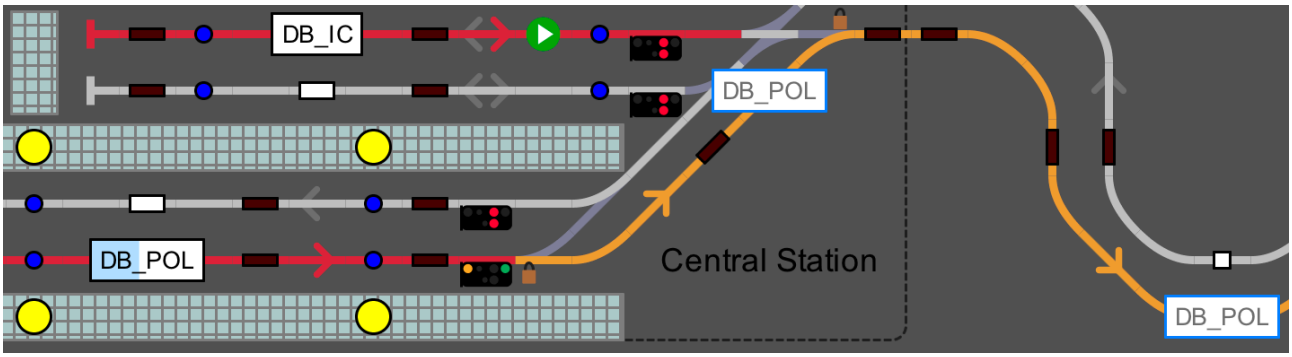


In case the block continues on another tab it is strongly recommended to copy at least the block element to the other side (and check the direction dots or also add the arrow). Put all other elements in the block (signals, feedbacks, arrow) at least on one side of the split so that 'Auto fill' will work properly on one tab except for the block connection on the side with the link element. This block connection can be filled automatically with the button 'Connect' on top of the other tab.

Tip: When using multiple tabs to control your layout, you can double click the 'Link' element to jump directly to the tab with the other side of the block.

Reservations

On a digital layout (except for block controlled systems), you can run your trains anywhere via manual control. The computer adds block control and makes sure that when you enter a block, the block will be protected by signals so that no other train is allowed to enter the same block. But what if two trains want to enter a block at the same time from a different direction? Here is where the reservations come in. A reservation reserves a block for a train, and for other trains the block will be treated as if the train were already in the block, and they will not have access to it. In the switchboard a reservation is indicated by a yellow/orange line, with the train name in the block element in gray.



The same principle holds for reservations of turnouts that connect the blocks. When a train is in one block and reserves the next block, then it is also necessary to reserve the turnouts that connect those two blocks (if there are any). You can see that a turnout is reserved by a 'lock' symbol and the reserved color (kind of yellow/orange). This prevents others from changing the turnout state until the turnout is released. A block element with a blue border indicates that this block is responsible for releasing the turnouts.

Reservations for blocks and the necessary turnouts can be made manually or automatically. When driving predefined routes (discussed in the next chapter) the computer will automatically make the necessary reservations, but it is also possible to make manual reservations. There are three ways of making reservations manually:

- Based on source: When you hold the Command key and click on a block element in the switchboard with a train, then it will try to reserve the next block in front of the train while following the possible turnouts in between with their current state.
- Based on destination: When you hold the Command key and click on a block element in the switchboard without a train, then it will try to reserve this block for a train that is approaching this block. If the block is approached from both sides, then the train on the 'Previous' side will be chosen, because that is the preferred driving direction. Approaching in this case means that a neighboring block is occupied or reserved by a train in the direction of this block.
- Based on source and destination: When you drag a train from a block element and drop it on another block element and select 'Reserve' in the popup menu, then it will reserve the block on which the train was dropped and all the blocks in between via a shortest path algorithm.

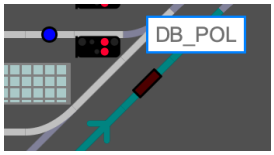
The last method is the safest one, because you specify the source and destination exactly. The other ones might be easier and faster and are intended for reservations with no ambiguity, such as when there are no turnouts between the blocks. See Appendix A for some key mappings

related to reservations.

Note: Make sure that the train is going in the correct direction before you make a reservation or else the reservation will fail. You can easily check this by looking at the arrow element in the block of the train. It should point in the direction of the block to reserve.

Reservations can be undone by holding the Shift and Command key and clicking on a train in a block element. This will remove the reservation on the front (so not necessarily the one on which was clicked). You can repeat this until all reservations have been removed.

Note: If the train is in manual mode, then shunt reservations will be made with a different teal color, choosing the appropriate shunt signal states.



Release blocks

A train is normally in one block and we call this the ‘Control’ block. This block is colored red and the name of the train is displayed in black letter in the block element. Sometimes the tail of the train is still in another block or even in multiple other blocks. These are called the ‘Release’ blocks, as they will soon be released when the train has moved to the control block. These blocks are also colored red, but the name of the train is in gray. When the direction of the train changes, the last release block will become the new ‘Control’ block and the previous ‘Control’ block will become a ‘Release’ block.

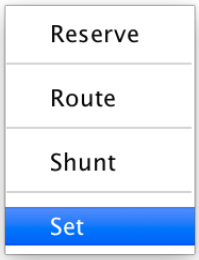
When you want to extend an existing train, you might end up covering an extra block for the train. To manually extend this in the switchboard you can hold the Command and Alt key and click on a block element to add a ‘Release’ block to a train. This way you prevent other trains to use this block. By holding the Shift, Command and Alt key and clicking on a block element, a ‘Release’ block will be removed.

Removing trains from blocks

We have seen a way to extend the blocks attached to a train by adding ‘Reserved’ and ‘Release’ blocks and how to remove them again. There is one more method to remove a train from a block, by holding the Shift key and clicking on a block element. This will not necessarily remove the train from the clicked block, but it will first remove the reservations in order from the front back to the ‘Control’ block and then remove all the ‘Release’ blocks from the tail back to the ‘Control’ block, and finally it will remove the train from the ‘Control’ block. So by repeatedly clicking on the ‘Control’ block, while holding the Shift key, you will remove the train from the switchboard.

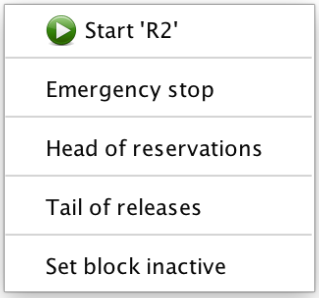
Moving trains over or into the switchboard

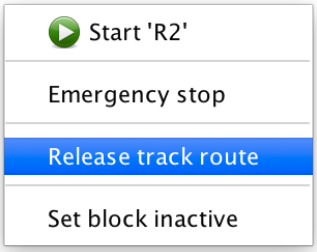
When putting trains onto the layout or manually moving a train from one position on the layout to another position on the layout, you also have to change their position on the switchboard. To put a train from somewhere onto a block, you can drag a train (from the 'Train control', the 'Train overview' or from a block element in the switchboard) and drop it on a block element. Normally a popup will appear and you have to select 'Set'. In case the train is not already on the switchboard or you are holding the Alt or Control key, the popup menu will not appear.



Context popup menu

Some of the release actions for reserved and release blocks can also be reached via a context sensitive popup menu of the block element, without using the modifier keys. This popup menu can be shown by pressing the (left) mouse button on the block and waiting until it appears.





'Head of reservations' means that all reserved blocks beyond this block will be released and this block becomes the head. The same can be achieved without a popup menu by holding the Shift and Command key and double clicking on a block.

'Tail of releases' means that all release blocks behind this block will be released and this block becomes the tail. The same can be achieved without a popup menu by holding the Shift, Alt and Command key and double clicking on a block.

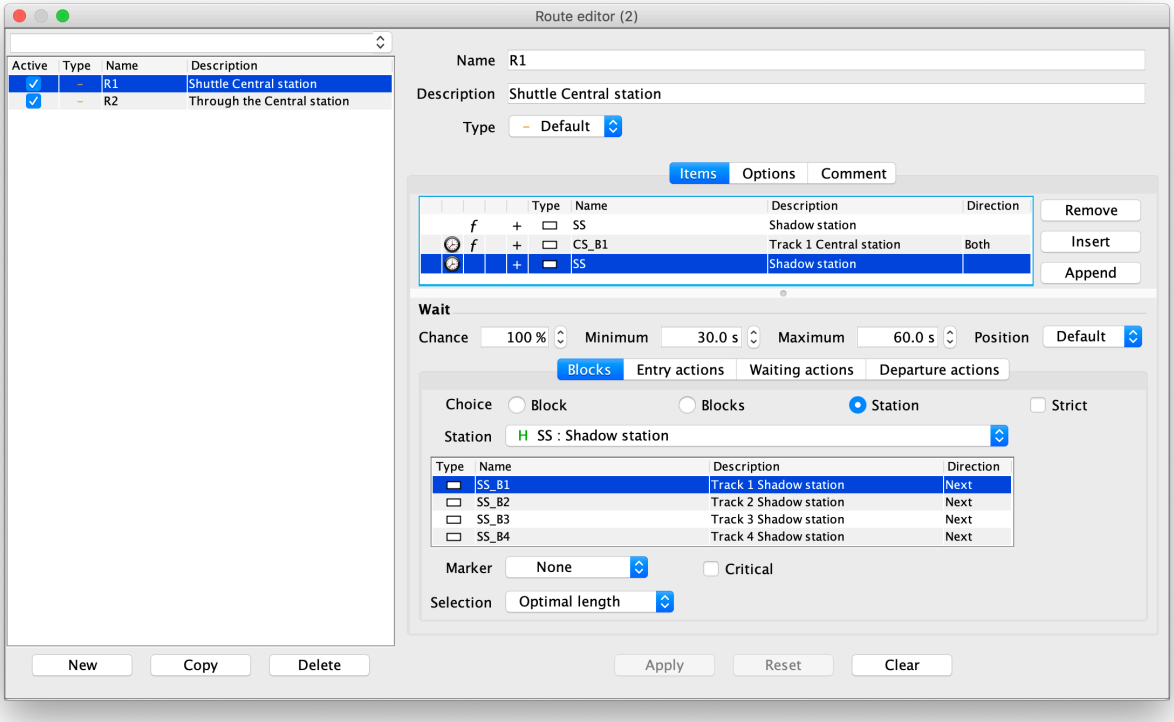
In case there are no release blocks, but there are still reserved turnouts or other accessories that are part of the track route, the option 'Release track route' will be available to release the track route. Only use this if you are sure that no part of the train is occupying (parts of) the reserved turnout. The same can be achieved without a popup menu by holding Alt key and clicking on the block.

The always available last item 'Set block inactive/active' makes the block inactive (dark gray line) or active again depending on the current state. The same can be achieved without a popup menu by holding the Alt key and clicking on the block, but only in case there are no track routes to release.

Tip: The popup menu items are spaced so that you can easily choose the correct item with your finger on a touchscreen.

Train routes[61]

Blocks provide some basic protection for collisions by not allowing two trains to be in the same block. Train routes allow you to automate the path a train is following on your layout. Train routes can be defined on their own and be attached to a train. Multiple trains can share the same route. To go to the route editor, use menu ‘Edit’ -> ‘Train routes’ or press Shift + Command + F4.



Routes are basically a list of blocks and/or stations that should be visited by the train. It is not necessary to specify all the blocks in the route, because iTrain can calculate the shortest path to the next block or station, but at least all the scheduled stops should be specified. By specifying more blocks between stops there will be less ambiguity about the path to follow, but alternative paths will be restricted, because all blocks specified have to be visited.

In most cases a route is made so that it can be repeated. In that case make sure that the last item in the route is also available earlier in the list (preferably at the start) so the route can be picked up again from an earlier point. So if a route is going from block or station A to B to C and back the order of blocks in the route must be A, B, C, B, A. For a circle with four blocks or stations it will be A, B, C, D, A. In these cases it is recommended starting and ending a route in a station block where it is logical to start and finish a route.

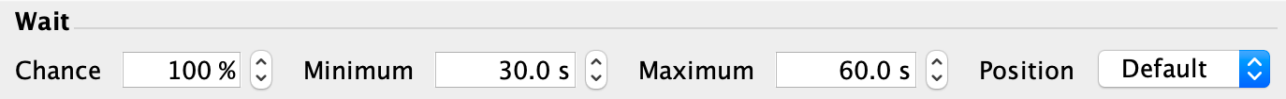
On the tab ‘Items’ you can press the ‘Add’ (append to the end) or ‘Insert’ (at the selected location) button to add an entry to the list in the table. Everything below the table is a specification for the item selected in the upper table. Every entry in the table shows the name and description of a block or station of the item in the list. Icons on the left of the name indicate that special features are attached to this route block item.

---

[61] This option is not available in the Lite edition of iTrain

Waiting time

It is possible to wait in every block - in which it is possible to stop - along the route. An actual wait of the train will prevent further reservations, until the waiting time is over.



At first you have to specify the chance that the train will wait. Zero means no waiting time and 100% means wait always. A value in between will let the system decide during execution of the route whether the train will wait. Only use this for intermediate stops in the route so iTrain can always find a place to wait in the route somewhere.

In case the train waits it will choose a waiting time that is equal or higher than the minimum and equal or lower than the maximum. So for a fixed waiting time specify the same time twice.

In the field 'Position' you can choose where you want the train to stop in the block:[62]

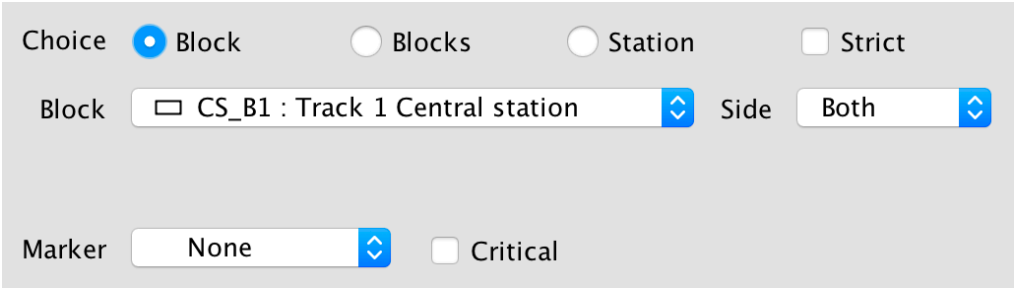
1. Default - this is, depending on an available platform and the train type, either along the platform or at the end of the block at the stop position
2. Start - always at the start of the block (at the stop position in the other direction)
3. End - always at the end of the block at the stop position

If a waiting chance and waiting time greater than zero are specified, a clock icon will appear in the second column of the table. 

Note: There is no use in adding a waiting time to the first item in the route, because that is only where the route will start. It is however recommended adding a 100% waiting time to the last item of a route, because that will be the waiting time before the route will be repeated or finished.

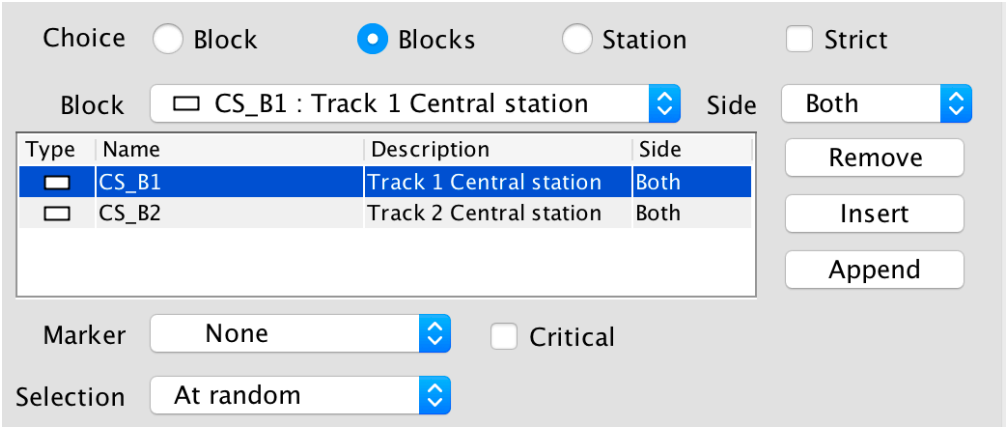
Blocks

Every item in the route refers to a single block, multiple blocks or a station:

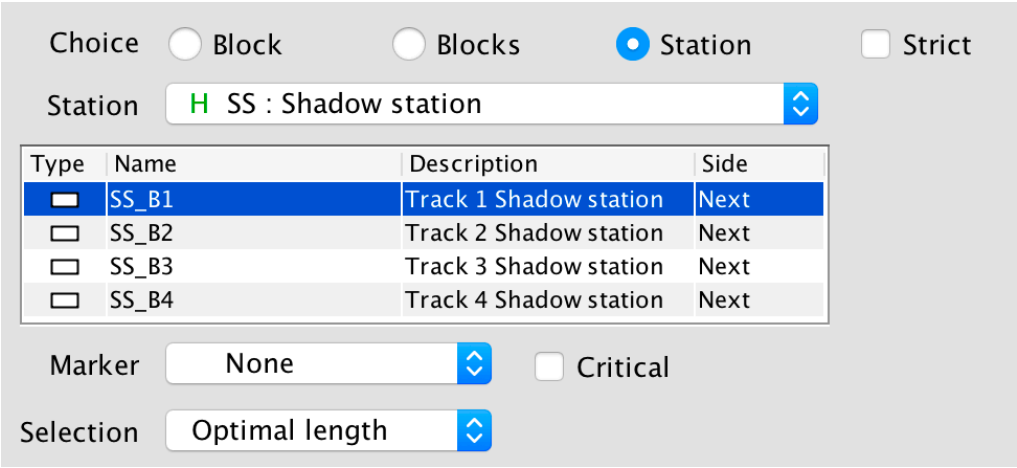


- Block - You have to select the block and the direction side in which the block should be driven when the train is entering the block.

[62] Stops at alternative positions in the block are only possible if 'Use positions' is used in the block.



- Blocks - You can make a list of blocks. Normally these blocks are parallel blocks, but it is not limited to that. It is allowed to take two blocks in different locations to generate some random behavior. You can also specify the direction side per block in which it should be driven when the train is entering the block.



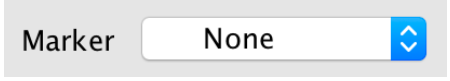
- Station - You have to select a station that is defined as a list of blocks. Automatically the list of blocks currently attached to this station will be shown, but the list is not editable here and not stored with the route itself. When the station definition changes the route will use the new list of blocks.

The option ‘Strict’ on the right side specifies that the selected blocks must be visited directly after the previous specified block without allowing any intermediate block in between. This can be useful if the route finder is too smart and finds an alternative path that should be excluded. This is indicated by a ‘→’ icon in the fourth column of the item table.

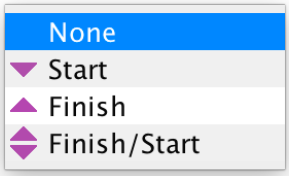
The last two choices (Blocks and Station) make the route more dynamic, because a block from a list will be chosen. This is indicated by a ‘+’ icon in the fifth column of the item table.

Marker

Markers are meant to mark the part(s) of the route that will be repeated in case a route has a repeat count and the part that needs to be repeated is not the whole route. For simple routes, markers are not necessary.



To add ‘Start’ and ‘Finish’ markers in the route, select an item and select the appropriate marker. The marker icons will appear in the first column of the table. The repeat time can be entered on the tab ‘Options’, described later.



Tip: Markers can be used to first position a train from a siding into the station, then repeat a schedule multiple times, and then put it back on a siding again.

Critical block

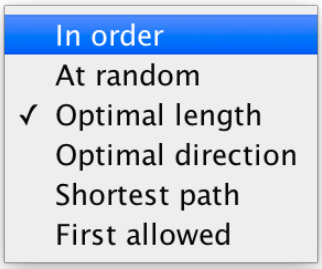
Every item in the route without a waiting time can be marked ‘Critical’. It means that if a block specified in this item is reserved the following block on the layout should be reserved as well. As long as that is not possible the block will not be reserved. Critical items will get an ‘!’ icon in the second or fourth column of the table.

Note: This is the same functionality that can be found on the direction side tabs of the block definitions, but now it can be specified per route. Use this option judiciously to prevent deadlocks, because nothing is allowed to leave anymore.

Tip: Routes for cargo trains can have the station blocks marked as ‘Critical’ so that a stop in a station is unlikely, because the next block has already been reserved or the train is waiting before the station.

Selection

In case multiple blocks can be chosen you have to specify the criteria to choose. This can be specified in the field ‘Selection’. The options are:

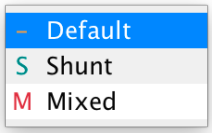


- ‘In order’ means the first block in the list will be chosen, unless this one is already occupied, then the next will be tried and so on.
- ‘At random’ means that the list is shuffled at random and then the first block will be chosen, unless this one is already occupied, then the next will be tried and so on.
- ‘Optimal length’ means all the blocks in the list are sorted in the order of their length from short to long, and blocks that are too short for this train are removed. Now the first block will be chosen, unless this one is already occupied, then the next will be tried and so on. This option is to prevent short trains from occupying a long block in a station so that a next longer train cannot enter it anymore, because only short blocks are left over. It is useful for shadow stations.
- ‘Optimal direction’ means all the blocks in the list will be sorted on the direction of the block in relation to the direction in which it will be entered. So blocks that are entered in a non-preferred direction will be chosen last.
- Shortest path means all the blocks in the list are sorted in the order of their path length to reach them. Now the first block will be chosen, unless this one is already occupied, then the next will be tried and so on. This is only useful if not all intermediate blocks in the route are specified.
- ‘First allowed’ always takes the first block in the list (even if it is occupied) unless the block or something in the path to this block is disabled or not allowed (specified via the permissions tab of the loc or train), then it will take the next and so on. It is useful for

fixed routes that are only changed when some blocks are inactive or another loc or train with different permissions is used.

Note: An electrical loc is never allowed to use a block without catenary.

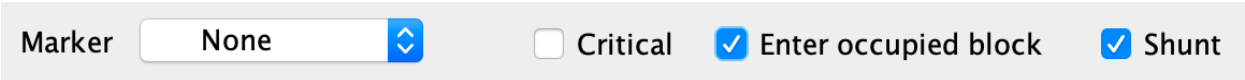
Shunt



As explained already, creating a new route in iTrain, defaults to a normal route, but there is also the option to shunt the whole or part of the route. When shunting, the following things are different:

- There is a separate maximum speed for shunting defined globally and/or per block.
- Reservations are made in different teal color instead of the yellow/orange color.
- Direction changes in single direction blocks are allowed.

You can either choose to make the whole route of type ‘Shunt’, or create a ‘Mixed’ route in which case the shunting parts are explicitly marked via the extra option ‘Shunt’ per item in the route.



In both cases a new option ‘Enter occupied block’ will be shown to indicate whether the movement should only enter a free block or whether it is allowed to enter an occupied block.[63]

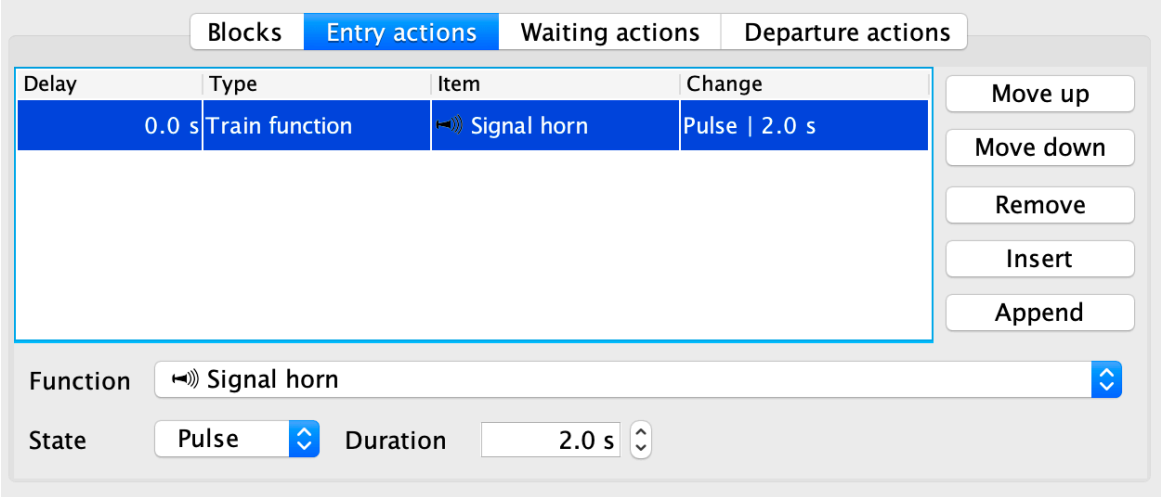
Note: The destination block must also allow the shunt in an occupied block on the tab ‘Options’ of the block properties. But it will only work properly if the block is technically able to detect an additional train in an occupied block. This can be achieved with a short entry feedback that remains free.[64]

Actions

It is possible to control train functions or set accessories in a route when entering a block, waiting in a block, or departing a block after a stop. Three additional tabs allow you to enter them.

[63] To enter an occupied block, the positions of the trains in this block must be known and the trains should not have an active route (or driving automatically without routes). In addition, the block has to use ‘Positions’ to be able to stop independently of stop feedbacks.

[64] It is also possible to use RailCom detection to detect an additional incoming train, but this will be less precise as RailCom processing has a variable delay.

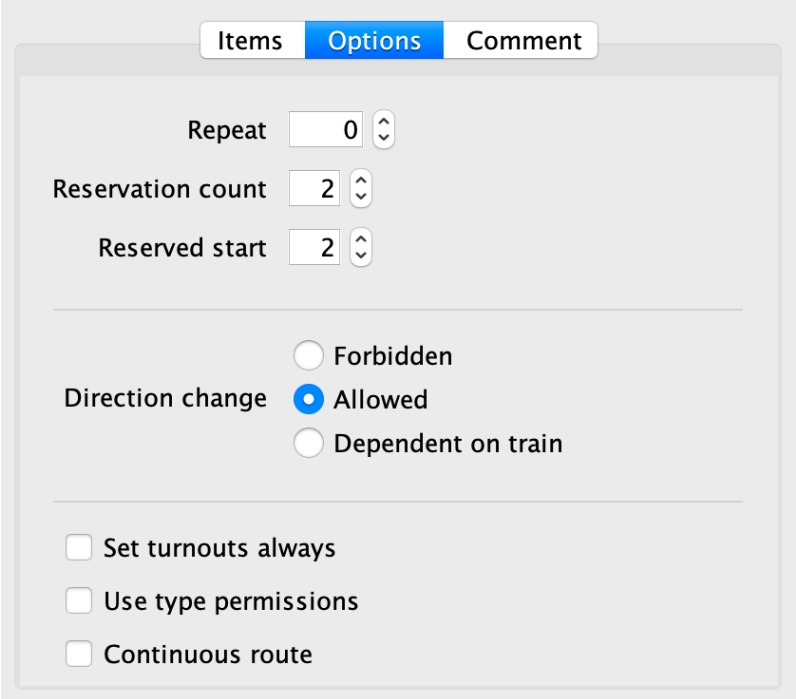


To add actions on a tab, use the ‘Insert’ or ‘Append’ button. You can edit the time that should be waited before executing the action in the first column called ‘Delay’. After selecting the type of action in the second column, an action specific editor will appear below the table to edit the specific properties. The editors are described in the next chapter ‘Actions’ in the subsection ‘Execution’. However, only a limited set of action types is supported in train routes, focussed on train functions and some accessory types.

Note: The train starts only after finishing the ‘Departure actions’ . That is the reason you can specify a duration for each train function in case the function will take some time to execute.

If actions are attached to a block in a route, a function icon will appear in the third column of the table.

Options



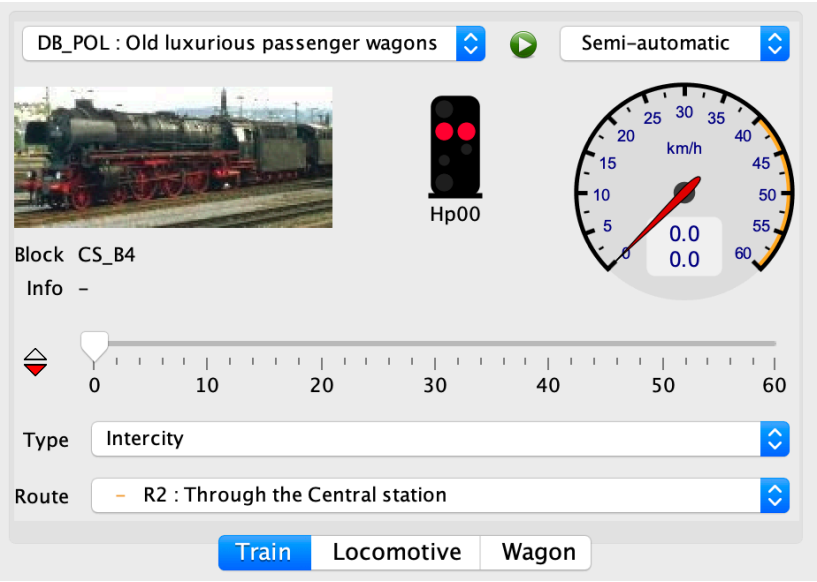
The route also has some settings that act on the whole route:

- The ‘Repeat’ count is the number of times that a route is being completely followed by a train before the route is stopped. A zero means the route is being repeated forever until the route is manually ended. If you have defined markers, then only the section between the start and end marker will be repeated.
- The ‘Reservation count’ is the number of blocks a route tries to reserve in front of the current block of the train. This is not a guarantee, but a best effort. It will however never try to reserve past a block with waiting time specified in the route until the train is stopped and the waiting time has passed. In some cases it may reserve more blocks if the train cannot stop in the last reserved block or if it is critical.
- The ‘Reserved start’ is the number of blocks that must have been reserved before the train actually starts to depart after the ‘Waiting time’ has passed or when the route has been activated. This is to prevent multiple waits for a signal of a train in a free block while it could have been waiting in the station. It does not have effect on the reservations or the signals either, but it only affects the departure time of the waiting train.
- The option ‘Direction change’ specifies if the direction of the train may be changed while executing the route. The third option ‘Dependent on train’ will allow the train to drive only in the direction with a cabin at the front. This last option is the default setting. In case of a mixed route, the shunting part is always allowed to drive in both directions.
- If the option ‘Set turnouts always’ is selected it is guaranteed that all the turnouts will be activated by the interface, even when the program thinks they are already in the correct state. This costs more switching time, but prevents errors by manual changes outside the program.
- If the option ‘Use type permissions’ is not selected, then the route will ignore the permissions defined at the train type for this route. This is useful if the type permissions limit a train in driving automatically without routes, but in specific routes you want to be able to access these blocks anyway.
- The option ‘Continuous route’ is meant to indicate that the part of the route that will be repeated has no waiting time at the end. Checking this option prevents a warning in the diagnosis.

Tip: With a mixed route and markers it is now possible to first shunt from a siding to a platform track, then drive multiple rounds without direction changes, and at the end shunt back to a siding in one route.

Route control

Routes can be attached to a train in the ‘Train control’. This associates a train with a specific route without activating the route at that moment.



Routes can be controlled in the 'Train control' and the 'Train overview' via the buttons 'Start' and 'Stop', their popup menu or a key combination.

In the switchboard, a route can be controlled via the context sensitive popup menu of a block element that contains a train. You have to press the (left) mouse button on the block element and hold it until the popup menu appears.

There are three commands:

- 'Start' - to activate the route. A route can only be started if the train is in one of the blocks along the route and will start at that position in the route. In case of multiple occurrences of the current block of a train in the route it will start at the first occurrence of the block in which the train will drive in the same direction as it is now. The key combination is Shift + F5.
- 'Stop' - to stop and deactivate a route immediately. The train will stop (not halt abruptly) at its current location. The key combination is Shift + F6.
- 'Finish' - to first visit all the remaining blocks of the route and stop after the train has arrived in the last block without any repeats. The key combination is Shift + F7. You can also select this by holding 'Shift' and clicking the Stop button. Now the button will show meaning it is still running, but will not be repeated anymore.

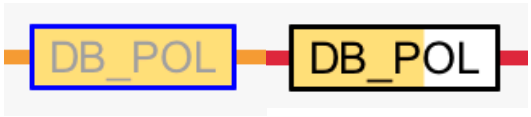
Tip: *If you want to activate the route, but not set the train in 'Automatic' mode, you have to use 'Shift' (for semi automatic control) or 'Alt' (for fully manual control) when clicking 'Start'. In this case you can control the speed manually via iTrain or a remote control. Turnouts and signals will be set automatically.*

Info

Information about the route will be displayed on the ‘Train Control’ below the image after ‘Info’ and can be added as an extra column ‘Info’ in the ‘Train overview’.

It will show a message why a route failed, what it is doing or an arrow followed by the next possible destinations to stop. The list of destinations will be reduced over time until one has been chosen.

Scheduled stop



Blocks reserved by a train in which it will have a scheduled stop (train will wait) will show a yellow bar behind the train name in gray. After stopping in that block the yellow bar will shrink with time until the waiting time is over. Now the route can be continued and reservations will be made again.

In the route info box, the same yellow bar will appear, but instead of the train name the time left will be shown.

Instant route

Creating a route takes some effort, but you will have full control of all the options and you can use the route multiple times. Sometimes you only want a simple route from A to B without options, and you wan to execute it only once. This can be achieved by dragging a loc or train to its destination.

After dropping the loc or train on the destination a popup appears. Four options will appear in which the second and third choice will create an instant route, but all are described here for completeness:

- Reserve - a full path will be created to the destination without direction changes. You have to drive the train manually or change the control type to ‘Automatic’.
- Route - a simple, temporary route to the destination will be created, only allowing direction changes dependent on the train.

- Shunt - a shunting route to the destination will be created, allowing direction changes and entry of an occupied block (depending on the destination). If the shunting light function is available in the train, it will be switched on during the movement.
- Set - just inform iTrain that the train is now set in a block by putting it on the track yourself.

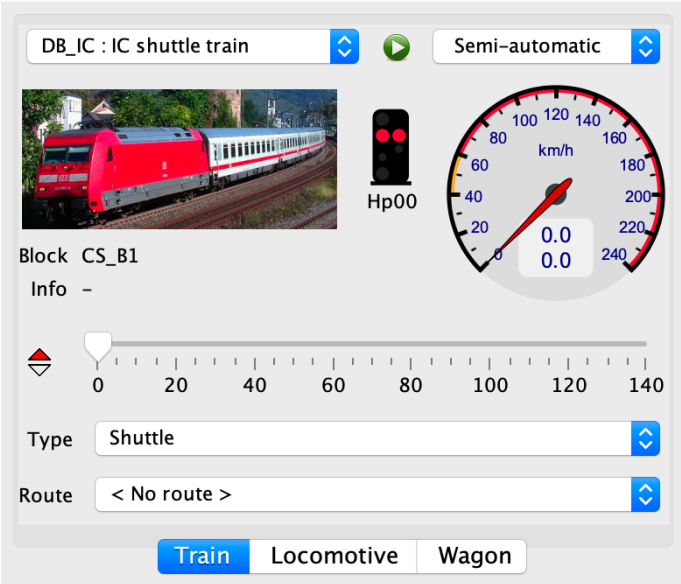
To use ‘Instant route’, select ‘Route’ or ‘Shunt’ and the route will be executed immediately. It will give ‘Info’ just like a normal route, but when the route has finished or the ‘Stop’ button is used, the temporary route is removed and you cannot continue. You have to create a new one again. The route originally attached to the train will remain there, and pressing ‘Start’ will start this route again.

Tip: You can use ‘Instant route’ to put a train in a position prior to starting the route attached to the train.

Automatic routing

To drive automatically it is not necessary to create a route and attach it to the train. You can also let iTrain find a route automatically based on its train type and the settings in the station and drive more or less randomly.

What actually happens is that iTrain will search all allowed paths from the current position of the train until there is a station it can find where it will wait. It may find multiple paths to different stations and it will start along one path. Along the path some blocks may be occupied that cause it to look for other paths and finally it will arrive in a station and it will wait. After waiting in the stations the process repeats itself. In some cases it will not find a station, but it will find the same block again where it started, for example in a circle or other loop. In that case it will just extend the route while driving.



Driving automatically without a route is very simple. Just go to the tab ‘Train’ of the ‘Train control’ and select ‘<No route>’ and start automatic driving via the option ‘Start’ somewhere (for example, on the ‘train control’ or ‘train overview’) just as with a normal train route.

All buttons and actions that you have used with routes can also be used. Thresholds in stations will work the same way and the field 'Info' will show what is happening and where the train is going. Finishing a route means drive on to the next station, but do not continue or do not extend the route anymore if no station is found.

Train type

The 'Reservation count' - the number of blocks a train tries to reserve in front of the current block of the train - is specified by the train type. Information about the waiting time in stations has to be specified in the station itself and also depends on the train type. That is why you could say that automatic routing is driving by train type instead of driving by route.

In the 'Train overview' you can see the train type in the column 'Type'. In the column 'Route' you can see if a route is specified or not.

Station

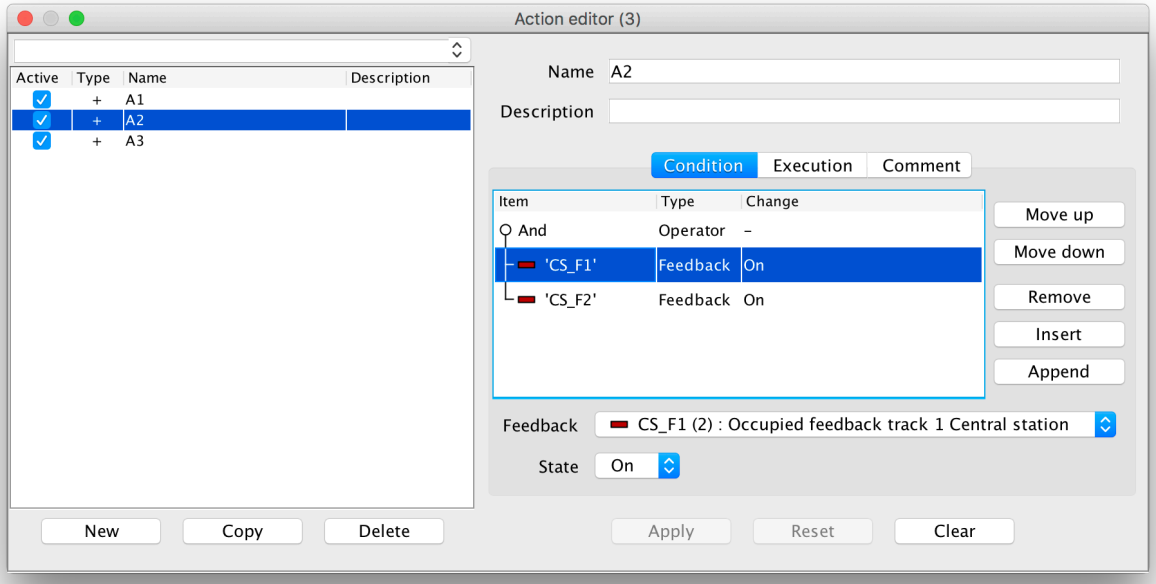
The station object, described earlier, has more properties that are related to automatic routing and that do not apply to normal routes.

- Selection - the selection of blocks in the station determines how a block in the station is chosen.
- Tab 'Train types' - specifies which train types will wait and if they might change direction after waiting in the station.

Note: Blocks in stations in which it is always necessary to change direction to continue, will only be visited by trains with either a locomotive or a control car on both sides.

Actions

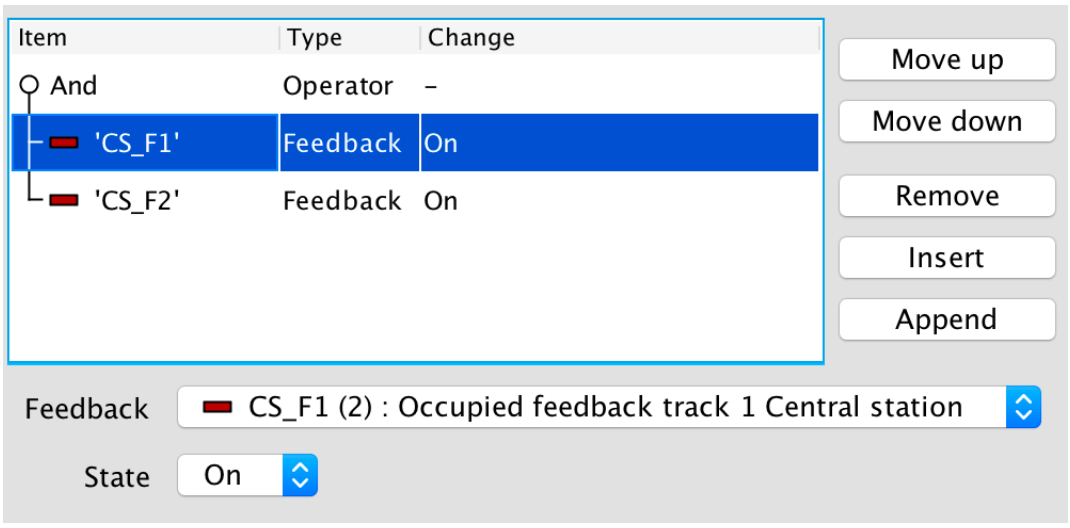
An event action - or simply called ‘action’ - is a list of things that will be executed (the action) when something happens (the event or condition).



You can create an action just as any other object in iTrain. After entering the name there are two important tabs called ‘Condition’ and ‘Execution’.

Condition

The condition triggers the action automatically and is the `when` or `if` part of the action.

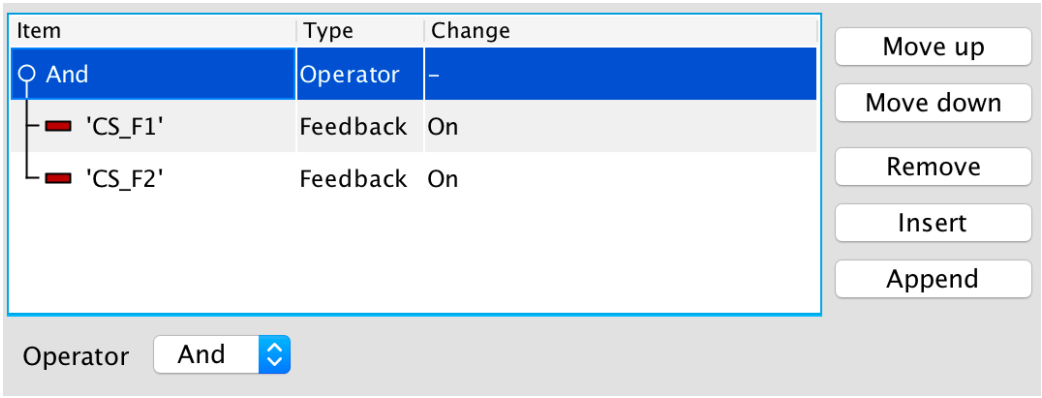


The conditions are listed in a table. With the buttons on the right side you can edit this table as with all tables in iTrain and remove, insert and append rows. After adding a row you have to select the type of the condition by double clicking in the table on the cell below ‘Type’. Depending on the type of condition an editor shows up at the bottom of the table with the input fields for this condition.

The most important types are described below:

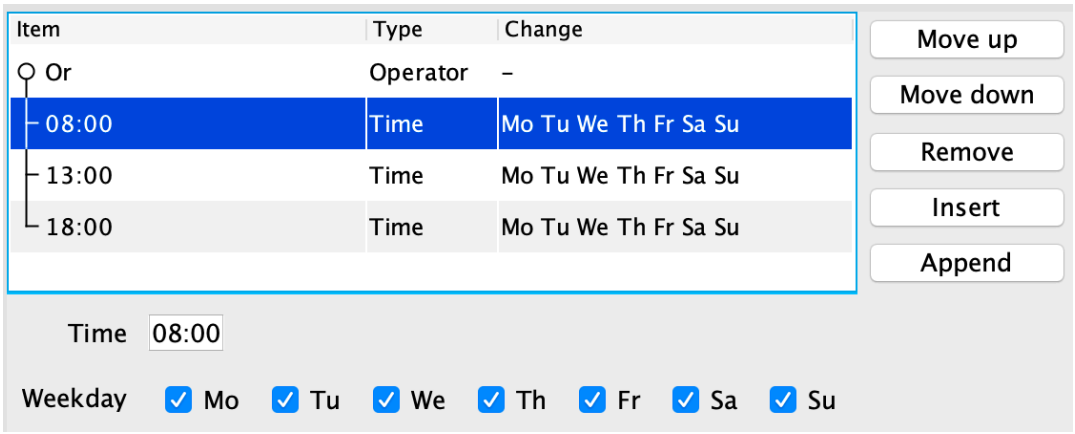
Operator

There are different types of conditions and they can be combined with operators so that all conditions must be fulfilled (AND) or one of the conditions must be fulfilled (OR)[65] to execute the action.



Adding an operator creates a group of conditions in the table. With the 'Move up' and 'Move down' buttons you can add conditions to and remove conditions from a group. It is possible to nest conditions in a tree to combine groups with AND and OR, but use this judiciously, because it will make it complex to debug.

Time

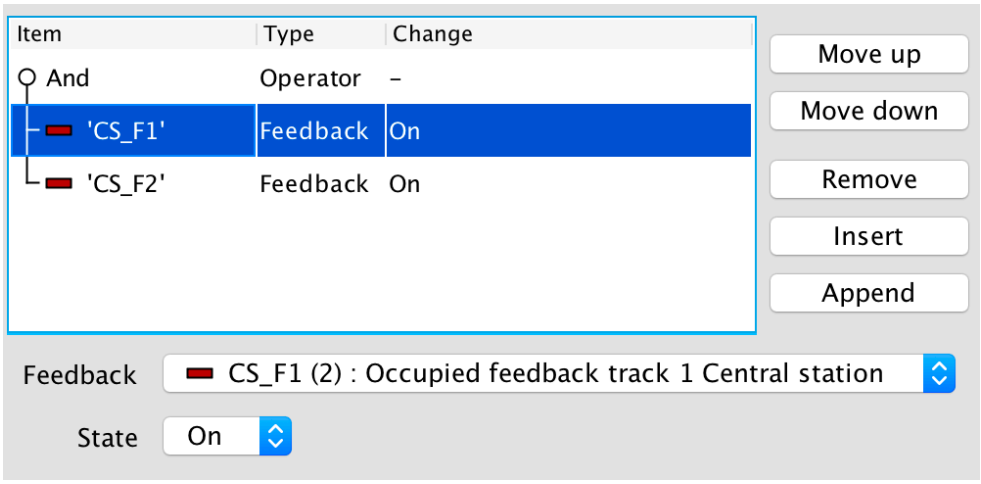


The condition 'Time' adds a time and the weekdays. The time is specified in hours and minutes between 00:00 and 23:59. The weekdays can be selected with checkboxes.

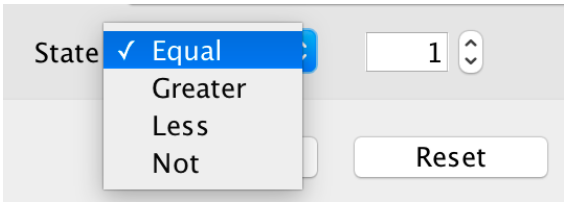
Tip: The condition 'Time' makes it possible to make a kind of schedule for a specific action, such as starting a route.

[65] If not explicitly specified a combination of conditions is automatically converted to OR.

Feedback



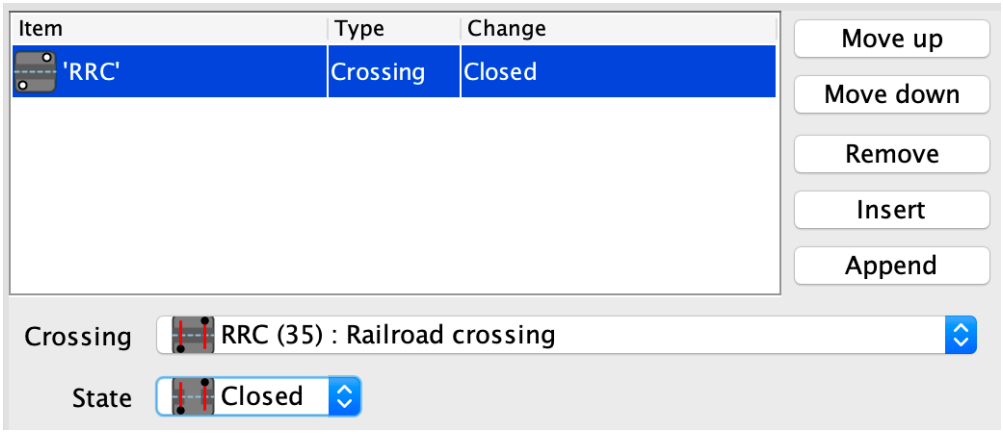
The condition ‘Feedback’ adds a feedback and its state. For most types of feedbacks this state will only be the ‘On’ state or the ‘Off’ state.



However, feedbacks of type ‘Value’ you can compare with a value via the choice ‘Equal’, ‘Greater’, ‘Less’ or ‘Not’.

Tip: The condition ‘Feedback’ can be used to let external buttons trigger specific actions.

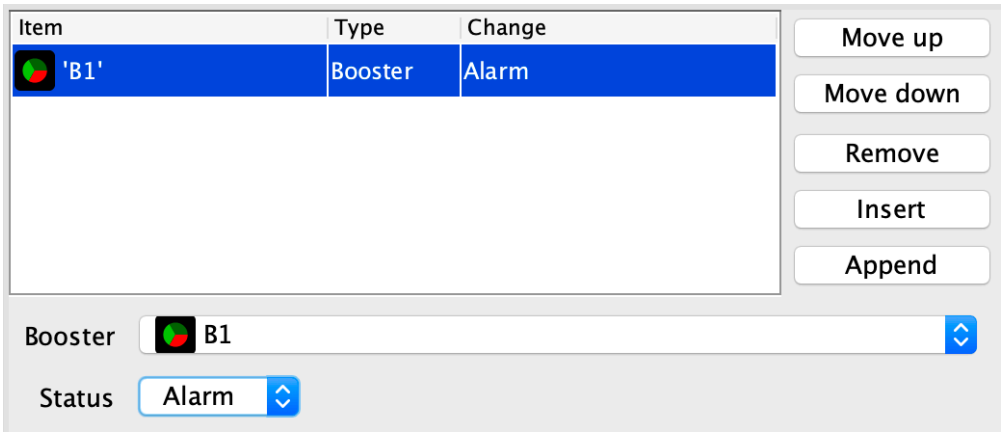
Accessory



There are a few conditions[66] to check the state of an accessory. You select an accessory from the list and the state of the accessory.

[66] Not all accessory types are available, to prevent that you interfere with the logic in blocks and start building your own.

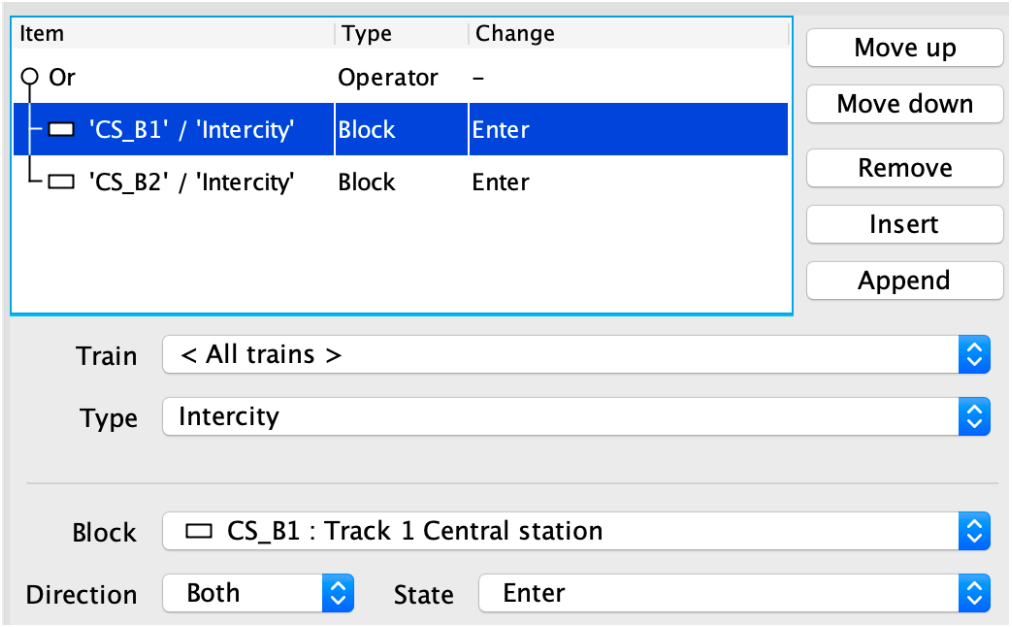
Booster



The condition ‘Booster’ adds a booster and a state.

Block

The condition ‘Block’ adds a block together with an optional direction side and a state.



There are six different states to choose from:

- Reserve - when the block is reserved by a train.
- Enter - when the block is entered by a a train.
- Wait - when the waiting time starts for the train, so after the train has stopped.
- Ready - when the waiting time is over and the train may start making reservations.
- Leave - when the the train is allowed to actually leave by the signal.
- Release - when the block is released by the train

In the box ‘Train’ you can specify whether the action should only be executed for a specific train, or may happen for any train, or only for all trains with a specific type.

In case of a condition in an OR group or if the condition is the first condition in the AND group, the train will be passed on to the execution of the action as the ‘Actual train’. This

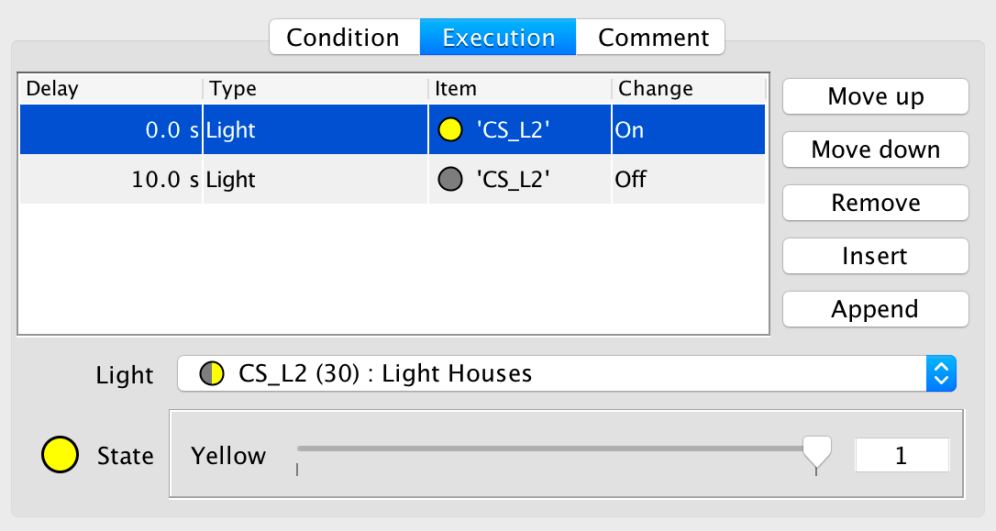
makes it possible to activate functions in the train depending on the actual train. For example, activate a horn for every train entering the tunnel if that function is available in the train.

Note: For actions triggered only by a feedback[67] there is no 'Actual train'. You either have to specify the train explicitly in the execution part, or better AND the feedback condition with a block condition (block condition must be first).

Execution

The tab 'Execution' contains a list of action items in a table that will be executed in order with some time in between. This time can be specified per item and is called the 'Delay'. This is the time that will be waited before the item will be executed. Items are always executed after each other and not in parallel.

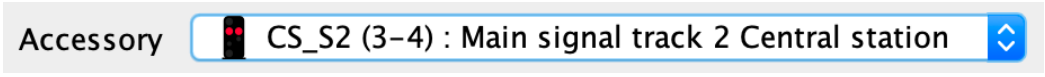
Tip: Except for the first item, it is recommended always using some delay, even if it is only 0,1 s (100 ms).



The actions items are listed in a table. With the buttons on the right side you can edit this table as with all tables in iTrain and remove, insert and append rows. After adding a row you have to select the type of the action item by double clicking in the table on the cell below 'Type'. Depending on the type of execution an editor shows up at the bottom of the table with the input fields for this action item.

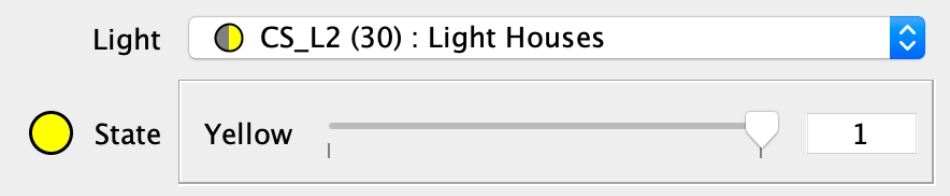
The most important types are described below:

Accessory
There are two ways to change an accessory with a different purpose:



- The general type 'Accessory' to toggle the state of accessory without specifying the new state. It behaves the same as you would click on an accessory in the switchboard with the mouse. You only specify an accessory from a list with all the accessories.

[67] Feedbacks only deliver their state, and optionally the locomotive number, to the block. Only the block is aware of the trains and not the feedback itself.



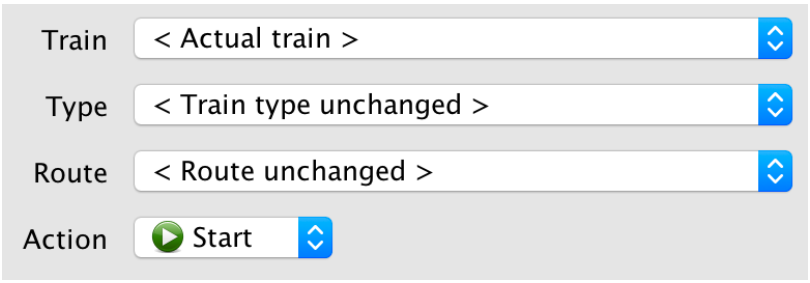
- The more specific types ‘Aspect’, ‘Turnout’, ‘Signal’, ‘Relay’, ‘Light’, ‘Sound’, ‘Crossing’ and ‘Decoupler’ to change to a specified state. In this case you specify both the accessory and the state.

Note: Although you can change the state of turnouts and signals for completeness, it is normally not necessary and not recommended doing so. Turnouts will be switched automatically by the routing system and signals will be automatically switched by the blocks. You should not interfere with this. Misuse followed by complaints will not be appreciated.

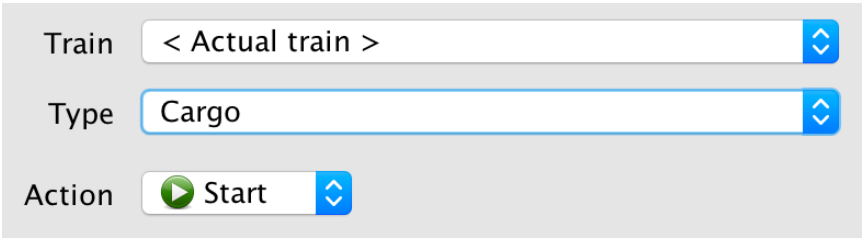
Track route

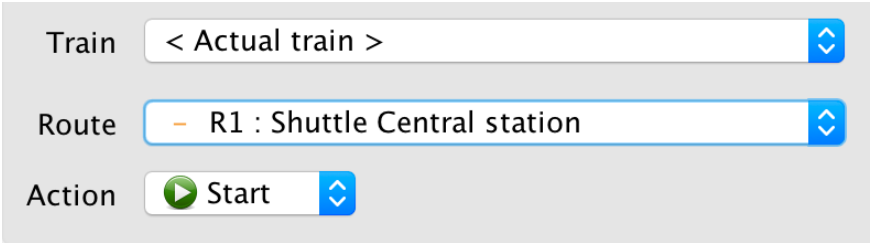
The type ‘Track route’ sets a track route into the active or inactive state (release).

Train route



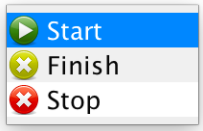
The type ‘Train route’ changes the routing of the train. You can change the train type or route and start, finish or stop driving automatically. First you have to select the train.



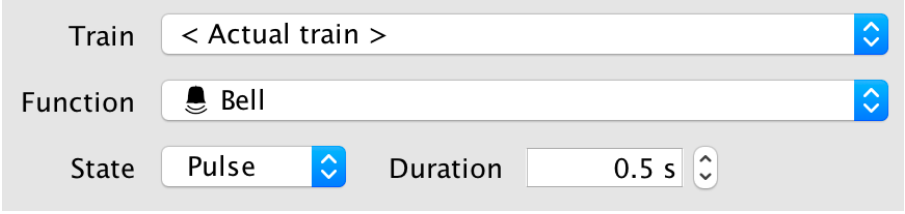


You cannot change both the type and the route, so once a type or route has been selected the other one will disappear. A type change will cause automatic routing to be selected.

Finally, you have to select the route action. Normally this will be ‘Start’ to start a route or start automatic routing, but it is also possible to ‘Finish’ or ‘Stop’ a route if this is necessary.

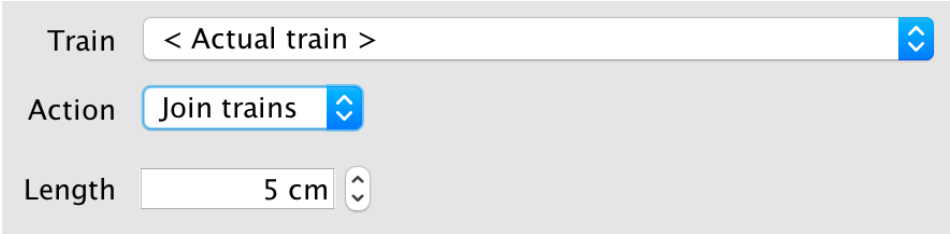


Function



With the types ending with ‘function’ you can change the state of a function of a vehicle. This can be on the whole train or on an individual locomotive or wagon depending on the selected type. In case of the train, you also have the option to select the ‘Actual train’ that is passed on by the condition. In case no actual train is available, the action will simply be skipped.

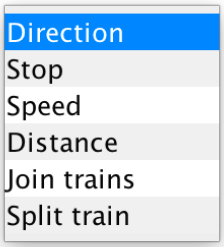
Shunt



With the types ending with ‘shunt’ you can directly control a vehicle for shunting. This can be the whole train or an individual locomotive or wagon depending on the selected type. In case of a train, you also have the option to select the ‘Actual train’ that is passed on by the condition. In case no actual train is available, the action will simply be skipped.

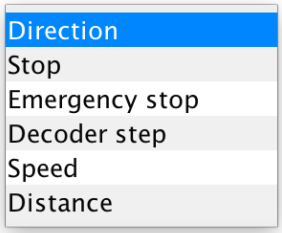
In case of ‘Train shunt’ there are the following options:

- Direction - to change the direction of the train.
- Stop - to stop the train.
- Speed - to set the desired speed of the train.
- Distance - to move with a minimum speed over a distance specified in ‘Length’.
- Join trains - join the specified train with another train in the same block. An additional distance can be specified in ‘Length’ to push the train x cm in the direction of the other train to make sure it couples well.[68]
- Split train - to split the current train into two trains. An additional position to split can be specified: Zero means split after the main loc. A negative offset means count from the rear of the train. The actual train direction defines the front and rear.



[68] You cannot use an additional action ‘Distance‘ before ‘Join trains’ as this does not allow you to bump into another train.

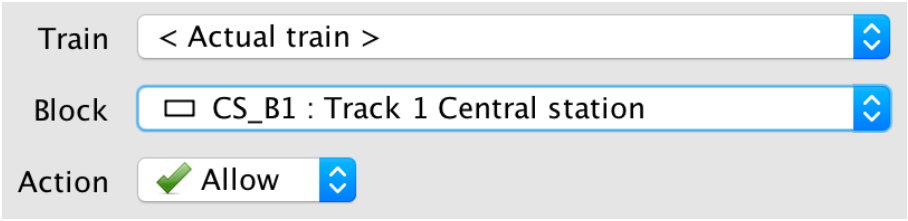
In case of 'Locomotive shunt' there are the following options:



- Direction - to change the direction.
- Stop - to stop with deceleration.
- Emergency stop - to halt without deceleration.
- Decoder step - to change the decoder step.
- Speed - to set the desired speed.
- Distance - to move with a minimum speed over a distance specified in 'Length'.

In case of 'Wagon shunt' the options are a subset of the 'Locomotive shunt'. They are not really there for shunting, but for controlling the decoder in the wagon directly (direction and step) in case of for example a vacuum cleaner.

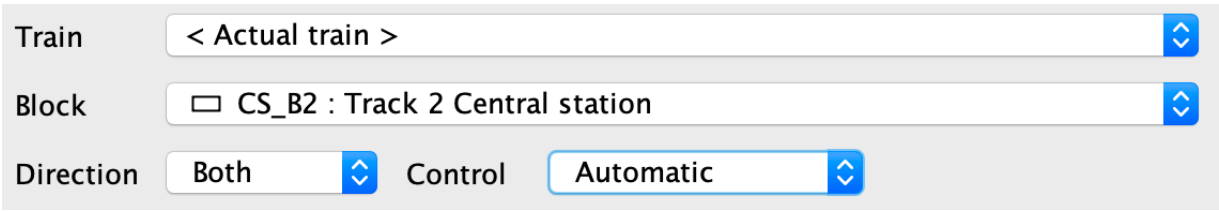
Train permissions



The type 'Train permissions' changes the permissions of the train including or excluding a block from the permissions list by selecting 'Allow' or 'Forbid'.

Note: Use this with care as the train permissions will be changed and this will not be visible directly in the switchboard leading to restrictions you might not be aware of later on.

Route to block



The type 'Route to block' will create an 'Instant route' for a train from the current block to the specified block.[69] In addition to the destination block you can specify the direction in this block to make it even more precise. The field 'Control' should normally be 'Automatic',

[69] The 'Route to block' action is similar to what happens when you drag a train to a block.

but in case you want to drive the train yourself with a throttle or handheld you can choose the mode ‘Semi-automatic’ or ‘Manual’.

Shunt to block

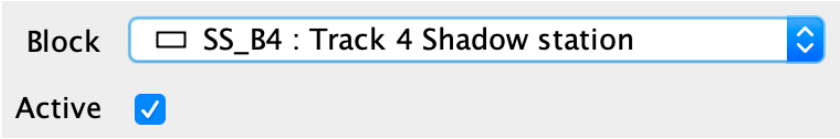
The type ‘Shunt to block’ is more or less the same as the previous ‘Route to block’, but in this case the train will shunt instead of using a normal route. This affects the reservation color, the speeds and the signal states.

Actual train

The type ‘Actual train’ changes the variable ‘Actual train’ to the train in the specified block. Normally the ‘Actual train’ is set by the condition, but with this action you change it during the execution. You can also initialize the variable when there is no condition that delivers a train at all.

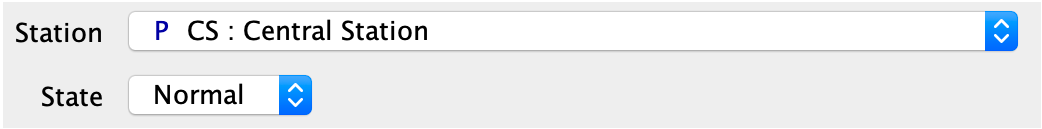
Tip: This looks like a simple execution item, but it is also a very powerful one in combination with other execution items.[70]

Block active



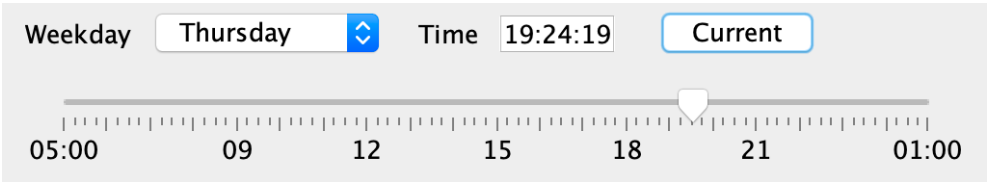
The type ‘Block active’ changes the active state of a block. This can be useful to limit the number of available tracks during the day or night for maintenance.

Station



The type ‘Station’ will put a selected station into a specific state as described in the chapter ‘Switchboard’, section ‘Station’.

Time

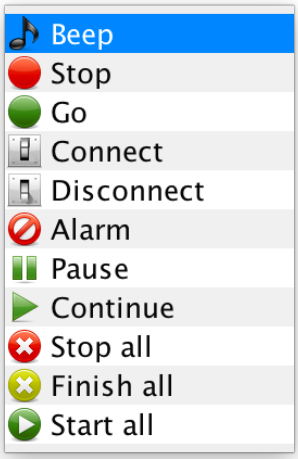


The type ‘Time’ will set the model time to a specific time and day of the week. The button ‘Current’ will set the fields to the current time and day of the week.

[70] The combination of the type ‘Actual train’ with ‘Shunt/Route to block’ makes it possible to automatically set a path from Block A to B that can be activated by two external buttons.

System

The type 'System' is a category of systems actions that influence the system as a whole. You can select the following actions from a list:



- Beep - to hear the system beep; can be used for testing.
- Stop - remove the track power on all interfaces.
- Go - set back the track power on all interfaces.
- Connect - connect all interfaces.
- Disconnect - disconnect all interfaces.
- Alarm - create an alarm in the system.
- Pause - pause the system, all signals blocked (red border).
- Continue - continue the system.
- Stop all - end all automatic driving immediately.
- Finish all - end all automatic driving gradually (routes will not be repeated anymore, driving without routes will end at the next station).
- Start all - start automatic driving for all active trains, use this judiciously to prevent chaos.

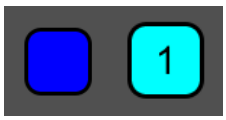
You will normally only use these system actions with a condition based on a feedback (external button) or maybe a condition based on time.

Command line

The type 'Command line' will execute a command on the so called 'Command line' (Windows) or in the 'Terminal' (macOS/Linux).

Note: This is an operating system dependent setting and will make the project dependent on an available command set of the computer.

Switchboard



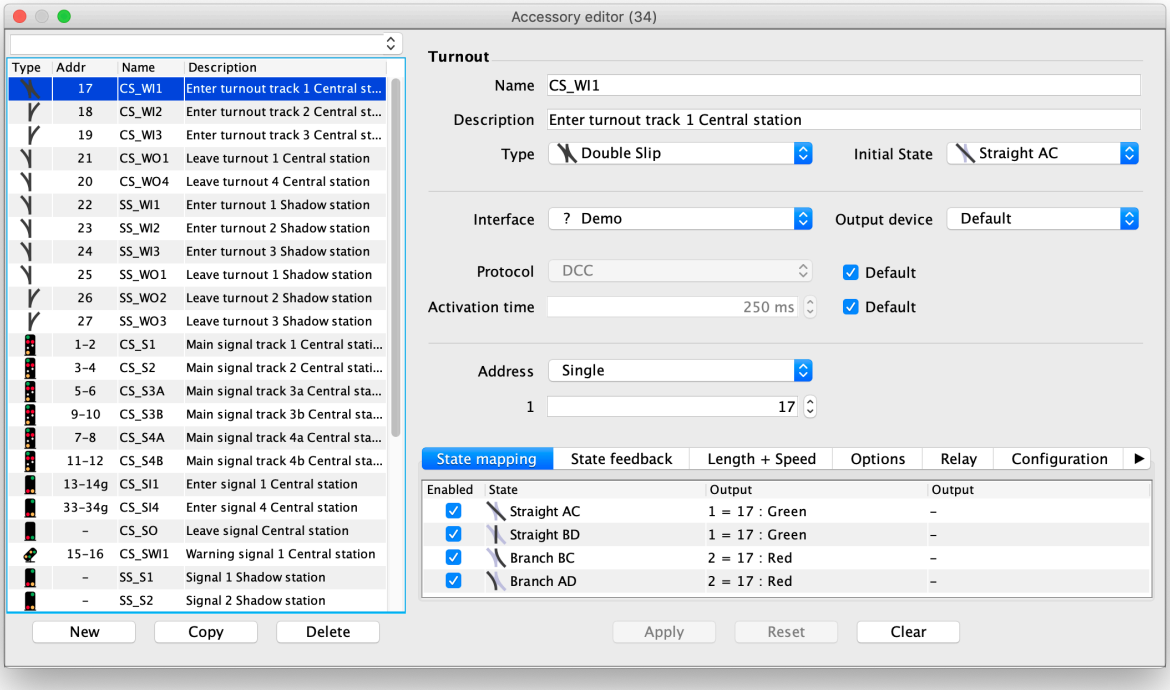
An action can be added to the switchboard. You can manually start it by clicking (without an 'Actual train') on the item. When the action is running, the color changes and it shows the number of active executions for this action. To interrupt and stop the previously started action, use 'Shift' + click. Now the action items will not be processed any further, but nothing is set back to their original values, so use this with care.

An action with an 'Actual train' may be started at the same time for multiple trains depending on the condition. In this case the number will be higher than one. To interrupt and stop all active actions, use 'Shift' + Alt + click.

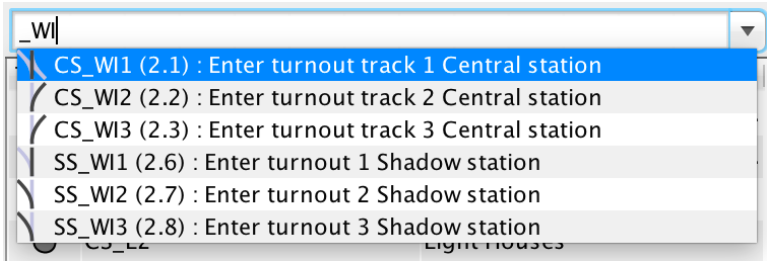
Tip: You can make an action inactive so that it will not react on the condition and you cannot accidentally click it. Use Alt + click to toggle the active state of the action in the switchboard. This will change the color of the action element.

Editors

Next to the interfaces, locomotives, wagons, train types, trains and train routes, you can also edit the feedbacks, accessories, track routes, actions, blocks, stations and boosters directly via the ‘Edit’ menu without going to the switchboard first. This is useful to quickly change a definition or to manage the list of objects.



A dialog will appear with at the left side a search field on top and below it a list with the objects of a type (accessories, feedbacks, track routes, etc.) known by iTrain.



To search an object just type at least two characters. If multiple items have been found a dropdown menu appears to select one of them. Only one item will be selected in the list below the search field.

List

To manage the list of objects you use the buttons below the list. You can add a new object by using the ‘New’ or ‘Copy’ button. In case of ‘Copy’, the currently selected object is used as a template and its definitions are copied to the new object. Only the name is adapted to create a unique name. The ‘Delete’ button removes the currently selected object from the list.

Note: The ‘Delete’ button really deletes the control object and it will remove all references to this object from other control objects. Only delete an object if you are sure it is not used or referenced elsewhere or else it could give unexpected results. Never

delete a control object and create a new one when you want to change the name or address.

The order in the list of objects on the left is also the order in which objects are stored internally and the order that will be used in dropdown boxes. You can change the order here in the following ways:

- Double clicking on the column header will sort the list based on the data in this column. This is a one time action, so you can first sort on column A and then on column B.
- Dragging items in the list to another position for fine-tuning.

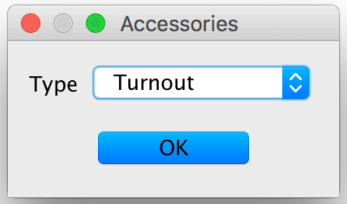
Properties
On the right side you find the specific object editor to change all object properties. You have to fill in at least a name and in many cases an address.

The buttons below the editor give some extra control over the editor:

- The 'Apply' button applies changes you have made in the input fields. This will immediately be reflected in all windows. If you select another object, the previously selected object will automatically be applied.
- The 'Reset' button discards the changes you made in the editor and reloads the fields with the current value. After an 'Apply' a reset will only discard changes made after the 'Apply'.
- The 'Clear' button clears all the fields.

Accessories

If you want to add a new accessory you have to select the type of accessory first. This can be done in two ways:



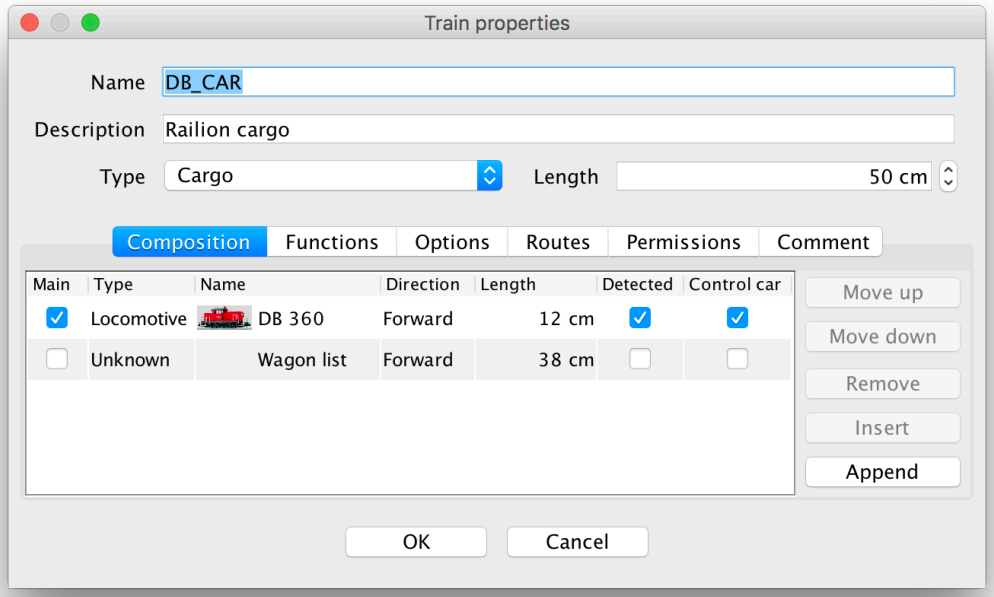
- Use the 'New' button. A dialog will appear to specify the type (Aspect, Turnout, Signal, Relay, etc.). After the selection you will see the appropriate fields on the right.
- Use the 'Copy' button to create a new accessory based on an existing accessory.

Blocks/Stations

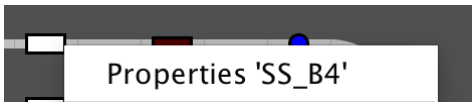
In case of the block or station editor, it is not possible to use the 'Auto fill' button or 'Create connections' popup menu now, because it is now not related to the switchboard where the information is coming from. This is only possible if you are in the switchboard editor and select a block element.

Properties

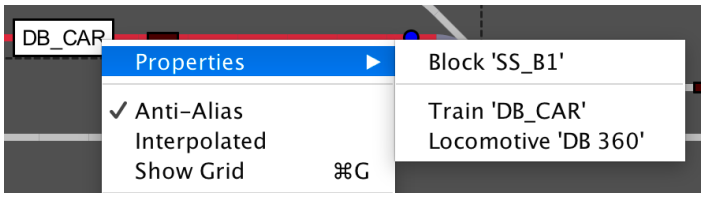
The editor allows you to edit properties of an object via the list of objects, but you need to look up the element again. It is also possible to edit an object directly by clicking the right mouse button while the mouse is above a specific object and selecting 'Properties' in the popup menus or context sensitive menu.



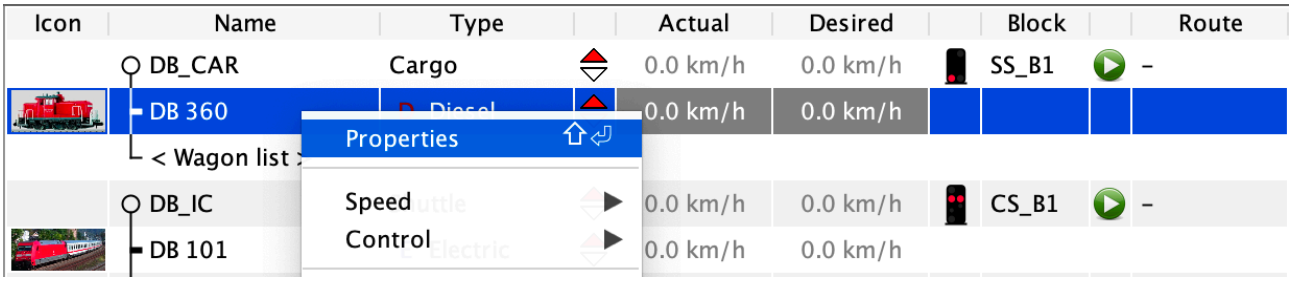
You will get a dialog with only the specific object editor and you can change properties instantly.



In case there are no trains in the block, you can select the block properties directly.



In case there are trains in the block, then first select ‘Properties’. A submenu appears with the block itself and all trains with their locomotives in this block so you can edit them directly.



In the ‘Train overview’ you can right click on a row and choose ‘Properties’ from the popup menu to edit the train, locomotive or wagon directly. In addition you can double click on a cell with a ‘Train type’, ‘Block’ or ‘Route’ to edit this object directly.

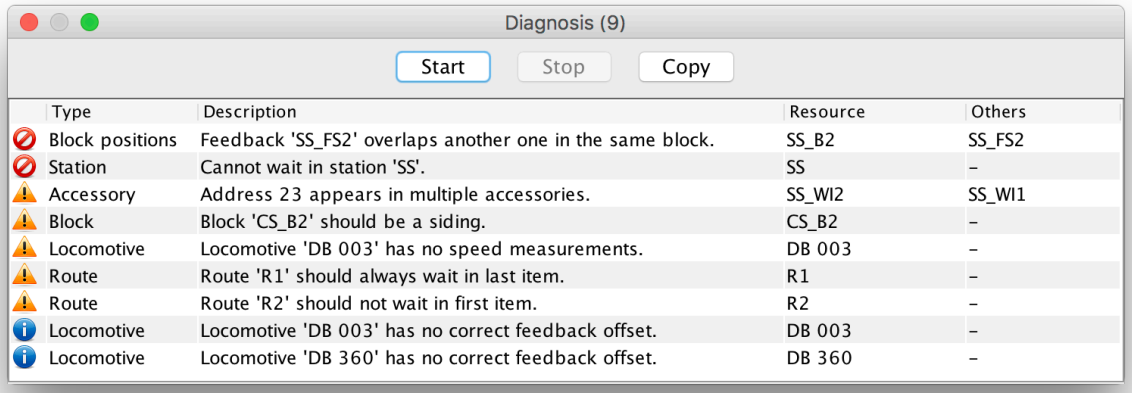
Double clicking on the labels ‘Block’, ‘Type’ or ‘Route’ in the ‘Train control’ also opens the specific editor for the object after the label.

Extra tools

The extra tools are available via the ‘View’ menu.

Diagnosis

Diagnosis is a tool to find problems or inconsistencies in the object definitions of your layout.[71] After activating the tool, your project will be scanned immediately and you will see the result in a table.



	Type	Description	Resource	Others
⛔	Block positions	Feedback 'SS_FS2' overlaps another one in the same block.	SS_B2	SS_FS2
⛔	Station	Cannot wait in station 'SS'.	SS	–
⚠	Accessory	Address 23 appears in multiple accessories.	SS_WI2	SS_WI1
⚠	Block	Block 'CS_B2' should be a siding.	CS_B2	–
⚠	Locomotive	Locomotive 'DB 003' has no speed measurements.	DB 003	–
⚠	Route	Route 'R1' should always wait in last item.	R1	–
⚠	Route	Route 'R2' should not wait in first item.	R2	–
ℹ	Locomotive	Locomotive 'DB 003' has no correct feedback offset.	DB 003	–
ℹ	Locomotive	Locomotive 'DB 360' has no correct feedback offset.	DB 360	–

The list has been sorted on the severity and the type of problem. There are three levels of severity:

⛔ Error - This is wrong and needs to be addressed as soon as possible.

⚠ Warning - This might be wrong or is incomplete. iTrain will perform better when solved.

ℹ Information - This is not a problem at the moment, but might be in the future.

In the ideal case the list is empty, but it is not a problem to keep some info messages as a reminder, for example for doing speed measurements for inactive locomotives later.

You can double-click on an item to open an editor with the underlying object/resource, so you can solve the problem. Use the button ‘Start’ to scan again and ‘Stop’ to stop scanning. The button ‘Copy’ is for putting the selected problem on the clipboard, for example to e-mail it or put it in the forum.

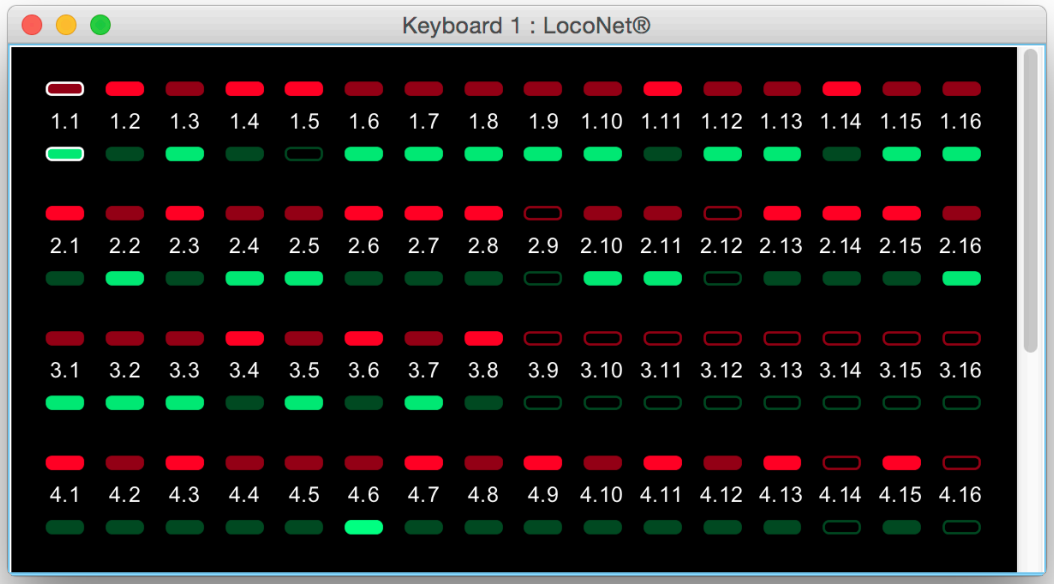
Tip: In case of unexpected behavior, always use the diagnosis tool with Cmd + ‘D’ before asking for help. This will save you and others time.

In some cases you may not directly understand the problem from the description, because it had to be short. However, you can look the meaning up on the forum and if it is not available ask for it. The number of checks will increase over time, so it is undesirable to keep this list up to date in the manual.

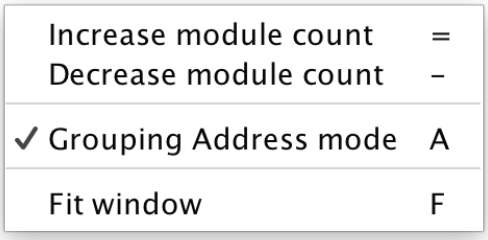
[71] It does not scan the switchboard itself so drawing errors will not be found.

Keyboard

The keyboard is an address based control tool for changing accessories like turnouts, signals, decouplers and relays. A keyboard is available for every interface in the project that supports pulse based accessories[72]. Every address has two buttons, 'red' and 'green'. When the button is filled, it is connected with an accessory object in the program. The highlighted buttons represent the state the accessory is in. Open buttons are not related to accessory objects in the program, but can still be activated for example to program a new signal.



A popup menu (right mouse click or Control key + click) allows you to increase/decrease the number of modules (also with '=' and '-' keys) or to toggle between 'Grouping' and absolute address mode (also the 'A' key).



The keyboard automatically scales to fit the width of the window. By using the 'F' key you fit the dialog window height to the current size of the keyboard.

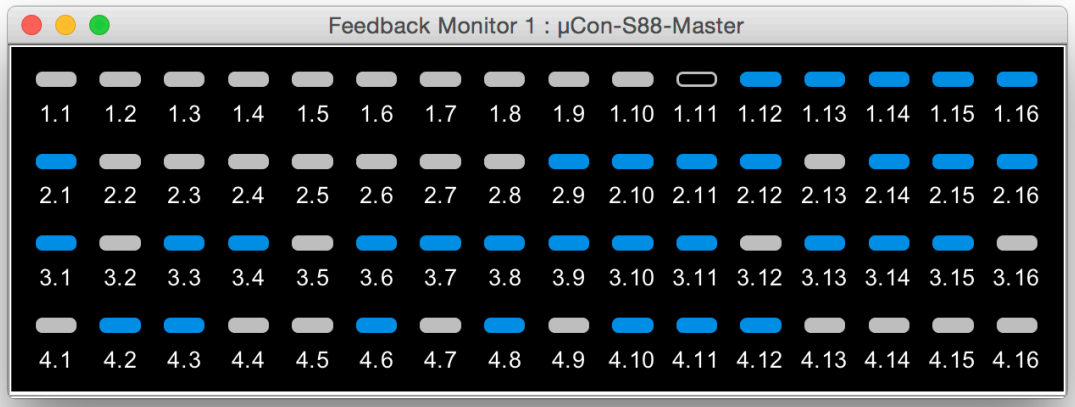
Note: Using the same address for two or more accessories is allowed in iTrain so they will all be switched when the state of one of them is changed. In some cases using the same address might indicate an error. Therefore addresses that are used more than once have a white border around the button to indicate this is a special case and you have to check if this is okay.

---

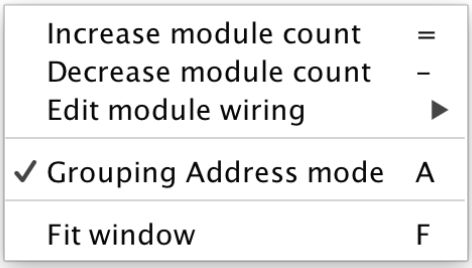
[72] The OM32 and OC32 have a mode for direct pulse based accessories, but the keyboard is only available for accessories switched via a connected 'Decoder63'.

Feedback Monitor

The feedback monitor is an address based monitor tool for showing feedback states. A feedback monitor is available for every interface in the project that supports feedbacks. Every button represents one feedback contact. If the button is filled it is connected with a feedback object in the program. A blue button means the feedback is activated. Open buttons are not related to feedback objects in the program, but still show the state they have received from the interface.

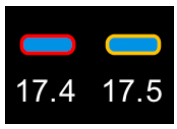


A popup menu (right mouse click or Control key + click) allows you to increase/decrease the number of modules (also with '=' and '-' keys) or to toggle between 'Grouping' and absolute address mode (also the 'A' key).



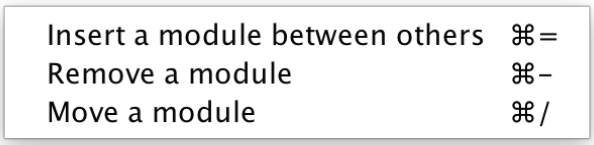
The feedback monitor automatically scales to fit the width of the window. By using the 'F' key you fit the dialog window height to the current size of the feedback monitor.

Note: Using the same address for two or more feedbacks normally indicates an error situation, and the button will have a red border/edge. In case feedbacks share an address because they have been connected via a relay (so that only one is actually wired at the same time), the button will have a yellow border/edge.

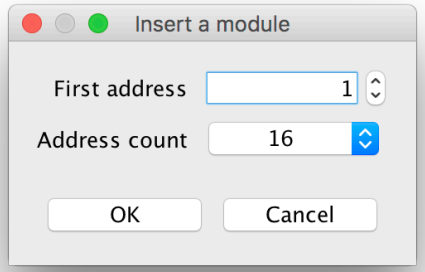


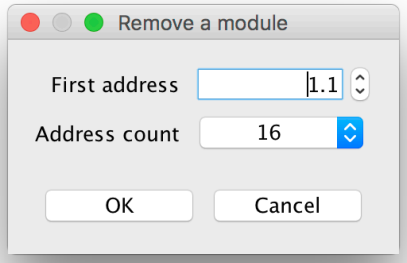
S88 Address numbering

Depending on the feedback system you use the addresses are fixed and configured per feedback decoder (for example LocoNet®) or in case of the widely used S88 system the addresses are assigned based on the location of the feedback module in the string of modules connected to the S88 bus. So adding or removing a module in the string will change the addresses of all feedbacks connected to modules later in the string.

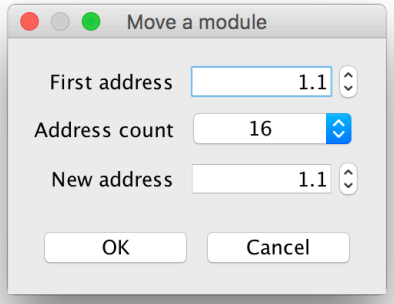


The menu ‘Edit module wiring’ in the popup menu of the feedback monitor has three menu items to add or (re)move a module, so that all addresses of the following feedbacks will be shifted up or down automatically.





A dialog will appear to select the first address of the module to be inserted or (re)moved and the number of addresses to be inserted or (re)moved (‘Address count’).



In case of a move, an additional “New address” field needs to be filled in to set the new address of the module. The result can be seen in the feedback monitor immediately after pressing the ‘OK’ button.

Extra[73]

The menu ‘View’ -> ‘Extra’ allows you to add a copy of a train overview or train grid in a separate window. You can put the extra views on a second monitor.

Depending on your license, it is also possible to add one or more switchboards for extra monitors. Here you can select another tab to show other parts of your layout at the same time. This will be useful if you are controlling a large layout that doesn’t fit on one screen.

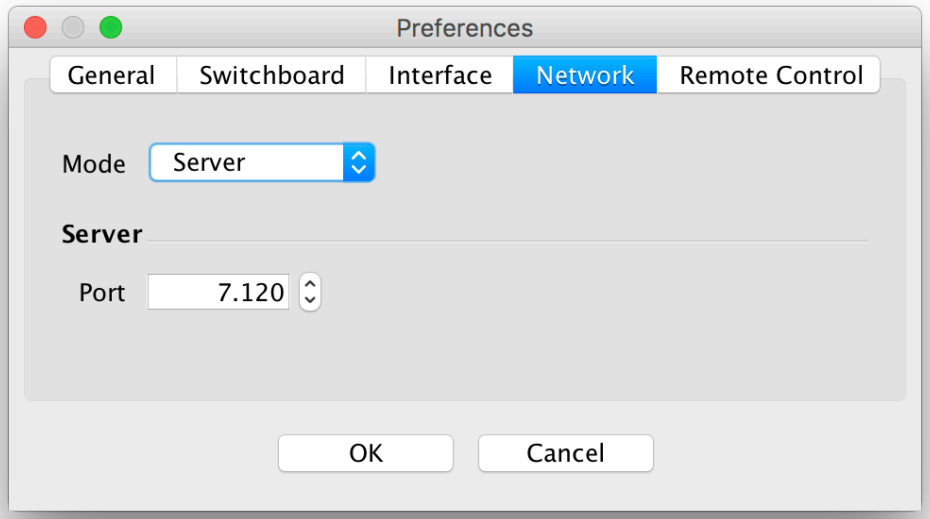
Tip: When you are dividing your view over multiple monitors, it might be useful to reduce the number of views in the main window via the menu ‘View’. For example, only showing the ‘Switchboard’ here and hiding the ‘Vehicles’ , or the other way around.

[73] This option is only available in the Plus edition (and higher) of iTrain. Extra switchboards are only available in the Professional edition.

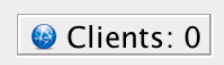
Network

By default iTrain runs in stand-alone mode, but it is also possible to run iTrain in a network. In that case you can start iTrain multiple times on different computers and control the layout with multiple people or at different sides of the layout. In a network, one iTrain program is the server and it directly controls the layout. The other iTrain programs are the clients and they communicate via the server with the layout over your local network.

Server mode[74]



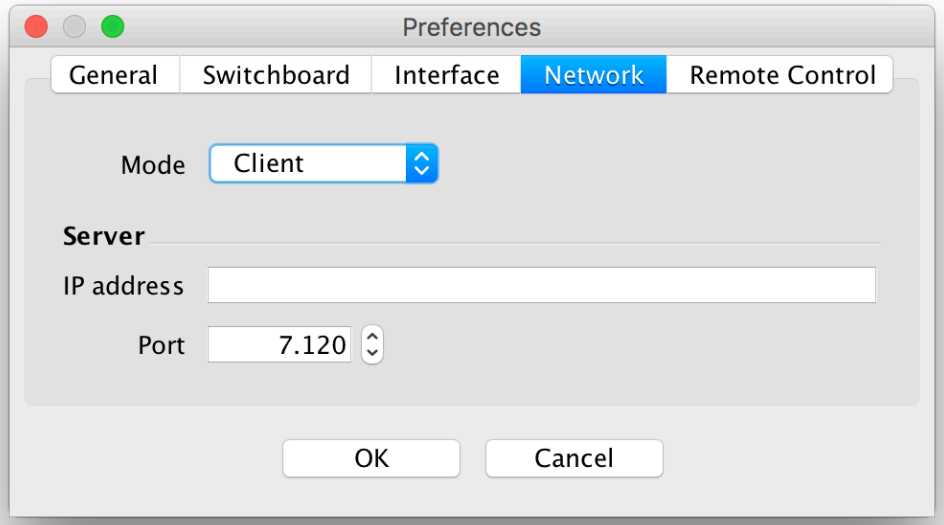
To start a network you first have to start one iTrain installation (the one that is connected to the layout) and in the preferences go to 'Network'. Change the 'Mode' to 'Server' (leave the port to 7120 unless it conflicts with another service already running on that port). iTrain is now in server mode and will wait for incoming connections from other iTrain programs in 'Client' mode. This setting will be remembered so that next time you start this iTrain it will immediately be a 'Server' until you change the mode to 'Stand alone'.



The fact that iTrain is in 'Server' mode can also be seen on the status bar where an icon followed by the text 'Clients: 0' appears. By hovering over this item in the status bar, a tooltip will appear with the name and IP address of the server. If the computer has multiple network interfaces such as a wired 'Ethernet' port, a wireless network (WiFi), etc. it is possible that two or more IP addresses will show up.

[74] Networking in server mode is only available in the Professional edition of iTrain.

Client mode



Now that one instance of iTrain is running in 'Server' mode you can start other iTrain programs on other computers, but they will by default start in 'Stand alone' mode. Via the same preferences and the tab 'Network', now select 'Client' mode and after the IP address fill in the IP address (or the name if this is known by the 'Client' computer) of the iTrain server. You can also leave it empty to automatically connect to a running iTrain server (when starting iTrain a look up will be done on the network for a server with the same version as the client). 1

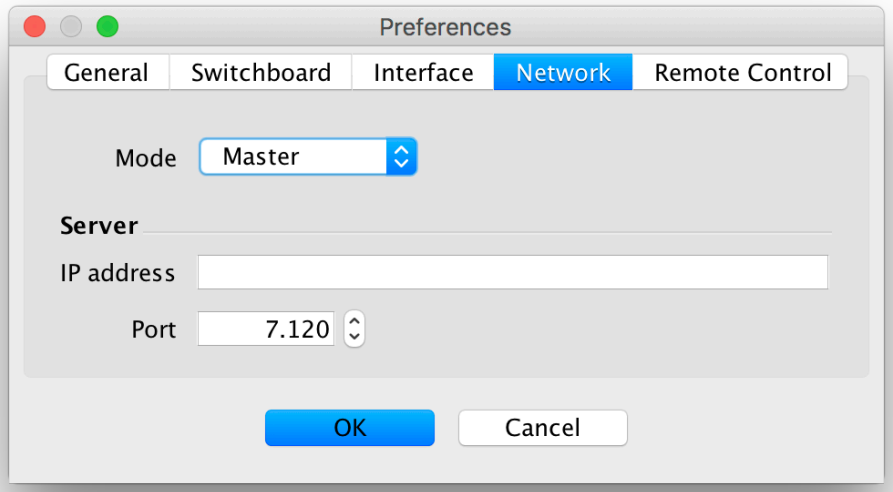
You have to close this iTrain program and restart it to run it in 'Client' mode and it will automatically try to connect to the server when it is started again. If it cannot connect to the server you will get an error message and it will start in 'Stand alone' mode again, but it will try to start in 'Client' mode again the next time you startup until you change the mode to 'Stand alone' again.

In 'Client' mode there are some restrictions:

- You cannot open and save files, because this is controlled by the server.
- You cannot change the interface or control its parameters, because this is the job of the server.
- You cannot edit the switchboard. Changes on the server will be sent to the client immediately.
- New objects like a new locomotive must be added by the server, but changes in the definition like the name or address can be done on the client side.

Master
There is also a special 'Master client' mode that does not have these restrictions and can be used to do everything you would normally do on the 'Server'.[75] This is useful in the following cases:

[75] Allowing all the editing functionality on the client had a big impact on the redesign of the interfaces and how resources like images and sounds are selected. It is possible now to select the server's serial ports and images/sounds (including previews) on the client.



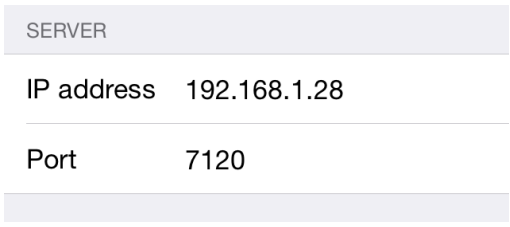
- The ‘Server’ does not have a screen or it only has a small screen that is not well suited for editing.[76]
- You want to move freely while editing your layout with your ‘Master’ client laptop attached wirelessly to the network. The ‘Server’ computer has cables for USB, network and/or monitor attached to it and cannot move.

Tip: A ‘Master client’ requires the same license as the ‘Server’, but a normal ‘Client’ with restrictions can be used by anyone with iTrain on his computer even without a license. So in a club, members can connect to the ‘Server’ without the need to install a license on their laptop. You only need access to the local network.

When closing the server, all clients will automatically be informed and it is not useful to continue working with iTrain in ‘Client’ mode. A client however, can quit at any time without affecting the server.

iOS

You can use iOS devices such as an iPhone, iPad or iPod touch to control locomotives with the latest ‘iTrain 5 remote’ App from the AppStore. The app will act as a client and connect to iTrain in server mode. In this case it is important to set up the network settings correctly. After installing the App you have to go to the general ‘Settings’ on the device and select the App.



Below the ‘Server’ you have to fill in the IP address of the server before you can start the App.

[76] An example could be a Raspberry Pi or a Mini-PC built into the layout or module together with a wireless access point for the clients.

Android

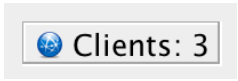
There is an 'iTrain 5.0 remote' App for mobile devices with Android 5.0 or higher to control locomotives. The App can be downloaded from the Google Play Store.

No further network settings on the mobile device are necessary, because the App will automatically connect to an available iTrain server on the network.[77]

N.B. Make sure that you download the Android App with the same major and minor version as your iTrain server version, or else it will not work. Only the first two numbers are important, so 5.0 is different from 5.1. The third patch number may be different, so 5.0.4 and 5.0.5 are compatible.

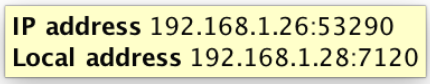
Status

On the server you can monitor all the connected clients via the status bar. You will see the number of connected clients immediately and by double-clicking on it a new dialog will appear with a list of all network clients.



In the title of the dialog you see the name and the IP address(es) of the server and the number of clients connected. Every connected client has a row in the table with its name, IP address, the version of the client, the operating system, the programming language and the time the connection started. By clicking on the red icon before the name you can disconnect individual clients.

In a tooltip per row you can find more details, like the port used on the client and the local or server address (and port) to which the client has been connected.



Note: iTrain has been designed in such a way that using more than one client does not add much extra overhead (just one thread) for the server. Only at the start of a connection a lot of data will be exchanged to inform the client of the current status of all objects and the layout. Afterwards the clients are only informed about the changes.

[77] Sometimes it necessary to make a change in the firewall on your computer to allow UDP port 7120 to find the iTrain server.

Appendix A: Defined keys

A modifier key like Shift, Control, Alt and Command is used by holding this key down while pressing another key at the same time.

We use the following abbreviations:

Shift - for the Shift key

Ctrl - for the Control key

Alt - for the Alt key

Cmd - for the Command key (on Windows and Linux this should also be read as Control key).

Globally defined keys

Key	Action
Escape	Stop: Takes power off the tracks, all vehicles will stop (use F5 to continue)
F1	Reserved for Help (no online help available at this time)
F2	Focus on train overview
F3	Focus on train panel
F4	Focus on switchboard
F5	Go: Put power on tracks again and vehicles will continue
F6	Toggle between going online/offline
F7	Toggle between pause/continue the system
F8	Finish all automatic driving gradually
Shift + F1	Show the manual in a PDF viewer
Cmd + F2	Edit Locomotives
Cmd + F3	Edit Trains
Cmd + F4	Edit Switchboard
Cmd + F5	Edit Blocks
Cmd + F6	Edit Interfaces
Cmd + F7	Edit Feedbacks
Cmd + F8	Edit Accessories
Cmd + F9	Edit Actions

Key	Action
Shift + Cmd + F2	Edit Wagons
Shift + Cmd + F3	Edit Train Types
Shift + Cmd + F4	Edit Train Routes
Shift + Cmd + F5	Edit Stations
Shift + Cmd + F6	Edit Boosters
Shift + Cmd + F8	Edit Track Routes
Cmd + O	Open File
Cmd + S	Save File
Shift + Cmd + S	Save File as ...
Cmd + I	Import ...
Cmd + E	Export ...
Cmd + P	Print Switchboard
Cmd + D	Diagnosis
Shift + Cmd + F	Full Screen (only on macOS)

Train keys

These keys will be available when the ‘Train overview’ or the ‘Train control’ has focus. To give the ‘Train overview’ focus, press F2 and to give the ‘Train control’ focus, press F3. Every time you press F3 again another ‘Train control’ gets the focus.

Key	Action
H	Halt or emergency stop
0	Brake to speed zero and stop.
1-9	Set speed in steps of 10 km/h (3 = 30 km/h).
Shift + 0-9	Set speed in steps of 10 km/h starting at 100 km/h (Shift + 4 = 140 km/h).
-	Decrease speed to the previous step of 5 km/h below the current speed (starting at 72 km/h it will be 70, 65, 60).
+	Increase speed to the next step of 5 km/h above the current speed (starting at 72 km/h it will be 75, 80, 85).

Key	Action
Backspace	Brake to speed zero and stop when driving, and when stopped, change the direction (so 0 or D depending on current speed).
D	Change the direction.
A	Change to automatic control. A train may start to drive if allowed.
S	Change to semi-automatic control.
M	Change to manual control.
Shift + Delete	Remove a block from a train. At first a reserved block is removed (at the front), then a release block (from the end) and finally the current control block.
Shift + F5	Start a route for a train (when a route is attached).
Shift + F6	Stop a route and/or stop the train.
Shift + F7	Finish the route, but do no repeat it anymore.
L	Main light (first available so including the front).
Shift + L	Main light on the rear side
V	Head lights (first available so including the front).
Shift +V	Head lights on the rear side.
C	Cabin light (first available so including the front).
Shift + C	Cabin light on the rear side.
I	Interior light
W	Driving wheel light
Shift + W	Engine light
B	Sound of bell
F	Whistle
N	Signal horn
E	Engine sound
Shift + E	Blow out cylinder
G	Generator
Shift + G	Relay steps

Key	Action
Z	Fan
Shift + Z	Pump
J	Compressed air
Shift + J	Brake sound
Q	Buffer impact
Shift + Q	Couplers engaging
X	Door (first available so including Door opening)
Shift + X	Door closing
U	Announcement (first available so including Station announcement)
Shift + U	Train announcement
K	Coal shoveling
Shift + K	Grate being shaken
O	Smoke
R	Direct control (no acceleration/deceleration).
Shift + R	Slow (decoder steps are related to lower speeds).
T	Decoupling (first available so including the front).
Shift + T	Decoupling on the rear side.
P	Pantograph (first available so including the front).
Shift + P	Pantograph on the rear side.

The numeric key pad can also be used to control trains. The '-' and '+' and the 0-9 keys behave the same as on the main keyboard. Extra functionality only available on the numeric pad is summarized below:

Key	Action
. or ,	Change the direction of the train.
*	Reserve the next block for a train.
/	Remove the most recent reservation on the front.
Shift + *	Add a 'Release' block to the back of a train.
Shift + /	Remove the last 'Release' block on the back of a loc.

Tip: Some of the keys attached to a train can also be used globally (without focus) when used in combination with the Alt key. In this case, the train selected in the 'Train overview' is controlled. It concerns the '-', '+', '', '/' and Backspace key.*

In case of the 'Train control', there are some additional keys to use in the Train Grid:

Key	Action
Alt + Cursor keys	Change the size of the train grid. Left and Up to decrease the size and Down and Right to increase the size.
Alt + F3	Fill the 'Train controls' with the available trains.

In case of the 'Train overview' there is one specific key:

Key	Action
Enter	To put the currently selected train in the overview, in the (first) 'Train control'.

Switchboard control keys

These keys will be available when the switchboard has focus. To give the switchboard focus, press F4.

Key	Action
Cmd + G	Toggle between show grid and hide grid.
Cmd + B	Fit switchboard on both width and height.
Cmd + W	Fit switchboard on width of the switchboard.
Cmd + E	Fit switchboard on height of the switchboard.

Keys that are defined and attached to switchboard elements can also be used here. It is better not to use the globally defined keys or the switchboard keys for this.

Switchboard edit keys

The following keys are can be used when editing the switchboard (Cmd + F4):

Key	Action
Cmd + B	Fit switchboard on both width and height.
Cmd + W	Fit switchboard on width of the switchboard.
Cmd + E	Fit switchboard on height of the switchboard.
Cmd + cursor keys	Select or turn the elements on the element toolbar on the right.
Cmd + T	Turn the selected element on the element toolbar clockwise.

Key	Action
Cmd + R	Turn the selected element on the element toolbar counter-clockwise.
Cmd + Enter	Select the object in the object browser that is attached to the selected element in the switchboard.
Alt + cursor keys	Change the size of the grid.
Shift + cursor keys	Select multiple cells.
Shift + Alt + cursor keys	Change the size of a resizable element on the grid.
Shift + Cmd + cursor keys	Move the selected elements over the switchboard.
Space	Put the selected element on the element toolbar on the switchboard.
Enter	Edit properties of the selected element in the switchboard.
T	Turn the selected element on the switchboard clockwise (only for single cell selection).
R	Turn the selected element on the switchboard counter-clockwise (only for single cell selection).
Shift + R / Shift + T	Turn the selected elements 90 or 180 degrees depending on the selection .
O	Toggle between showing and hiding the object browser.
C	Compress or trim the switchboard so that empty cells at the borders are removed.
Delete or Backspace	Delete an element on top (single cell selection) or delete all selected elements.
Shift + Delete or Backspace	Delete an element below (single cell selection) or delete all selected elements.
G	Grouping: assign an element to a block/turnout or, if no group has yet been selected, select a block/turnout.
Shift + G	Ungroup: Remove an element from the selected block/turnout.
Cmd + Z	Undo the most recent remove, cut or move operation (only works before a selection change).
Cmd + X	Cut the selection. Delete the selected elements and put them on the clipboard.

Key	Action
Cmd + C	Copy the selection. Put the selected elements on the clipboard.
Cmd + V	Paste from clipboard. Put the elements on the clipboard on the switchboard (and widen the grid if necessary).
L	Replace current item with a line element.
B	Replace current item with a block element.
P	Replace current item with an arrow element.
F	Replace current item with a feedback element.
S	Replace current item with a signal element.
Shift + B	Replace current item with a buffer element.
Shift + E	Assign a Text element to the selected cells in the switchboard.
Shift + P	Assign a Platform element to the selected cells in the switchboard.
Shift + S	Assign a Station element to the selected cells in the switchboard.
Shift + H	Assign a Building or House element to the selected cells in the switchboard.
Shift + I	Assign an Image element to the selected cells in the switchboard.
Alt + R	Rename a tab
Alt + I	Insert a tab at the current position
Alt + '+'	Add/append a tab at the end
Alt + '-'	Remove the current tab
Alt + Page up	Go to the next tab
Alt + Page down	Go to the previous tab

Tip: In general, the Command or Control key in combination with a normal key, affects the element toolbar on the right. The Alt key combined with cursor keys is for sizing. The Shift key is associated with selections or cells that are resizable.

All tool windows

For tool/floating windows with for example a train grid, switchboard or keyboard, the following keys are defined:

Key	Action
Alt + F	To fit the window size to the preferred size of the contents.

Key	Action
Escape	Stop: Takes power off the tracks, all vehicles will stop (use F5 to continue)
F5	Go: Put power on tracks again and vehicles will continue

Appendix B: Feedbacks

Feedbacks are very important in computer control. They are the eyes or sensors that feed the computer with information on what is happening on the layout. For a program like iTrain, very sophisticated feedbacks are not necessary. Only an ‘on’ or ‘off’ or ‘1’ or ‘0’ is enough. We call these binary sensors. They detect if something is pressed or not, something is occupied or not, or something is passed or not. The ‘something’ will be determined by the logic in iTrain.

We can distinguish two types of feedbacks:

1. Occupancy - they inform the system if (part of) a track is occupied.
2. Momentary - they inform the system that an event happened, for example that (part of) a train was passing.

Occupancy

An occupancy sensor or feedback has the advantage that it can be read out any moment to tell if (part of) a track is occupied. No train movements are necessary. I recommend this type of feedback over momentary feedbacks when used in association with blocks.

To create an occupancy feedback, it is generally necessary to isolate one rail of a track. This can be done by using plastic connections or plastic isolators or by cutting a rail. The isolated part should have a wire that can be connected to a device or decoder that can read feedbacks outputs. Depending on the rail type, there are two methods:

- 3-rail (Märklin HO) - the isolated rail is normally connected to ground and can be attached to any feedback decoder based on ground detection (for example S88). An occupancy detector works the same as a ‘contact rail’.
- 2-rail (others) - the isolated rail still needs to feed the track with power, so that locs will run, so the isolated rail is connected via a current detection system to the normal power and the current detection system is connected to a feedback decoder. In many cases, both devices are combined into one. Only locs and wagons with light or other power consuming devices are detectable.

Momentary

Momentary contacts just give a short pulse when something is activating them. An example is a reed contact that is activated when a magnet is above it. This can be used to detect a train by attaching a reed contact between the rails in the track and putting a magnet below a loc or other part of a train. This method is not appropriate for smaller tracks scales like N and Z.

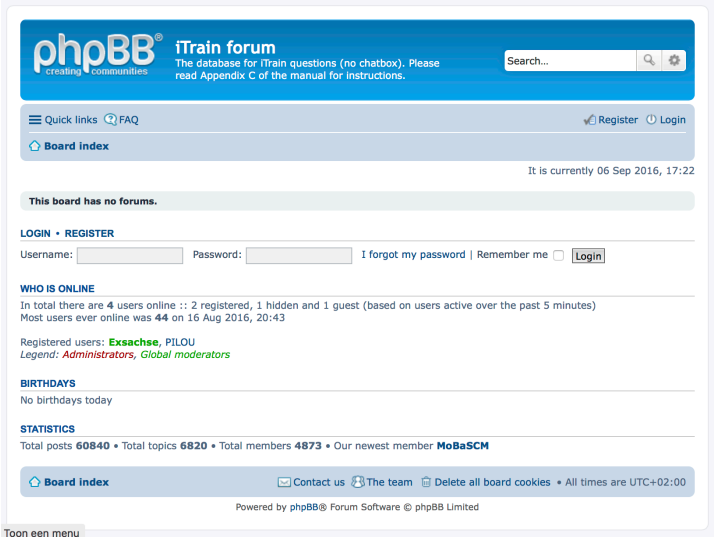
iTrain supports both types of feedbacks, but I personally prefer occupancy feedbacks over momentary feedbacks to detect trains in blocks. The choice you make will depend on your requirements, your scale and track and your investigation of costs and benefits. Be informed by what is available in the market and look up the web for more information.

Appendix C: iTrain user forum

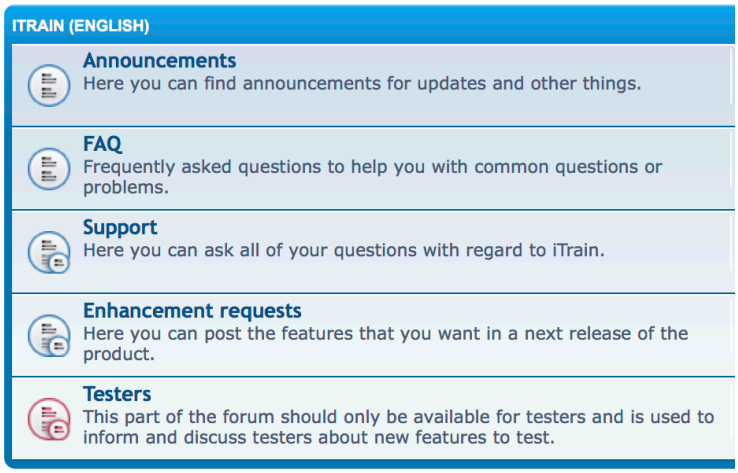
The iTrain user forum is there to inform you about and give you support on iTrain. It is available at https://berros.eu/forum/ via a web browser. It is not an open forum and only registered users can read it. You have to register with a self chosen username and password. Your registration will be verified by the webmaster so you have to choose a decent username and e-mail address or it might be refused.

Note: Choose the same e-mail address as with the registration or purchase of iTrain on the website so we know from who the request is coming and we can approve it soon. An unknown e-mail address might not be approved as the forum is only meant for registered iTrain users.

Initially it will start in English, but you can set the language and timezone you prefer.



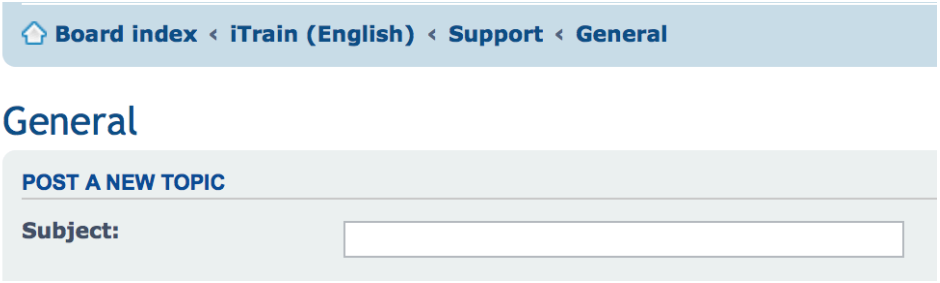
The iTrain forum has been divided into different sections each containing a sub-forum for a specific language. Each sub-forum contains the same division so you can read and ask questions in your preferred language by choosing the right sub forum.



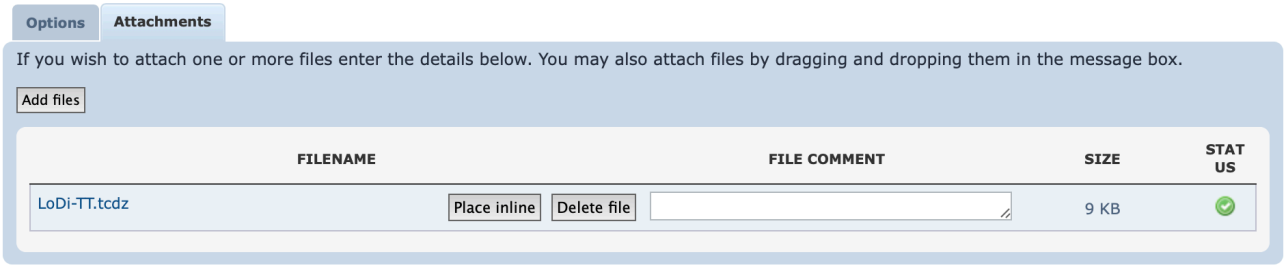
The first section ‘Announcements’ is a section to keep you up to date with the latest information about iTrain, such as releases and trade fairs. You cannot start an announcement, but only react on it. The same holds for the second section ‘Frequently

Asked Questions'. In this section there are answers to questions that a lot of customers have and it is recommended to read this first.

The third section 'Support' is really for support questions. Here you can ask your own questions or respond to others. First choose the right category. Then use 'NewTopic' to add a new topic.



Always choose a good subject so your question can be found easily by other users. Also provide the information that is necessary for someone else to help you. In many cases this involves adding your own layout project so people can see what you have been doing. The layout must first be saved within iTrain ending the filename with `.tcdz` and then you can add it to the forum via the second tab 'Attachments' by pressing the button 'Add files'.



Try to formulate your question well and read it again before you submit it. Well formulated questions are answered more quickly, because there is less need for additional questions that will take more time. For additional questions not related to the current subject you should create a new topic and not continue the current thread. The questions also function as a database for new users in which they can find answers and the better the subject names are and the shorter and more to the point the topic is, the easier it is to find.

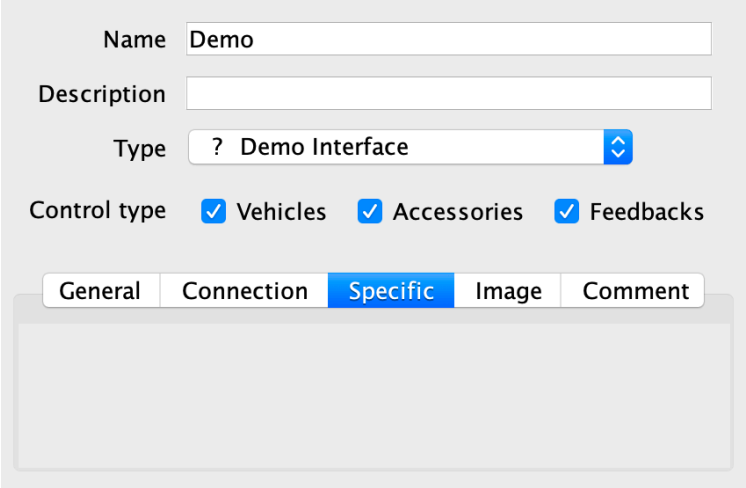
The last section 'Enhancement request' is for more experienced users who know what is possible in iTrain, but still lack some functionality or have some suggestions how to improve it. There is no guarantee that your suggestion will be included in iTrain within a certain time-frame, but all will be read and if possible discussed to get clear how it should be changed. The best ones will be considered to be implemented in the next or a later release.

In some cases you would like to exchange some details with another user that should not be visible to others. In that case send a PM (=Private message) to another user. Never put your address, phone or e-mail address in a topic if you want contact with other users. Just ask for contact and interested people can respond with a PM to exchange contact details.

Please be polite, patient and forgiving to other members both in asking and answering. We want to welcome new members, but also appreciate members who are spending a lot of time answering your questions and building this iTrain community. In case something happens that is not correct or really annoys you, please send an e-mail to forum@berros.eu so we can intervene or solve it.

Appendix D: Interface specifics

Next to the settings that are available for all interfaces, every interface has its own ‘Specific’ settings on the third tab of the ‘Interface Editor’. Some interfaces may have an empty page.



If the control of the interface has been restricted by changing the checks for ‘Control type’, then some options may not be visible, because they do not apply in this case.

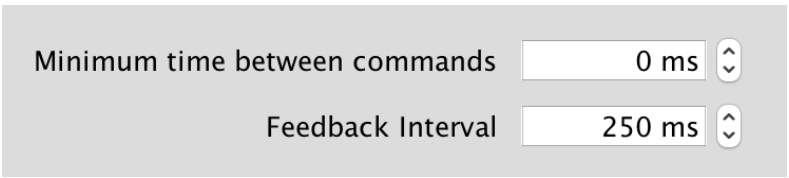
Demo

The ‘Demo’ interface is a dummy interface to be used for testing when you have no command station. It will treat every decoder as if it were connected to a system that supports all its possibilities. For example, an MFX decoder will have all steps and all functions are supported.

Märklin 6051

This is an interface protocol supported by many older systems and known as the P50 protocol. It was originally designed for the Märklin Interface 6050/6051 that can be used with the Märklin Control Unit 6021.

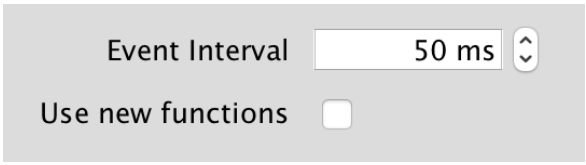
In some case it is necessary to specify a minimum time between sending two commands in case the flow control is not optimal and some messages get lost. In that case common values are in the range 20-50 ms.



The interface needs to regularly read all the feedbacks to see if they have changed (polling). In the field ‘Feedback Interval’ you can specify the time between reading the feedbacks. It is good to keep the value low (<= 250 ms), but when reading too many S88 modules with a small interval there will not be enough time to read them all before a new read is necessary. In such a case a new command station or adding a HSI-S88 is a good option to get quick feedback response times.

P50X

The P50X protocol, developed originally for the Uhlenbrock Intellibox, is an eXtended version of the P50 protocol (used by the Märklin Interface 6050/6051). But in fact it is much more than the very limited P50 protocol.



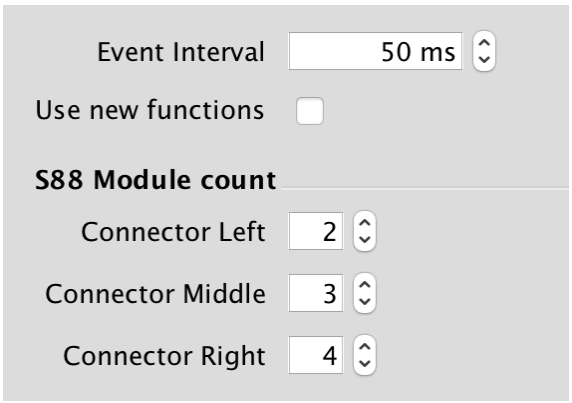
The P50X protocol will not inform iTrain of changes, but iTrain has to regularly ask if something has changed and what. This is called event polling and in the field 'Event Interval' in milliseconds you can specify the time between two requests for changes. A good default value is 100 ms, because if nothing has changed it will not exchange much data. With higher baudrates it can be made shorter as the overhead of the request is less.

The P50X protocol by default supports loc functions f0-f8. In the protocol there is an extension to also support f9-f16. If a command station supporting the P50X protocol is used that has implemented this extension, you can check 'Use new functions' to be able to use f9-f16, but by default leave it off to prevent errors in the communication.

TAMS MasterControl / RedBox
The TAMS MasterControl (or RedBox) tries to emulate the Intellibox as much as possible on the communication level, but it is actually another device. Some commands have been added that are only supported when choosing 'TAMS MasterControl/RedBox' explicitly.

TAMS already supports the extension for functions f9-f16, but with the latest firmware it will support an extra command for functions f17-f28. The option 'Use new functions' is now used to select between support for f0-f16 (off) and f0-f28 (on).

OpenDCC (P50X)
The OpenDCC command station is an open source command station that is available with two protocols: P50X and Lenz XpressNet. This interface supports the P50X protocol with the same extensions (functions f0-f28) as with TAMS Master Control. Support for the new Bi-Directional-Bus protocol (BiDiB.org) has been implemented in a separate interface.



In most cases, iTrain will determine how many feedback modules must be read. Because OpenDCC supports three strings of S88 modules, it wants to know how many S88 modules are connected to every connector (left, middle, right), so that it can calculate the correct module number for every S88 module.

Raptor
The Raptor is much more than a standard command station, but it can be used as a normal command station via a module that supports the P50XR protocol (based on P50X with some Raptor additions).

Raptor supports the extension for functions f9-f16 and also the extension for f17-f28. The option ‘Use new functions’ is now used to select between support for f0-16 (off) and f0-f28 (on).

MRdirect
MRdirect is software (and extra hardware) to create a command station out of an old DOS-PC, from Marco Roede (NL). It uses the P50X protocol and since version 8.0 MRdirect is compatible with iTrain. It supports the extension for extra functions f9-f16 without any setting.

Twin Center
The Twin Center is very similar to the Intellibox, but it is not the same. To be able to use the Twin Center with iTrain it is necessary to check ‘Basic settings’ of the ‘Interface’. The ‘Syntax’ must be set to ‘IB’ and not ‘TC’. It is also recommended to change the baudrate (‘Bit per second’) to 9600 baud. After this change the Twin Center will act like an Intellibox regarding computer control.

Intellibox (P50X)
Only the Intellibox (IB) models with a real RS232 serial port support the P50X protocol. The last model that supports it is the Intellibox IR. Newer models with a USB connector only support the LocoNet® protocol.

For the Intellibox it is recommended to check ‘Basic settings’ of the ‘Interface’ via the menu with respect to ‘Syntax’ and ‘Bit per Second’. The option ‘6051 + IB’ or ‘only IB’ must be chosen. For the baudrate (‘Bit per second’) 9600 or 19200 is recommended.

Note: If your Intellibox also supports the LocoNet® protocol (see ‘Basic settings’ -> ‘Syntax’) it is a good choice to use the LocoNet® protocol, because it is a better protocol without polling. In that case change the ‘Syntax’ to ‘LocoNet’ on the IB, and in iTrain select ‘LocoNet®’.

LocoNet® [78]

Intellibox USB (LocoNet®)
All new Intellibox (IB) systems (II, Basic and COM) have a USB connector and only support the LocoNet® protocol (not the old P50X anymore). The only difference between this Interface setting and the LocoNet® interface is that all the settings for 'Serial' and 'Specific' are automatically set to correct default values for an IB with USB so that you don't have to change anything. For details about the options read the section about LocoNet®.

DR5000 USB (LocoNet®)
The Digikeijs DR5000 is a command station that acts like chameleon. It can behave like an XpressNet and as a LocoNet® command station, both via USB and via a network connection. All these four interfaces are already available in iTrain, but for convenience we have added the most commonly used one via USB with LocoNet®.

The only difference between this Interface setting and the LocoNet® interface is that all the settings for 'Serial' and 'Specific' are automatically set to correct default values for a DR5000 with USB, so that you don't have to change anything. For details about the options, read the section about LocoNet®.

LocoNet®
The LocoNet® interface implements the LocoNet® protocol and can be used with most LocoNet® interfaces (for example a LocoBuffer, an Intellibox, or an Uhlenbrock LocoNet® interface).

NB: This interface is only available with a trial or official license (so not without a license), because it is used under license from Digitrax, Inc.

The first option, 'Virtual', is a special option for manually controlling vehicles and/or accessories on other interfaces if the box 'Control type' on this interface is not checked for this type. LocoNet handhelds or throttles can be used to control another system[79] that is not LocoNet®-compatible while iTrain is responsible for the synchronization. It is not necessary to have a command station on the LocoNet to control vehicles, as iTrain will act as a slot manager in that situation.[80]

[78] *LocoNet® is property of Digitrax, Inc. and is used under license.*

[79] Even analog locomotives and accessories without an address can be controlled via a so called 'pseudo address'.

[80] A stand-alone LocoNet with its own power supply, but without a command station, is then sufficient.

In the older LocoNet® specification, loc functions f0-f12 are supported. In the new specifications, f13 and up (to f9999 or higher) are also supported via a set of new LocoNet® commands. If the option 'Use new functions' is used then it will be possible to switch f13 and higher with systems supporting that.

In the LocoNet® specifications there are two commands to switch an accessory: One without and one with an acknowledgement. In general it is preferred to use the command with an acknowledgement, but some devices (for example the Uhlenbrock 63410) will not accept this command and accessories will not be switched. To be fully compatible you can uncheck 'Switch acknowledgement'. By checking the box you can enter a starting address from which accessories will use the command with acknowledgment. Lower addresses will still use the command without an acknowledgment.

A RailCom detector can deliver the address of the decoder and the polarity to calculate the direction of the vehicle. Officially there is no space for the polarity bit in the LocoNet® protocol, but some vendors[81] use the highest bit of the feedback address to return this polarity bit (halving the available feedback address range).

LocoNet® informs iTrain when a feedback state changes, but the current state of the feedbacks cannot be read directly. (However, on the Intellibox iTrain reads it directly from the memory of the command station, but this is an exception.) Some feedback modules will report their whole state when a command for a specific accessory address is sent (it will inspect the bus for such a command). This is called the 'Feedback report address'. It is possible to specify a range of values so that if you use a lot of feedback modules not all will report at the same time. The maximum range is limited to 11 values, for example from 1010-1020 for DCC. A range from Start 0 to End 0 means: 'do not use it', which is also the default setting.

Note: When using an Intellibox with USB interface it is better to select the interface 'Intellibox USB (LocoNet®)' that has all options configured with default values for these devices. This 'LocoNet®' interface has good default settings when using it with a LocoBuffer with a serial port. In case you use it with an Intellibox with an RS232 serial port, you will have to change the 'Syntax' on your IB and select Stopbits '2 bits' on the Serial tab.

LocoNet® Multicast

This interface has been designed with the MGV101/GCA101 in mind. This is an ethernet based LocoNet® interface that can deliver a lot of current on the LocoNet for large layouts and works via UDP Multicast[82].

LocoNet® TCP/IP

The LocoBuffer is a common device to interface with a command station based on LocoNet®. Because handheld devices with WIFI are becoming more and more popular to control your layout, some people have added a wireless or wired interface (wireless via a wireless router) to the LocoBuffer. To support this a new interface has been added to send LocoNet® messages over the TCP/IP via a network interface.

---

[81] The Digikeijs DR5088RC supports this polarity bit.

[82] UDP Multicast lets multiple devices communicate with each other so it is not point to point, but also no broadcast.

LocoNet® Server

The LocoNet® Server is also LocoNet® over TCP/IP, but then in the format of LbServer[83], so with readable text. This interface has been added to communicate with the Digikeijs DR5000 command station over the network, but can be used for other purposes as well.

[83] LbServer has been described on http://loconetovertcp.sourceforge.net

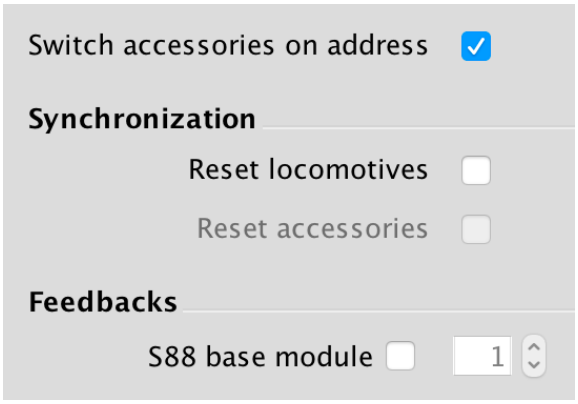
ESU

The ESU interfaces have the most sophisticated protocol to interface with a computer. The devices can synchronize their database with iTrain. By default they will try to merge both databases. That means that you only have to enter your locs and wagons (and optionally accessories) once in either iTrain or the ESU device. Objects entered in iTrain will be uploaded to the ESU device and vice versa.

Tip: Wagons in iTrain will be visible in the ESU device as locomotives with image numbers 54, 55 or 61. The question mark image with number 3 has been reserved for cars and should not be used for locomotives.

ECoS ESU

With the firmware version 3.0 and higher for the ECoS it is also possible to switch all the accessories directly by address instead of switching them via equal objects in the ECoS. In this case, check the option 'Switch accessories on address' (the accessories will not be synchronized or merged with the command station, because that is not necessary anymore).

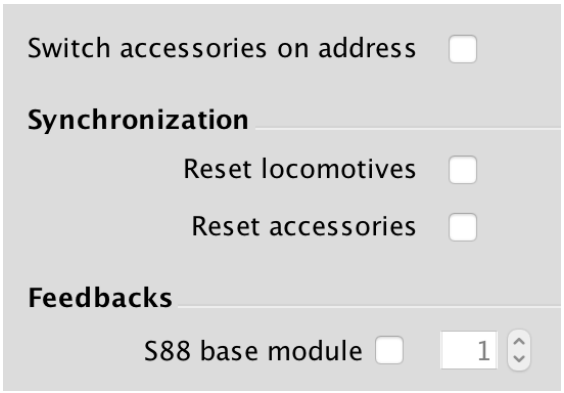


In some cases, when switching between different projects, it can be useful to remove objects on the ECoS or to prevent having multiple objects with the same address in the ECoS. In that case it is better to use the reset option that will remove objects not used by iTrain from the ECoS. This can be set separately for the locomotives and the accessories.

The ECoS can use both S88 feedbacks and ECoS-Detectors. The ECoS-Detectors have a fixed module number, but the address of an S88 module is based on the order in the string. To prevent that addresses will overlap if both types of feedback modules are used, it is possible to specify the number of the S88 base module. This is the number of the first S88 module.

Märklin Central Station 1

Märklin article no. 60212 has been built similar to the ECoS 1 by ESU. It is not supported anymore by Märklin, but in the past you could upgrade the hardware at ESU to a 'CS 1 Reloaded' or 'CS1R'. This made it possible to use firmware version 3.0 and higher so that it will act like an ECoS 1. In case you have upgraded you can better choose interface 'ECoS ESU' and change the description to 'CS1 Reloaded'.

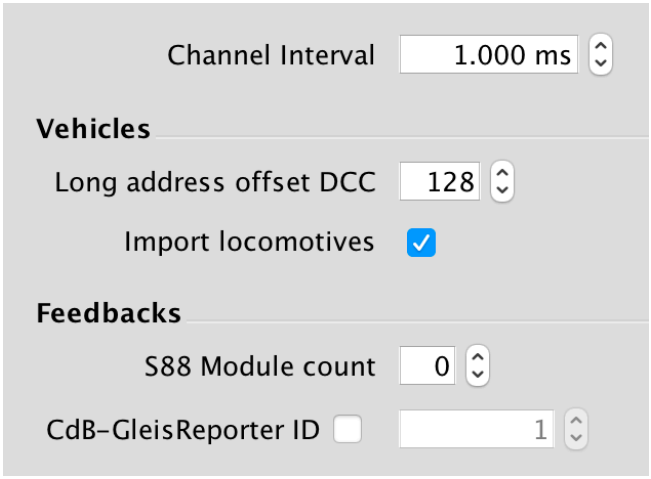


If you still have the original device, use this interface and do not use the 'Switch accessories on address' option, because it has not been fully implemented in the firmware. The other options work the same as on the ECoS interface.

Note: This interface requires firmware version 2.0.3 or 2.0.4 to be installed on the Central Station 1. Older firmware versions are not supported.

Märklin Central Station 2/3

The Märklin Central station 2 and Central Station 3 (CS2 and CS3) have no relation with the Central Station 1 (CS1) and use a completely different protocol based on the CAN-bus.



The interface continuously reads status data (also called channels) from the CS2/3, such as the voltage, current and temperature of all boosters (including the built-in one). In 'Channel Interval' you can specify how often this happens.

For DCC locs you can specify whether the long or short address will be used based on an offset. All locs with addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

Depending on the setting 'Import locomotives', iTrain will try to import the loc database from the CS2/3 when going 'Online' and tries to merge the data with the data in iTrain. However, it cannot upload locs to the CS2/3, but in general that is no problem since non-MFX locs can also be controlled without an entry in the CS2/3.

The CS2 and CS3+ have the possibility to directly connect S88 modules at the bottom of the device. To distinguish these S88 modules from other feedbacks modules directly connected to the CAN bus, you have to specify the number of S88 modules that are attached to the device in the field 'S88 Module count'.

The S88 connector on the CS2 works, but has one drawback, because it is not possible to directly specify the number of feedback modules connected to the device itself and it may not read them all. A better solution for large layouts is to use the new LinkS88 module or the CAN-digital-Bahn feedback modules directly on the CAN bus.

Tip: A workaround to specify the number of S88 modules on the CS2 is to create a 'memory' entry in the CS2 with an S88 contact number one higher than the highest one you use. This way the CS2 will at least scan all the modules that are needed in iTrain.

The S88 connector on the CS3 is only available on the Plus model, but does not have this drawback. The number of feedback modules can be specified via the settings of the GFP3.

The 'CdB-Gleisreporter ID' only applies when using the CAN-digital-Bahn feedback modules directly on the CAN bus. You can here specify the ID of these modules so they will be recognized and some address space will be allocated for them.

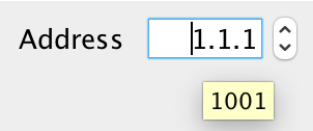
Link S88

The LinkS88 (60883) is a device that is connected directly to the CAN bus of the CS2 or CS3. It offers 16 built-in feedback contacts and three connectors for three S88-busses. Bus 1 and 2 have an S88N-connector (RJ45) and bus 3 has the original S88-connector.[84]

All the inputs on the device itself and the three busses must be put into one address space. Within iTrain one thousand addresses have been reserved for every bus and the inputs on the LinkS88 itself is seen as bus 0. In case of a single LinkS88 attached to the CS2/3 this leads to the following (assuming a maximum of 31 modules per bus) address assignment:

Bus	Address	Grouped	Description
0	1-16	1.1 - 1.16	Track inputs on LinkS88 itself
1	1001-1496	1.1.1 - 1.31.16	Module 1-31 attached to left side, S88N
2	2001-2496	2.1.1 - 2.31.16	Module 1-31 attached to right side, S88N
3	3001-3496	3.1.1 - 3.31.16	Module 1-31 attached to middle, S88
x	x000-x496	x.1.1 - x.31.6	Modules on additional busses.

Instead of the absolute address, the grouped address will by default be shown in iTrain, in which the bus, module and input number are separated by dots. The absolute address is available as a tooltip.



If more S88 modules on the bottom connector or other LinkS88 devices are connected, the bus number will count on after 3, so every LinkS88 takes up 4 additional busses (so 4-7 for the second).

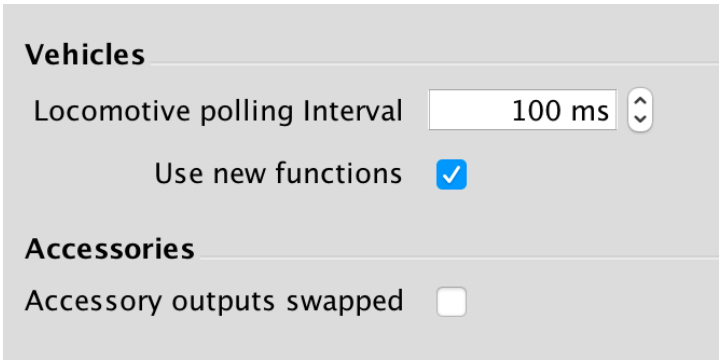
[84] The LinkS88 has almost the same functionality as the LDT HSI-88 or the μCon-S88-Master, but it is a dedicated solution for the CS2/3 and you don't need an additional interface in iTrain.

XpressNet

This section contains a list of all interfaces that are based on the XpressNet protocol.

Lenz XpressNet/X-Bus (Serial / USB / IP)
Many interfaces are based on the XpressNet protocol from Lenz. Originally the system was called X-Bus, but since version 3 it is called XpressNet. The old locomotive commands used in X-Bus version 1 and 2 are also supported by this interface.

Lenz itself delivers an interface with a serial port, one with a USB interface and one with an Ethernet or LAN interface, here called IP (=Internet Protocol) interface. In general the protocol is the same, but the USB and IP versions prefix the data with two extra bytes, so choosing the right interface is important.



The Lenz interfaces have the option 'Locomotive polling Interval' in milliseconds. XpressNet informs iTrain about many changes, but not of the state of handhelds that are controlling locs. However it does inform iTrain when a train is taken over by a handheld. iTrain will regularly read the locomotive information when a loc is taken over by a handheld, to know the speed and functions. A good default value is 100 ms.

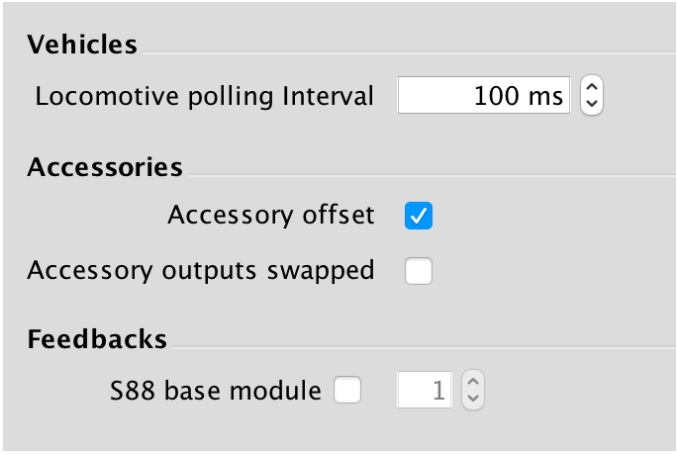
In the new XpressNet protocol version 3.6, support for extra functions f13-f28 has been added. In case you use this protocol version, you can check 'Use new functions' to be able to control them as well, but in case you don't know, just leave it off to prevent problems in the communication.

There is some discussion about how the two accessory outputs should be interpreted among users of XpressNet. In case all your turnouts switch differently from what you would expect, you can use the last option to correct that at once.

S88 XpressNet LI
The Roco Multimaus is very popular in start-sets, but it is not possible to connect it to a computer directly. There are some devices available on the internet that create an interface based on XPressNet via a serial port, and at the same time deliver support for connecting S88 modules and deliver their state via XpressNet. In the Netherlands the most well known device is the 'S88XpressNetLI'.

Next to the 'Locomotive polling Interval' that is described with Lenz XpressNet, there is another option that has to do with an issue with the 'Multimaus' concerning accessory addresses. Accessory decoders programmed with another DCC system at address 1 will have address 5 in the Multimaus. Even worse, decoders programmed at address 1-4 with a Multimaus may not even be visible with another DCC system. If you use a Multimaus, always program your accessories at address 5 or higher. In case you want this Multimaus address 5 to appear as 1 in iTrain, check the option 'Accessory offset'. This way it is easier

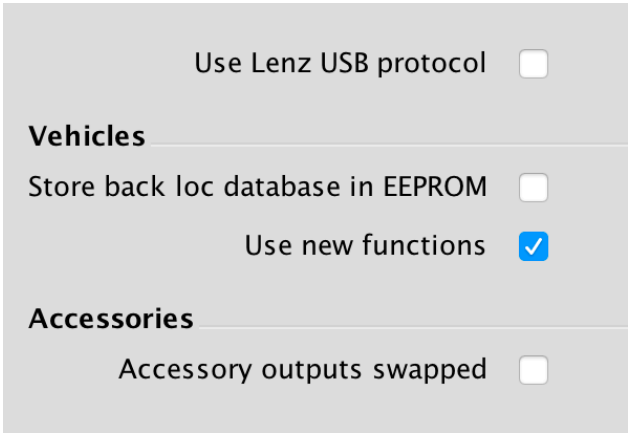
to later migrate the project to another DCC command station when you outgrow the Multimaus.



The option 'S88 base module' allows you to set an offset in the feedback address range for the S88 modules. It is configurable to prevent overlap with possible turnout feedback devices that have a fixed address space. Allowed values are 0-64. Zero means no S88 feedback. Value one means that the first S88 module uses feedback address 1-16 and so on.

MoBaSbS

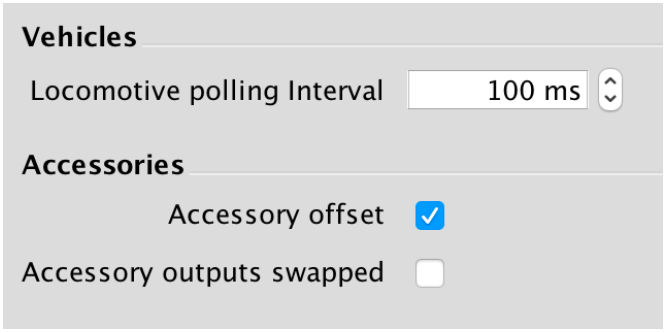
The 'ModellBahnSelbstbauSteuerung' or MoBaSbS is a system based on the XpressNet protocol and has some popularity in Germany. It works with both DCC and Motorola and also supports synchronizing the loc database in iTrain with the command station. In addition, it has extra commands to inform iTrain of changes in the loc speed, and therefore does not need polling.



It supports both the Lenz serial and Lenz USB protocol so you have to set the settings the same as the dipswitches on the MoBaSbS. The loc data of the MoBaSbS and iTrain are only synchronized when going 'Online'. New loc data in iTrain is uploaded to the RAM of the MoBaSbS. To store back the loc database in EEPROM when going offline, select this option. The 'Use new functions' is provided in case the MoBaSbS will support the f13-f28 in the future, but currently it should be left unchecked. The last option is to swap the interpretation of the accessory outputs in case turnouts and signal switch in the opposite way.

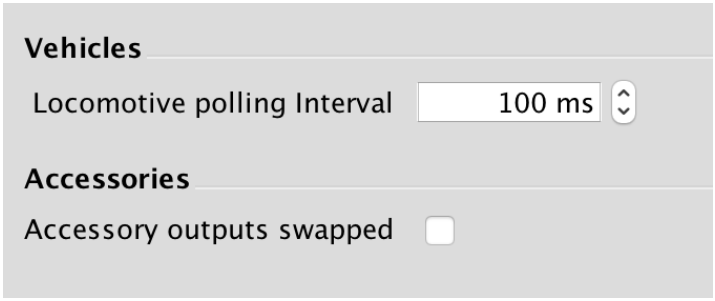
ZF5

The 'ZF5' is a system from CT-Elektronik based on the XpressNet protocol, but with some important changes that are covered in this interface. It only support vehicles and accessories.



Rocomotion

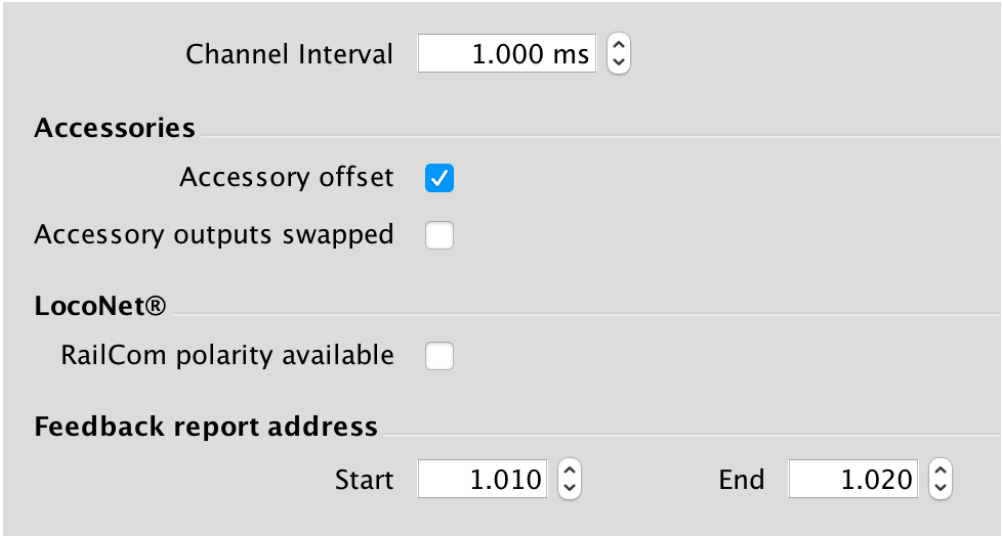
Rocomotion is a proprietary protocol used by Roco based on XpressNet, but with some additions for their own feedback modules (10787). This interface only supports the older 10785 device (with a serial port).



The blue command station 10832 with a USB port is not supported via the USB and Rocomotion and we recommend to buy an 'S88 XpressNet LI' to be able to use iTrain with this system.

Roco Z21

The Roco Z21 is a system based on the XpressNet protocol, but with some adjustments for more efficiency and new commands specific for this device.



The interface continuously reads status data (also called channels) from the Z21, such as the voltage, current and temperature of the internal booster. In ‘Channel Interval’ you can specify how often this happens.

There is another option that has to do with an issue with the Z21 concerning DCC accessory addresses. Accessory decoders programmed with another DCC system at address 1 will have address 5 in the Z21. Even worse, decoders programmed at address 1-4 with a Z21 may not even be visible with another DCC system. If you use a Z21, always program your accessories at address 5 or higher. In case you want this Z21 address 5 to appear as 1 in iTrain, check the option ‘Accessory offset’.

Tip: The Z21 supports the Roco feedback modules 10787, but it is also possible to use other feedback systems via the LocoNet® bus on the Z21 in iTrain. An affordable solution is to connect an s88 to LocoNet® adapter (s88LN) to this bus and use the well known s88N modules.

The Z21 has a LocoNet® bus and also has some options that have been described in the LocoNet® section such as ‘RailCom polarity available’ and ‘Feedback report address’.

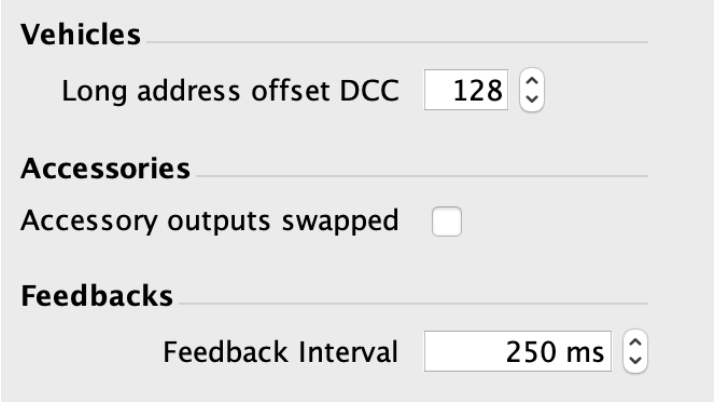
Note: The white z21 (with lowercase z) does not have a LocoNet-Bus and misses some other features compared to the black Z21, but it can be used as well by iTrain via the ‘Roco Z21’ interface.

Hornby

Hornby is another interface based on XpressNet for the Hornby command stations, with the settings on the Serial tab adapted for this system. It is not officially supported, because I have not been able to test it myself. In case you are interested, contact us.

NCE Power Cab/Pro

An interface for the NCE has been added, because of many requests from users from the UK, but it is not ideal for computer control. iTrain does not receive the changes on the throttle via the computer interface, and so you can control it either manually or completely automatically via iTrain but not combined. This interface should still be considered experimental for the time being.



For DCC locs you can specify whether the long or short address will be used based on an offset. All locs with addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

The second option is to swap the interpretation of the accessory outputs in case turnouts and signal switch in the opposite way.

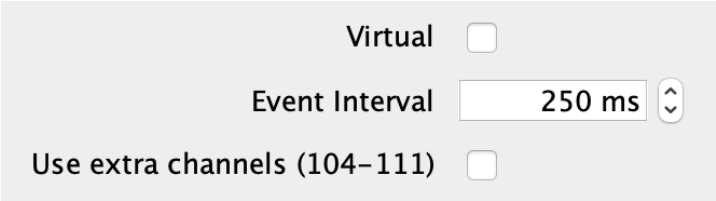
The last option, ‘Feedback interval’, can be used to specify how often the feedbacks will be read as this can only be achieved by polling.

Selectrix

This section contains a list of all interfaces that support the Selectrix SX-bus.

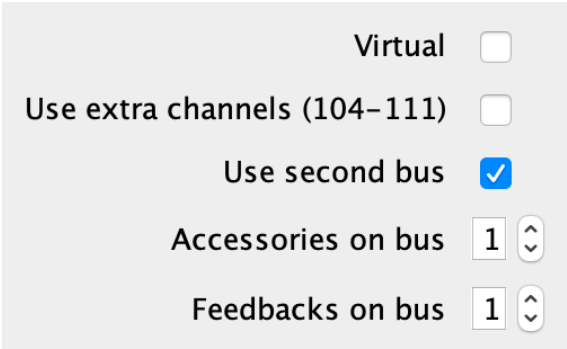
Some of the interfaces have the option 'Virtual' for manually controlling vehicles and/or accessories on other interfaces if the box 'Control type' on this interface is not checked for this type. Selectrix handhelds or throttles can be used to control another system[85] that is not Selectrix-compatible while iTrain is responsible for the synchronization.

Selectrix
This interface is the original Selectrix protocol with one SX-bus. It will read out the complete SX-bus regularly to inform iTrain of changes on the bus.



The time between reads is called the 'Event Interval'. Because this process is quite slow, you have to use 250 ms when using a baudrate of 9600. With higher baudrates you can make the interval smaller and the reaction time will be better. Today there are new, faster protocols based on a Selectrix bus available that are also supported by iTrain.

Rautenhaus SLX
The Rautenhaus SLX interface can be used for both the SLX825 and the SLX852. The SLX825 only supports one bus, while the SLX852 supports two buses and can also be used as an interface without a command station to only switch accessories and read feedbacks.

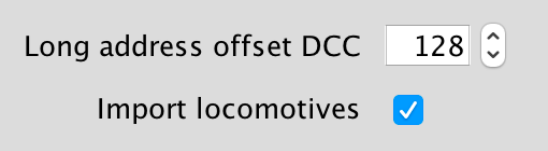


Rautenhaus uses an efficient event mechanism that notifies changes on the bus so that no polling is necessary. You only have to configure if you want to use the extra channels (104-111), and in case you use a second bus you have to specify to which bus the accessories and feedbacks are connected.

Rautenhaus RMX
The Rautenhaus RMX interface can be used for the command station RMX950USB or the separate interface RMX952. It automatically supports two buses where the first RMX bus (0) is for locomotives and the second SX bus (1) is for the accessories and feedbacks.

---

[85] Even analog locomotives and accessories without an address can be controlled via a so called 'pseudo address' via Selectrix.

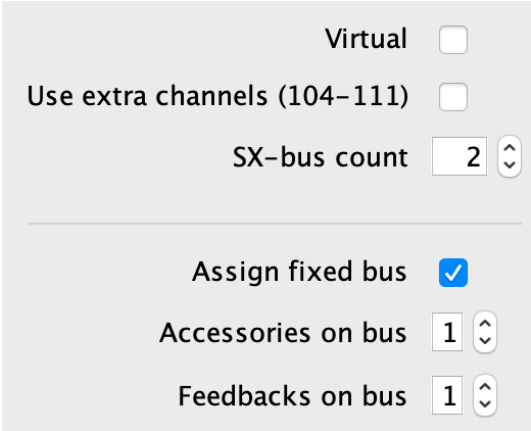


The loc database of iTrain and the RMX950 are synchronized when the interface is going 'Online' if the 'Import locomotives' option has been selected. Selectrix, Selectrix2 and DCC locs are supported on this interface.

For DCC locs you can specify whether the long or short address will be used based on an offset. All locs with addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

Rautenhaus RMXnet
The Rautenhaus RMXnet interface has the same settings as the normal RMX interface, but it uses a network connection to connect to the software 'RMX-PC-Zentrale 2.0' instead of connecting directly to the RMX command station itself.

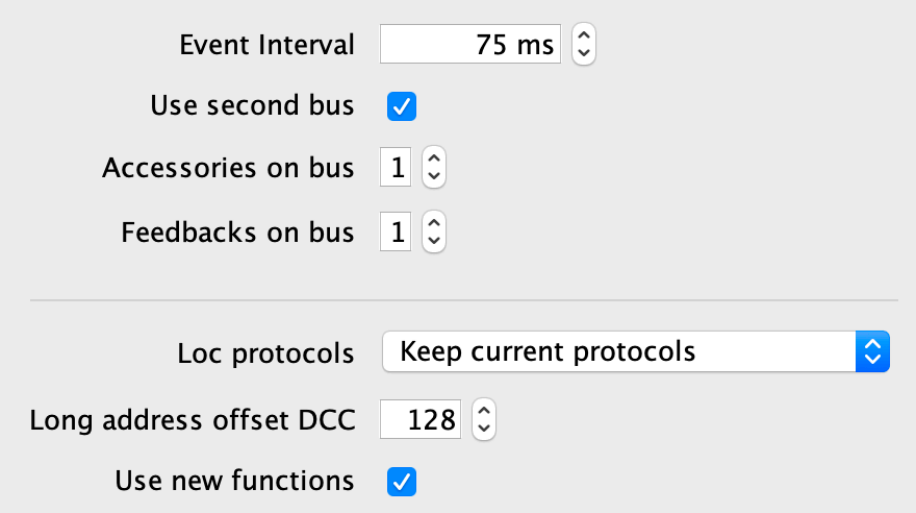
Müt 2004
The Müt 2004 is a pure Selectrix command station with some extensions in the protocol to support multiple SX-buses. It also uses an efficient event mechanism that notifies changes on the bus so that no polling is necessary.



The first option specifies if you want to use the extra channels (104-111) so you can use more addresses. In case of doubt leave it unchecked.

The other options concern how you divide your accessories and feedbacks over the available SX-buses. First you have to specify the number of buses used. Theoretically this number can be higher than two, but in practice you can select between 1 or 2. If you use more than one bus you can choose to assign a fixed bus to all accessories and feedbacks and use addresses in the range 1-103/111 as with all Selectrix systems. The other option is not to use this and select the bus of every accessory or feedback individually via the address by adding 1000 to the address when it is on bus SX1 (and 2000 for SX2 and so on). For SX0 you just use the normal address in the rang 1-103/111.

FCC (Doehler & Haass / MTTM)
The Future-Central-Control is a command station based on two Selectrix buses that next to Selectrix also supports Selectrix 2, DCC and Motorola locs. It is still necessary to read all the SX-bus data regularly, but because it uses a very high baudrate, the delays are minimal. A typical 'Event Interval' can be 50-100 ms.



It is possible to use a second bus for accessories and/or feedbacks. In case you use a second bus, you have to specify to which bus the accessories and feedbacks are connected.

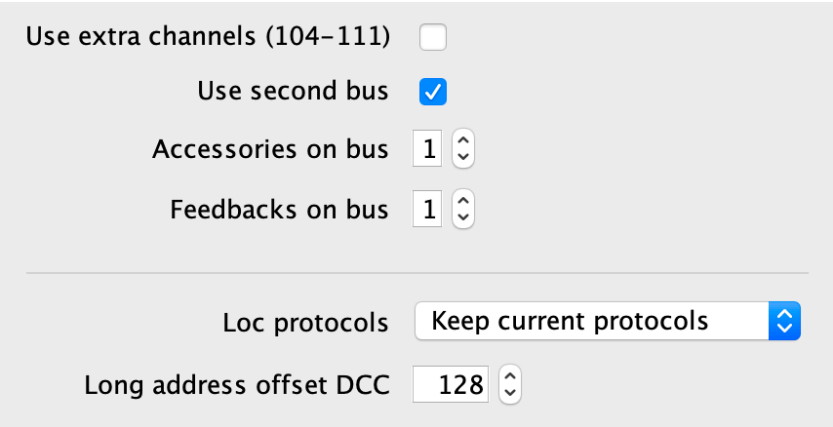
The FCC is a multi protocol command station, but you have to configure the command station with the protocols that will be used. To change it you can use the box 'Loc protocols'. Only in case the value is not 'Keep current protocols', the setting will be applied to the command station when going 'Online'. This setting is not stored in iTrain sessions. To change it, select it here and go online and the used protocols in the command station will be changed and stored in the FCC. The next time you start iTrain the value in the box will be 'Keep current protocols' again.

For DCC locs you can specify whether the long or short address will be used based on an offset. All addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

Starting with firmware 1.13 there is support for up to 32 functions with SX2 and DCC instead of the maximum of 16 functions in older versions. By checking the box 'Use new functions', iTrain will use the commands to support these extra functions.

Stärz ZS1
The Stärz ZS1 communicates identically to the Rautenhaus SLX that has been described before and has the same options.

Stärz ZS2
The Stärz ZS2 is a multi-protocol command station (you can update the ZS1 to a ZS2) that supports up to 16 locs of type Selectrix 2 or DCC in addition to all the Selectrix features of the ZS1.



You can configure the protocols used on the track via the command station, but you can also can use the box ‘Loc protocols’ here. Only in case the value is not ‘Keep current protocols’, the setting will be applied to the command station when going ‘Online’. This setting is not stored in iTrain sessions. To change it, select it here and go online and the used protocols in the command station will be changed and stored in the ZS2. The next time you start iTrain the value in the box will be ‘Keep current protocols’ again.

For DCC locs you can specify whether the long or short address will be used based on an offset. All addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

Stärz ZS2+
The Stärz ZS2+ supports up to 32 locs of type Selectrix 2 or DCC, but uses a completely different protocol from the ZS2. In iTrain you have to select the interface ‘FCC’ for this command station, but leave the box ‘Use new functions’ unchecked as the ZS2+ does not support these extra functions.

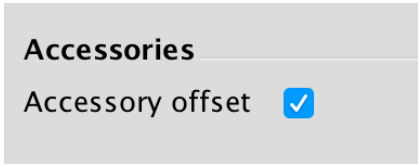
Zimo

Two different protocols from Zimo are supported:

MX1
The 'Zimo MX1' interface has been developed for the MX1 command station with firmware version 3.06 only. Only locomotives and accessories are tested on this interface and will be officially supported, although the feedbacks have been implemented. No specific configuration is necessary.

Extra information about the voltage and current of the command station is displayed on the status bar.

MX10
The MX10 uses the Zimo CAN protocol 2.0 and can be connected via Ethernet (=Network) or via USB (=Serial). Both types are supported in iTrain.



The DCC standard for accessories has some ambiguity. Therefore basic accessory address 1 in iTrain could be mapped to two different numbers with an offset of 4. Most vendors use the option that resembles the choice when 'Accessory offset' is checked and the first four addresses in the DCC address space are not used. If you want to use these unofficial addresses then uncheck this option, but take care that the accessory address space shifts with 4 in that case.

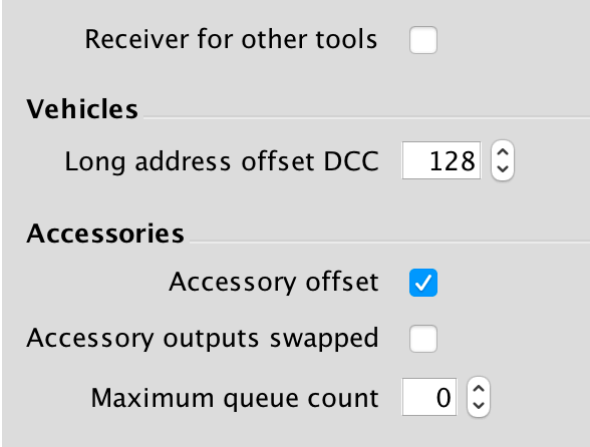
Massoth

The Massoth interface has been developed for the DiMAX command stations with a firmware version 2.59 or higher. No specific configuration is necessary.

Information about the current and its limit will be visible on the status bar. If the protocol of the loc in iTrain conflicts with the one in the Massoth then iTrain will change it in the Massoth.

BiDiB

The BiDirectional Bus (BiDiB) interface can be used to control any command station that follows the BiDiB standard as defined on http://bidib.org. In case the interface is not listed as an interface on your computer, just ask for a new license with BiDiB support by e-mail for free.



iTrain will read all the options from the features defined in the nodes. Only a few DCC options can be specified here.

This interface can function as a receiver for other tools[86] like the BiDiB Wizard or Monitor via a network socket. When this option is active these tools can connect with an online BiDiB interface and configure accessories while iTrain is controlling the layout. In these tools you have to select 'Serial over TCP' and specify the IP address of the computer running iTrain (or use `localhost` if it is on the same computer).

For vehicles (locs) you can specify whether the long or short address will be used based on an offset. All addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

The DCC standard for accessories has some ambiguity. Therefore basic accessory address 1 in iTrain could be mapped to two different numbers with an offset of 4. Most vendors use the option that resembles the choice when 'Accessory offset' is checked and the first four addresses in the DCC address space are not used. If you want to use these unofficial addresses then uncheck this option, but take care that the accessory address space shifts with 4 in that case.

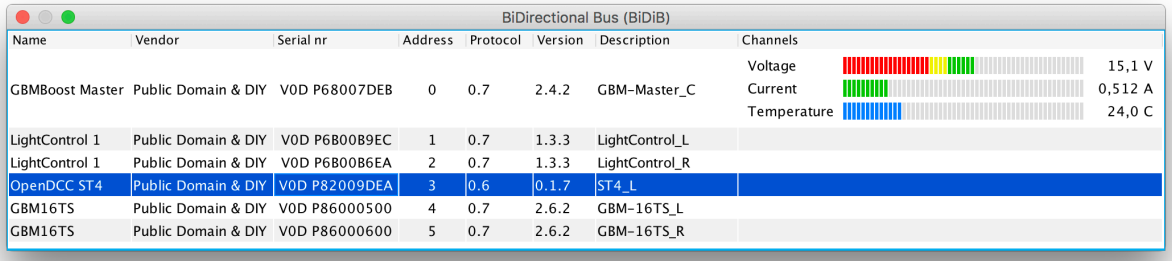
By default the commands are sent to the non-DCC accessories directly, even if new ones arrive while previous commands have not finished yet. In case every command absorbs a lot of power, this might give a peak current. By specifying a 'Maximum queue count' greater than 0, queues will be created and commands will be executed in order per queue, but commands for different queues will be executed in parallel. The module number modulo the maximum queue count will determine the queue number. So if the queue count is equal to the number of BiDiB modules, every module will work in parallel, but commands sent to the same module will be executed sequentially.

The main difference between BiDiB and other interfaces is that BiDiB tries to avoid addresses as much as possible. So all accessory and feedback ports on BiDiB nodes can

[86] The required versions for the BiDiB-Wizard is 1.10 and for the BiDiB-Monitor 1.0.

be specified by selecting the node from a dropdown list and entering the local port on the node instead of entering a more global address for the interface.

Having no addresses for feedbacks would make the Feedback Monitor useless. Therefore iTrain assigns global addresses to feedbacks based on the order of modules and the port number. This address may vary per session and is only there to have some visual feedback in the Feedback Monitor, especially when no feedbacks have been defined in iTrain yet. The addresses will not be displayed in the feedback editor.

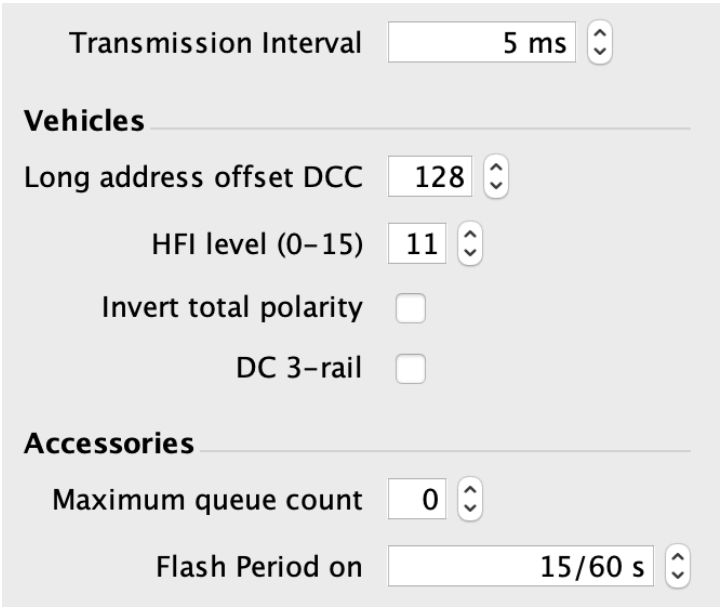


The list with available modules with extra information can be shown by clicking on the ‘Info’ icon of the BiDiB interface. Pressing the ‘Identify’ button on a BiDiB node will select the node in the list.

VPEB / DinaSys

This section contains a list of all interfaces that are a result of the development of the Dinamo system to run analogue and digital locs on one layout by controlling every block individually. Except for the 'Dinamo' interface, all other interfaces are useful as a second interface next to any other non-Dinamo system as well.

Dinamo (RM-x)
The Dinamo interface works with firmware version 3.0 or higher on the RM-U or RM-C and can be connected to either the serial or the USB port depending on what is available.



The 'Transmission Interval' is configurable, but in general keep it on the default value of 5 ms.

For DCC locs you can specify whether the long or short address will be used based on an offset. All addresses equal to or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

The 'HFI level' is the level for the front light of an analog loc. It is recommended to keep it below 12 so that direction changes are detected immediately.

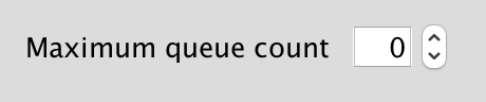
The 'Invert total polarity' option considers the polarity of all blocks to be the opposite. This is to prevent that most blocks need to have an 'Inverted' polarity and only some a 'Standard' polarity. If the polarity setting for all blocks looks correct and your analog loc is driving the wrong direction, then select this box.

When using 3-rail track the analog locs should be modified to run DC instead of AC and you have to select 'DC 3-rail' so that polarity will be set correctly for 3-rail.

The 'Maximum queue count' has been described in the OM32 interface section.

The 'Flash Period on' is a general setting that is currently not used by iTrain.

OM32
The OM32 interface works with both the OM32 and the OC32. The OM32 interface is normally used as an additional interface (lowercase s), because it can only control accessories.



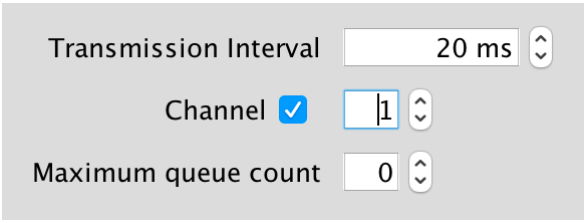
By default the commands are sent directly to the OM32, even if new ones arrive while previous commands have not finished yet. In case every command absorbs a lot of power, this might give a peak current. By specifying a 'Maximum queue count' greater than 0, queues will be created and commands will be executed in order per queue, but commands for different queues will be executed in parallel. The module number modulo the maximum queue count will determine the queue number. So if the queue count is equal to the number of OM32 modules, every module will work in parallel, but commands sent to the same module will be executed sequentially.

For turnouts one output with a high (1) or low voltage (0) is used that naturally fits with servos. Relays and decouplers also use one output. Signals can use multiple outputs to connect all the different light bulbs or LEDs.

NB: You can also select this interface when you have got an OC32. The OC32 can be controlled by either sending OM32 commands that are generated by iTrain, or by selecting the preprogrammed 'Aspects' in the OC32. You can choose this per accessory. When using the 'Aspects', the number of outputs used is determined by the definition in the OC32.

OC32

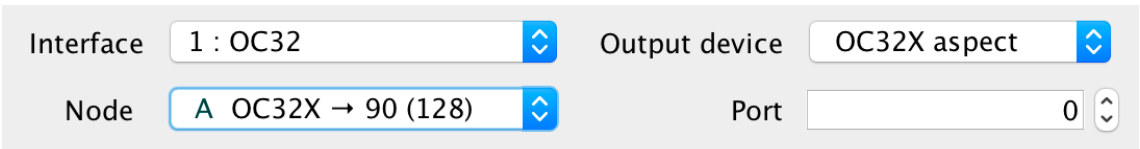
The OC32 interface only works with the OC32, but uses the same commands as the OM32. It uses bidirectional communication instead of the unidirectional communication of the OM32, so it can also read data or confirm requests. Only use this if all the devices on the USB-RS485 converter are OC32 devices.



The 'Transmission Interval' is configurable, but in general keep it on the default value of 20 ms.

The 'Channel' is only used for extended addressing[87] to indicate the channel on which iTrain will send messages to the modules. This channel reflects the DIP-switch settings on the OC32. The module number of the OC32 is equal to the extended address and should be programmed in the OC32 via its configuration tool. If you uncheck the channel, everything works in classic mode with the module number equal to the DIP-switch setting.

The 'Maximum queue count' has been described in the OM32 interface section.



[87] Extended addressing is only available with OC32 firmware version 3. It allows you to use a maximum of 96 modules on the bus via one channel instead of the original 16 modules. Using even more modules by using multiple channels at the same time is not considered useful as the bus would become too slow.

A new ‘Output device’ has been introduced, called ‘OC32X aspect’, that allows you to directly select the ‘Node’ or ‘Module’ from a list with names. In addition, you have to select the local port or pin on the module with values 0-127, allowing virtual ports for multiplexing.

The pins on the OC32 can be used as inputs. These inputs will be visible in iTrain as feedbacks. Just select the correct ‘Interface’ and ‘Node’ (=module) and the ‘Port’ with values 0-127, allowing virtual ports for multiplexing.

PM32/OM32

The PM32/OM32 is an experimental interface to use PM32, OM32 and OC32 devices separately from a Dinamo system on their own RS-485 network with the USB-RS485 converter. The PM32 is a device to switch 64 pulse driven devices like ordinary turnouts (no servos).

The ‘Transmission Interval’ is configurable, but in general keep it on the default value of 20 ms.

DTC

The DinaSys Turntable Control (DTC) is a complete solution for a turntable with the following features in combination with iTrain:

- The position of the bridge will be changed with smooth movements by slowly accelerating it at the start and decelerating the bridge near the end position.
- It will read the actual position of the bridge at any time via sensors even when you change it manually.
- It allows the sidings connected to the turntable to be switched off via built-in relays.[88]

If the option ‘Polarity offset’ is set greater than zero, the internal polarity relay of the DTC will be switched based on the position of the turntable. The offset indicates the first connection with normal polarity and divides the turntable in two halves. If there are 48 connections and the polarity offset is 11, then the bridge will have normal polarity at position 11-34 and inverted polarity at position 35-48 and 1-10. This option is not necessary for Dinamo users as the polarity will already be switched on the block output of the TM-H, so they should leave it zero.

Within the DTC interface two addresses have been reserved for special purposes. By defining an A/B relay with this interface and address 98, you can activate and release the emergency stop of the bridge. A relay with address 99 will show the state of the built-in polarity relay.

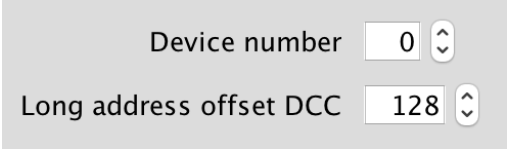
---

[88] You can share one TM-H or TM44 output for all sidings switched via relays in a Dinamo system.

Littfinski Daten Technik (LDT)

Digital-S-Inside 2 / DiCoStation
The DSI-2 interface is an interface developed for the DiCoStation. The DiCoStation is a device with an HSI-S88 USB built-in and it can be used as a command station by using the DSI software on a computer with the Windows operating system. This interface only supports version 2 of the DSI software that is XML based, and is not compatible with version 1 which is P50X protocol based.

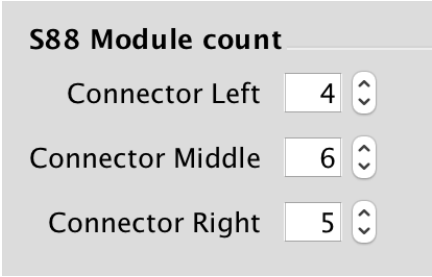
When the DSI software (`DsiService.exe`) runs on Windows it acts as an intermediate hub and iTrain connects over the network to the DsiService, so you have to specify a host name or IP address. Normally you will run iTrain and the `DsiService.exe` on the same computer, but is possible to run iTrain on another computer with for example macOS or Linux.



The device number is normally 0, but can be changed if you are using more than one DiCoStation at the same time.

For DCC locs you can specify whether the long or short address will be used based on an offset. All addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

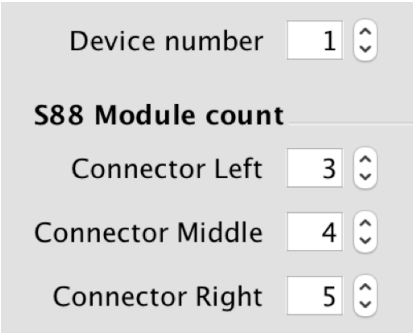
HSI-S88
The HSI-S88 interface only works with the serial version of the HSI-S88. The HSI-S88 is normally used as an additional interface, because it can only read feedbacks.



In most cases, iTrain will determine how many feedback modules must be read. Because the HSI-S88 supports three strings of S88 modules, it wants to know how many S88 modules are connected to every connector (left, middle, right), so that it can calculate the correct module number for every S88 module.

HSI-S88 USB
The HSI-S88 USB is not just an HSI-S88 with a built-in USB-serial adapter, but it is a different device that, with the DSI software, could act as a DiCoStation. The HSI-S88 USB only has drivers for Windows. So this interface option is not selectable in iTrain with other operating systems.[89]

[89] The HSI-S88 serial in combination with a USB-serial adapter is as cost effective and better interchangeable between operating systems.

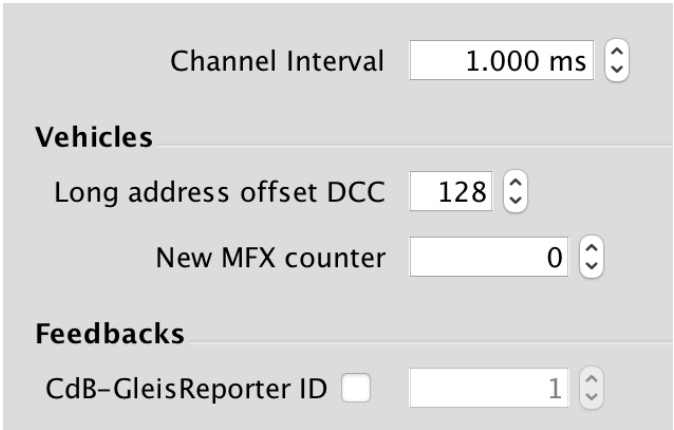


The HSI-S88 USB has the same ‘Module count’ settings as the HSI-S88 and an additional ‘Device number’ to identify the device in case you use more than one HSI-S88 USB.

CAN-digital-Bahn

The CAN-digital-Bahn project is a set of devices based on the CAN-bus developed by Thorsten Mumm.

CAN-Control-Schnitte
The 'CAN-Control-Schnitte' or shortly 'CC-Schnitte' is the connection between the Märklin Gleisbox, the computer and other modules of the CAN-Digital-Project all connected via a CAN bus. It is a full interface that supports the loc protocols DCC, Motorola and MFX, accessories in DCC and Motorola format and feedbacks such as the 'GleisReporter' or 'StromSniffer'.



The interface continuously reads status data (also called channels) from the command station such as the voltage, current and temperature of all boosters (including the built-in one). In 'Channel Interval' you can specify how often this happens.

For DCC locs you can specify whether the long or short address will be used based on an offset. All locs with addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

The registration of new MFX locomotives is done by iTrain. MFX locs will only announce themselves if they think they are new to the system. The decision of a loc to announce itself is based on an MFX counter value that it reads from the track. By increasing this value all locs that are not on the tracks when the interface goes online will announce themselves when they are put on the track.

In practice it means that this value should be set once to a value comparable to the number of MFX locs you have (with a minimum of 5) and it should only be increased when you delete locs from iTrain.

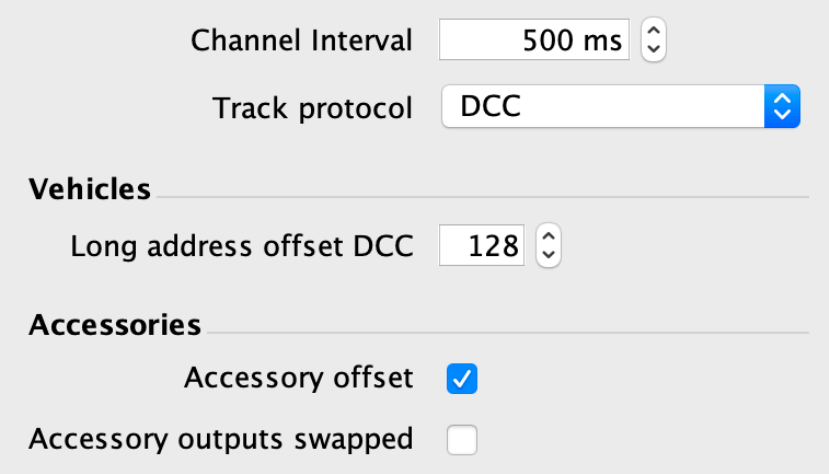
In the field 'CdB-Gleisreporter ID' you can specify the ID of the GleisReporter modules so they will be recognized and some address space will be allocated for them.

CAN-PC-Schnitte
The 'CAN-PC-Schnitte' interface is normally used as an additional interface, because it can only read feedbacks. It is similar in functionality to the HSI-S88, but using the reliable CAN bus in combination with feedback modules called 'GleisReporter' or 'StromSniffer'. It has a USB-connector and requires no specific configuration.

Lokstore Digital

Lokstore Digital was originally involved in the development of the µCon product line from LSDigital (see next section), but has improved and extended the features in the LoDi line of products. All devices have a network interface. Most of the configuration can be done with their own software 'LoDi-ProgrammerFX'.

LoDi-Rektor
The LoDi-Rektor is a command station combined with a booster management system. It supports both µCon and LoDi boosters. In addition it supports the µCon-RailSpeed and the LoDi-TrainSpeed for speed measurements.



The interface continuously reads status data (also called channels) via the Rektor, such as the voltage and current of all boosters. In 'Channel Interval' you can specify how often this happens.

You can select the allowed protocols via the option 'Track protocol'. DCC is fully supported including 'Extended Accessory' and 'Programming on Main' (PoM).

For DCC vehicles you can specify whether the long or short address will be used based on an offset. All vehicles with addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

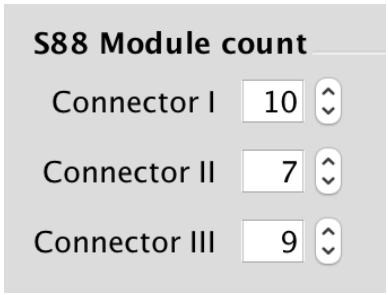
The DCC standard for accessories has some ambiguity. Therefore address 1 in iTrain could be mapped to two different numbers with an offset of 4. Most vendors use the option that resembles the choice when 'Accessory offset' is checked and the first four addresses in the DCC address space are not used. If you want to use these unofficial addresses then uncheck this option, but take care that the accessory address space shifts with 4 in that case.

LoDi-Shift-Commander
The LoDi-Shift-Commander is an additional interface only for accessories, and it needs no additional specific configuration in iTrain. It can be used to switch turnouts, signals and all kinds of lights. It receives the model clock changes from iTrain to simulate daylight changes via a 4-C-LED module.

LoDi-S88-Commander
The LoDi-S88-Commander is an additional interface only for feedbacks, and it needs no additional specific configuration in iTrain. It contains two connectors for S88N modules with support for S88.2 to process and send RailCom information back to iTrain.

LSDigital

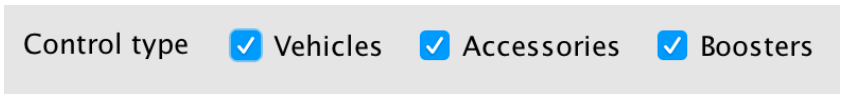
µCon-S88-Master
The µCon-S88-Master interface is a device to read in S88N modules from three strings, just like the HSI-S88, but it is a network interface and connects to the computer via Ethernet. It has configurable options to filter the input and remove some distortions, resulting in a clean output. It is normally used as an additional interface, because it can only read feedbacks.



Because the µCON-S88-Master supports three strings of S88N modules, it wants to know how many S88 modules are connected to every connector (I, II, II), so that it can calculate the correct module number for every S88 module.

µCon-Manager
The µCon-Manager in combination with the µCon-Boosters is a command station independent booster system. The µCon-Manager interface is normally used as an additional interface next to a command station, because it can only read boosters. In addition it also supports the µCon-RailSpeed on the same bus.

The interface continuously reads status data (also called channels) via the manager, such as the voltage and current of all boosters. In ‘Channel Interval’ you can specify how often this happens.



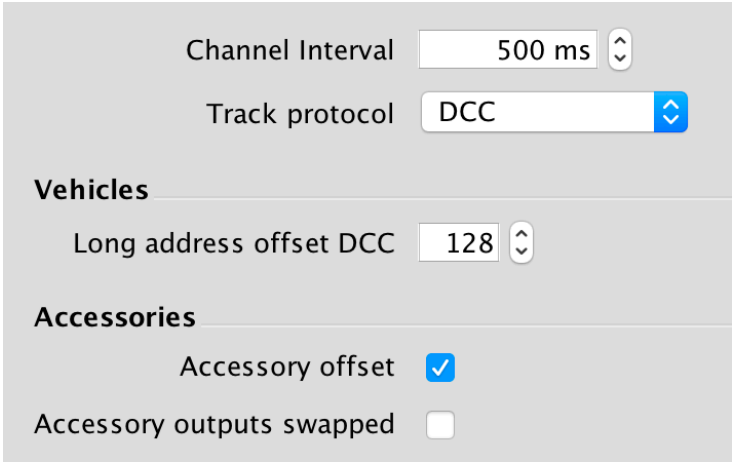
The same device can be upgraded with new firmware (at least version 3) to function as a generator for rail protocol signals as described in the next chapter. You have to select the control type ‘Vehicles’ and/or ‘Accessories’ to activate the generator function. In that case you should not attach a command station to the ‘CDE’ or ‘5-Pol’ connector of the device.

Generators

Most command station communicate with the computer via a proprietary or common interface protocol to receive their instructions. These instructions will be translated into commands on the track via protocols such as DCC, Motorola, Selectrix, etc. This is not a one to one translation, as track commands will be repeated often to guarantee that the locomotives will get their information even when some of the messages are skipped because of bad contact between the wheels and the track. This is called the locomotive refresh cycle. This is normally the responsibility of the command station.

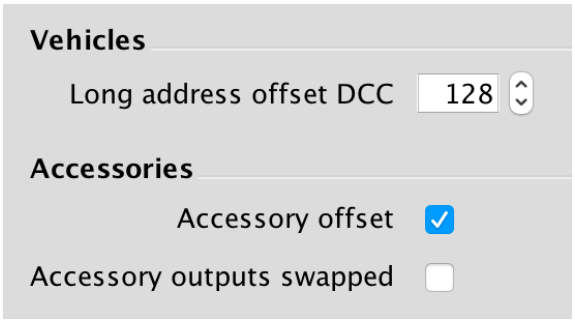
Some devices can only generate electrical signals related to for example the DCC standard. In this case the generation of the logical DCC signals including the refresh is the responsibility of iTrain and only the electrical conversion is done by the device.

µCon-Generator



The µCon-Generator is a µCon-Manager that has the option to generate either DCC or Motorola signals based on logical DCC or Motorola packets supplied by iTrain. You can select the protocol via the option 'Track protocol'. The interface supports both vehicles and accessories (including the extended DCC aspects for DCC). It will act as a normal µCon-Manager if only the control type 'Boosters' has been selected. Specific options for vehicles and accessories only apply to DCC and have been described at the end of this chapter.

SPROG



The SPROG is a device that will generate the electrical DCC signals based on logical DCC packets supplied by iTrain. The interface supports both vehicles (to control locomotives but also car systems based on DCC) and DCC accessories including the extended DCC aspects. The maximum number of vehicles that can be active with this interface at the same time is at the moment fixed at 32 to keep a regular DCC refresh throughput.

DCC Options

For DCC vehicles you can specify whether the long or short address will be used based on an offset. All vehicles with addresses equal or higher than the specified value use the long address. This is relevant for the addresses from 1-127 that can be both short and long addresses.

The DCC standard for accessories has some ambiguity. Therefore address 1 in iTrain could be mapped to two different numbers with an offset of 4. Most vendors use the option that resembles the choice when ‘Accessory offset’ is checked and the first four addresses in the DCC address space are not used. If you want to use these unofficial addresses then uncheck this option, but take care that the accessory address space shifts with 4 in that case.

RailCom

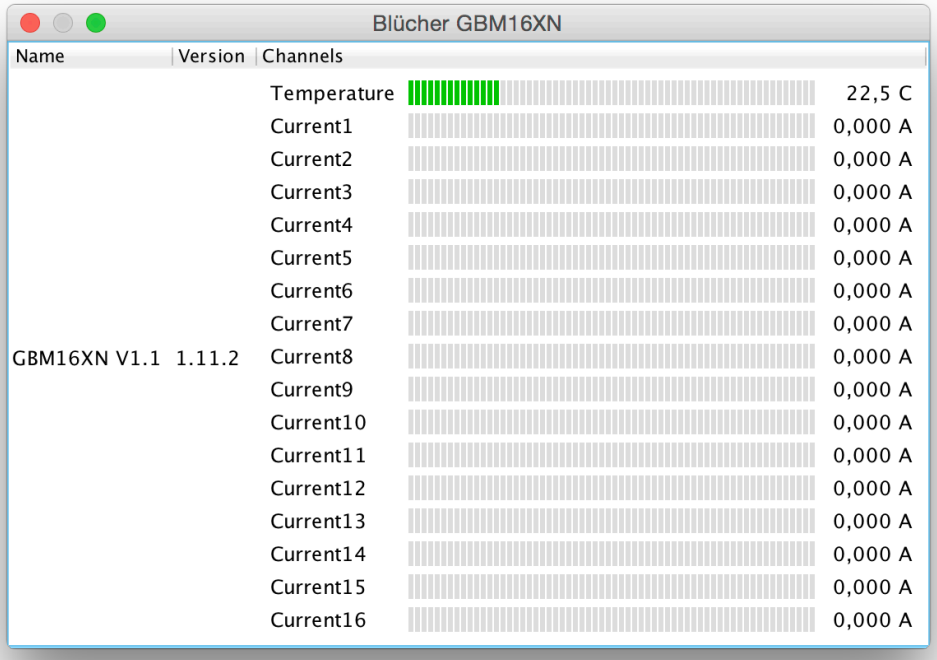
RailCom detectors can read the address of loc decoders that support it and show this number on the switchboard in a feedback element.

When using blocks and attaching the feedback to a block (for example as an extra feedback), the address will be matched with a loc and the train containing the loc will be shown in the block element. This can be used to automatically detect locs in blocks that are put on the track without manual intervention.

TAMS RailCom Link
The TAMS RailCom Link is an additional interface only for feedbacks to read RailCom detectors.

When reading out a CV value of a loc decoder with another device (for example, a throttle or command station), it will show the CV value on the status bar.

Blücher GBM16XN
The Blücher GBM16XN is an additional interface only for feedbacks that exchanges data directly via the built-in USB-Connector of the GBM16XN instead of via a separate LocoNet® adapter. Next to normal occupancy detection this module also reads the address of the locomotive in the section via RailCom.



Additionally the temperature of the device and the currents in all the 16 sections are read and can be displayed in a floating window.

Games on Track

Games on Track GT-position
The GoT GT-P is an additional interface only for feedbacks to use the virtual blocks in GT-Position as feedbacks in iTrain. It is a network interface, because it connects to the GT-P software via a network socket. The GT-P software only works on Windows.

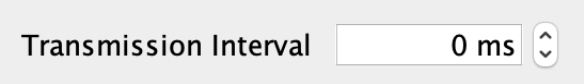
No specific configuration in iTrain is necessary, but in GT-P you have to specify the name of the feedback in iTrain in the comment field of the virtual block or use the same name in both software.

Instead of using the name, it is also possible to specify the iTrain feedback address in the comment field by prefixing it with a # (for example #34). It is not necessary to give GT-P feedbacks in iTrain an address, because this would lead to extra administration. The only advantage would be that you can use the Feedback monitor and that you can change feedbacks names in iTrain without affecting the settings in GT-P.

DMX

There are several interfaces to control DMX devices. Within iTrain, this is primarily intended to control light. BiDiB has a separate module for controlling DMX devices via the BiDiB bus, but there are also separate interfaces that can control DMX directly.

Art-Net
The 'Art-Net' network interface[90] is an additional interface only for accessories, especially light objects, that are controlled via DMX with 256 levels (8-bit) for each color.



The 'Transmission interval' determines how often the DMX-universes will be sent to the interface. A value of zero means all changes are sent directly without delay, but using zero might overload the system.

Speed measurements

In the section 'Speed measurements' was mentioned that there are specific devices that will be controlled via an interface. The µCon-Manager and the LoDi-Rektor can control speed devices like the µCon-RailSpeed via a booster bus as part of an interface with more responsibilities. In this section we focus on interfaces that are only there for speed measurements.

KPF Zeller Speed-Cat
The 'KPF Zeller Speed-Cat' is an additional interface only for speed measurements on roller bank. No specific configuration is necessary.

RTZec Speedbox
The 'RTZec Speedbox' is an additional interface only for speed measurements to measure speed based on two detectors separated 10 cm from each other. No specific configuration is necessary.

---

[90] The DMX-Light-Interface from LSdigital is one of the devices supported by this interface.