

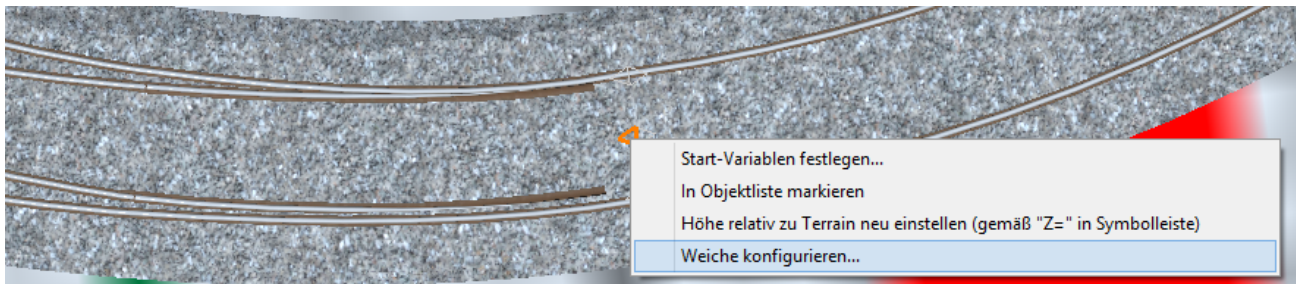
# Routes, signals and electrical switches

## Table Of Contents

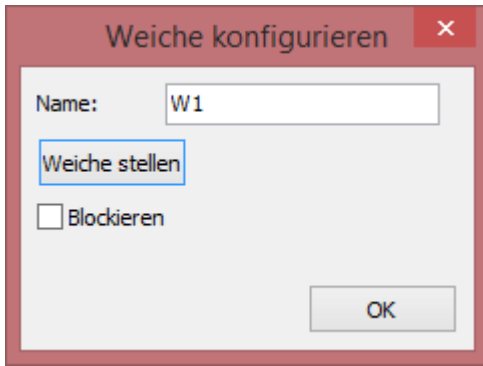
- [1 Prepare switches](#)
- [2 Create district](#)
- [3 Create route](#)
- [4 Add switches and paths](#)
  - [4.1 Further properties of the routes](#)
- [5 Configuring triggers and beacons](#)
  - [5.1 Placing Triggers](#)
- [6 Connecting simple signals](#)
- [7 More complex signals](#)
  - [7.1 Signalobjekt konfigurieren](#)
  - [7.2 Signal im MapEditor konfigurieren](#)
- [8 Equipping the vehicle](#)
- [9 Script Implementation](#)
  - [9.1 Vehicle](#)
  - [9.2 Simple Signals](#)
  - [9.3 Complex Signals](#)
  - [9.4 Request By Scenery Object](#)

## 1 Prepare switches

It is not *necessary*, but *advisable*, to give names to the switches (at least those which are to be used in [routes](#) ). To do this in "Track" mode, right-click on the marker pointing in the direction of the turnout tip and click on "Configure turnout..." in the context menu.



In the appearing dialog box the name of the switch can be entered immediately.

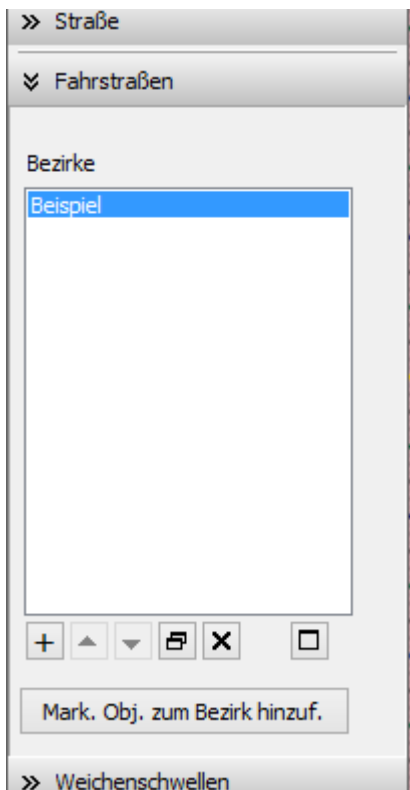


## 2 Create district

For the sake of clarity and performance, [routes](#) are grouped into districts that work independently of each other. These can be imagined as individual interlockings.

The grouping is particularly relevant for performance because each district contains an exclusion matrix which has the size "number of [routes](#) to square" and which is used to check whether an "enemy" route is already in use when a route is requested. Suppose you had a large network with 20 signal boxes with 10 [routes](#) each. If you didn't define districts, you would need a 200x200 matrix with 40,000 entries. If the [routes](#) were grouped to the respective signal boxes, then there would be 20 matrices with only 10x10 = 100 fields, i.e. only a total of 1,000 entries (compared to 40,000!).

The districts are managed via the list box known in many other areas, which is located on the left in the section "[Routes](#)".



### 3 Create route

In the Properties dialog box (accessible via double-click or the button with the rectangle below the list) the district can first be given a name and further basic properties of the district can be set:

- **Deactivate, if track is free again:** Active [routes](#) will be activated automatically, if its tracks are free after they were occupied before. Option is set on for default.
- **Signals show only, if route is active:** If this option is set off, the switch [signals](#) assigned to a special route will be also on, if the route *could* be activated, even if it is not activated. So they will always show the current direction of the junction and not only if there is an assigned route active. Option is set on for default.
- **Deactivation by switching junction:** If you switch a junction manually although there is a route active using this switch, it will be automatically deactivated. Option is set on for default.
- **Auto activation by track occupation:** if a route will be occupied even if it is not active and if the junction directions fit and there is no "enemy" route active, it should be activated automatically. Option is set on for default.

Below these options, you can create or remove the individual [routes](#) with "Add" and "Delete". If one of them is clicked, its properties can be edited directly below the list:

[lotus-simulator.de/index.php?attachment/18734/](https://www.lotus-simulator.de/index.php?attachment/18734/)

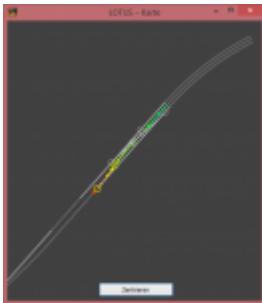
Dies umfasst aktuell den Namen der einzelnen Fahrstraße und die Weichen, die gestellt werden sollen, wenn die Fahrstraße aktiviert wird.

## 4 Add switches and paths

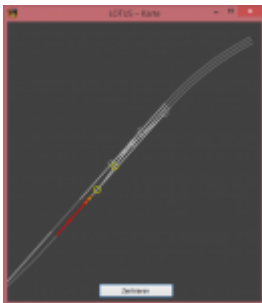
Together with the properties dialog box, a 2D map also opens. In it, paths are added or removed from the [routes](#). In general, only paths are added or removed; the MapEditor automatically determines which switches belong to the route.

Paths can be added to one of the following three categories:

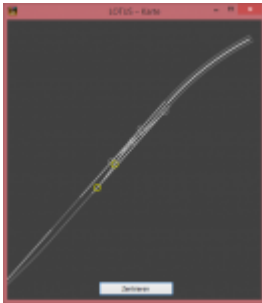
- The normal paths that make up the actual route. This area starts behind the signal and ends behind the last switch. So that the route can be inserted, these must be free of vehicles and the associated switches must be set. Accordingly, they are also used to determine the route exclusion, i.e. which other [routes](#) can be set at the same time. As soon as the vehicle enters this area, the route is considered to be occupied and the associated signal drops to "Stop", provided that this has been configured accordingly. If the vehicle leaves this area completely again, the route is dissolved.



- Activation paths: These are only used if the interlocking district is a "real" interlocking, i.e. not if it is only a matter of protected switches, which are set on site by actuating commands from the vehicles. The activation paths are located in front of the associated signal. As soon as these are used, the interlocking automation or the virtual dispatcher is requested to insert the corresponding route, provided that it fits to the trip and the train is less than 2 minutes early or the option "Autom. activation" of the route has been checked - and of course only if it is allowed/capable to be inserted.



- Check paths: The area between the last switch and the target signal: If the train is already here, then the route can be dissolved, since all switches have been cleared. Nevertheless, no [routes](#) should be inserted if there is a train here or the track belongs to a route that has already been inserted. Therefore, this area is checked before the route is inserted.



Depending on which category paths are to be added to or removed from, one of the three associated buttons must be activated by clicking on it. After that, the paths can be added to or removed from the 2D map by clicking on them.

Except for audit trails, care must be taken to align the paths correctly. The MapEditor automatically ensures that the paths, if they are correctly linked, are aligned uniformly. If this alignment is to be reversed, the associated "Reverse direction" button must be clicked.

The entry in the route list is updated each time, so that you can always see there, among other things, whether the corresponding switches have been added as desired.

Attention: The option "Auto-activation" should never be used in "real" interlocking districts, where especially switches are driven from different directions! It should only be used in typical simple situations in the tramway area, where the [routes](#) consist of only two switches at most, which are driven from one direction only.

#### 4.1 Further properties of the [routes](#)

- Autom. activation: This is actually cheating, which is why there will be a corresponding realism option that can be used to disable this feature: In reality, every train in a interlocking district must generally also have a train number or - in the case of shunting [routes](#) - at least communicate with the dispatcher. The fact that one simply comes along "like that" with a train and then [routes](#) are automatically engaged anyway, otherwise only occurs with self-block [signals](#), but never with [routes](#) that cover switches. To make it a bit nicer for the explorer, you can provide [routes](#) with this option. This way - if the train is not running on schedule - the route will be set automatically as soon as the train passes the activation paths.
- Remove all paths and switches from route: This is exactly what is done here - the route is then free of any paths.

## 5 Configuring triggers and beacons

Now that the [routes](#) have been defined, it is time to use triggers to activate and deactivate them

automatically. "Triggers" are special scenery objects (invisible in the simulation). Triggers will be able to take over various tasks, but at the moment they are only used to insert and resolve [routes](#).

## 5.1 Placing Triggers

You can find the trigger object on the right in the category "Helpers". It just has to be placed "approximately" on the track; the internal "connection" to the track is still guaranteed. An exact alignment/rotation is also not necessary, but the rough direction (forwards or backwards) is very important, because the sensors/triggers always only react to the sensors/triggers of "their" direction.

Right-click on the trigger and select "Trigger Properties..." to open the properties dialog:

[lotus-simulator.de/index.php?attachment/8277/](https://www.lotus-simulator.de/index.php?attachment/8277/)

First, you select the route district to which the trigger should belong.

If the vehicle sends a positioning command to the trigger, then it should insert a certain route - depending on the positioning command. This behaviour can be set in the following three dropdown menus. If the trigger is to ignore certain or all positioning commands, "-" must be selected in each case.

If the vehicle leaves the area of the driveway, it is to be dissolved (deactivated) again. If this should not be done by the track clearance messages, a trigger can also be assigned a route for this purpose. Here the trigger does not "wait" for a certain information/request from the vehicle, but the cancellation is triggered immediately when the trigger is "touched".

[lotus-simulator.de/index.php?attachment/8279/](https://www.lotus-simulator.de/index.php?attachment/8279/)

## 6 Connecting simple [signals](#)

A simplified system is used for simple [signals](#) (e.g. standard BOStrab switch [signals](#) such as W1, W2, W3 etc.), where a scenery object only indicates whether a certain route has been taken or not:

- Placing and selecting the signal
- Select the district on the left
- Click "Add marked object to selected district" below the district list.
- Select the route to display this signal.

## 7 More complex [signals](#)

### 7.1 Signalobjekt konfigurieren

Komplexe Signale müssen zuerst vorbereitet werden: Im Object & Vehicle Tool gibt es im Bereich "Allgemeine Einstellungen" die Schaltfläche "Signalbegriffe". Hierüber kann ein Texteingabe-Feld geöffnet werden, in welches (vereinfacht gesagt) diejenigen Signalbegriffe eingetragen werden müssen, die das Signal zeigen kann. Hierbei ist aber zu beachten, dass

- das "Nichtvorhandensein" von Fahrstraßen usw., also ein Hp0 oder ein dunkles Weichensignal usw., nicht ein eigener Signalbegriff sind
- keine "Kombinationen" eingetragen werden
- maximal sind 32 Zeilen und somit Begriffe erlaubt

Kann das Signal beispielsweise sowohl Vor- als auch Hauptsignal-Begriffe anzeigen, dann erhält es bspw. die Einträge:

- Hp1
- Hp2
- Vr1
- Vr2

**FALSCH** wäre also sowas wie:

- Hp0+Vr0
- Hp1+Vr0
- Hp2+Vr0
- Hp1+Vr1
- usw.

Warum das so ist, wird vermutlich klarer, wenn wir uns mit der Konfiguration im MapEditor beschäftigt haben! 😊

## 7.2 Signal im MapEditor konfigurieren

Sobald dem Signal-Objekt auf die beschriebene Weise die möglichen Signalbegriffe mitgeteilt wurden, erscheint im MapEditor, wenn man auf das Signal mit rechts klickt, im Kontextmenü der Eintrag "Signal-Eigenschaften". Wird dieser angeklickt, öffnet sich das hierfür vorgesehene Dialogfeld.

Von der Struktur her gibt es für jeden Signalbegriff einen Listen-Abschnitt, zu dem jeweils beliebig viele Fahrstraßen hinzugefügt werden können. Die Anzahl der Signalbegriffe ergibt sich durch die Eingaben beim Signal-Objekt im Object & Vehicle Tool. Für jeden Fahrstraßeneintrag kann jeweils individuell der Bezirk ausgewählt werden, damit es auch möglich ist, Signalbegriffe bezirksübergreifend zusammen zu stellen. Außerdem wird jeweils ausgewählt, ob geprüft werden soll, ob die Fahrstraße aktiv ist, deren Gleise belegt sind oder ob sie aktiv ist und gleichzeitig ihre Gleise nicht belegt sind.

Das ganze System arbeitet nun wie folgt: Für jeden Signalbegriff wird stets geprüft, ob für zumindest eine der Fahrstraßen die ausgewählte Bedingung erfüllt ist. Falls dies der Fall ist, "gilt" dieser Signalbegriff. Schließlich werden die so ermittelten "Gesamtzustände" jeder Fahrstraße kodiert und per Script an das Signal übertragen.

## 8 Equipping the vehicle

For a vehicle to be able to interact with the beacons, it must have at least one sensor, bi-directional vehicles require at least one at each end. This is set up in the object settings in the "Sensors" section:

[lotus-simulator.de/index.php?attachment/8309/](https://lotus-simulator.de/index.php?attachment/8309/)

The parameters mean:

- Y position: The position of the sensor on the vehicle in the longitudinal direction.
- Against direction of travel: The sensors always only interact with triggers that have the same orientation, so it must be specified here in which direction this sensor is to be installed.

## 9 Script Implementation

### 9.1 Vehicle

On the script side, there are already the following possibilities of interaction between vehicle sensor and track trigger:

- Send messages to all triggers in range with `SendMessageToTrigger(Self; ID: string; value: string; sensor_index: integer);`. So far the following IDs are supported:
  - "SWITCH" with the values (each as string!) "0" (left), "1" (right) and "2" (straight ahead) for transmitting a positioning command.
  - "LINEROUTE", where the numbers of the line and the route and the name of the ITCS server are transmitted, as a string separated with a "/". So line 123 with route 45 becomes the string "123/45/{name}". The transmission of the line and route ensures that LOTUS uses the available, valid [timetables](#) that match the ITCS server name to find out which route is stored for this and uses this to give the appropriate control command. The name of the ITCS server is usually fetched beforehand with the function `PIS_GetITCSServer(self: integer)`.
- If a procedure is included in the script with the declaration `procedure OnEnterLeaveTrigger(triggerid: string; entering: boolean; sensorindex: integer);`, it is always called when the vehicle reaches or leaves a trigger. "Triggerid" is currently not used, later it will be used e.g. for identification for IBIS etc..

In GT6N, for example, both functions are used together: When the driver activates a positioning command, this value (for a certain maximum distance) is "earmarked". If `OnEnterLeaveTrigger` is then triggered, the script immediately uses `SendMessageToTrigger` and sends the command to the trigger. Other variants are also possible: If the trigger has a larger radius (e.g. for contact positioning commands or longer contact loops), `SendMessageToTrigger` could also be triggered immediately if the driver actuates the positioning command.

### 9.2 Simple [Signals](#)

Similar to the [traffic lights](#), only the "PUBLIC" variable `signalroute_active: boolean;` must be included. If everything is linked correctly and the route is inserted, this variable becomes "true", otherwise it is "false". However, only simple [signals](#) can be realized with this variable. Complex [signals](#) will be added later! 😊

### 9.3 Complex [Signals](#)



Also with complex [signals](#), the data is transferred via a "PUBLIC" variable, namely `signalstate: integer;`. Attention: In contrast to simple [signals](#), this is an integer variable!

The signal terms are now entered bit by bit in this variable during operation - see also Bit flags explained . If the signal has e.g. the terms Hp1, Hp2, Hp3 and Hp4, then the value of the variable `signalstate` will take the following values:

- 1, if *only* the Hp1 conditions apply
- 2, if only the Hp2 conditions apply
- 4, if only the Hp3 conditions apply or
- 8, if only the Hp4 conditions apply.

If several conditions apply at the same time, then these values are simply added together.

You can then disassemble the conditions as follows:

- if (`signalstate and 1`)  $\neq$  0, then the Hp1 condition is fulfilled
- if (`signalstate and 2`)  $\neq$  0, then the Hp2 condition is fulfilled
- if (`signalstate and 4`)  $\neq$  0, then the Hp3 condition is fulfilled
- if (`signalstate and 8`)  $\neq$  0, then the Hp4 condition is fulfilled

It is more complex, but still problem-free, when additional indicators are used.

Example: A Hp4 only capable subway signal with track number display. The signal terms would then be:

- Hp4
- Track 1
- Track 2
- Track 3
- Track 4
- Track 5
- Track 6
- Track 7
- Track 8
- Track 9

Assume that there are three [routes](#) A, B and C leading to track 3, 4 and 7.

Then in the MapEditor in the signal configuration all three [routes](#) are assigned to the signal term Hp4, to the signal terms track 3, track 4 and track 7 only the [routes](#) A, B and C, and to the remaining signal terms no [routes](#) are assigned at all.

According to the above formula, the variable `signalstate` will then take the following values:

- Route A: Hp4+Track 3, i.e.  $1 + 8 = 9$
- Route B: Hp4+Track 4, i.e.  $1 + 16 = 17$

- Route C: Hp4+Track 7, i.e.  $1 + 128 = 129$

In the signal script the signal terms can then be taken apart again with the above "and-formulas":

If `(signalstate and 1) <> 0`, then the signal should go to Hp4, otherwise to Hp0. Only if it shows Hp4, then it should check the remaining values in `signalstate` and then derive accordingly how it should display the respective track numbers.

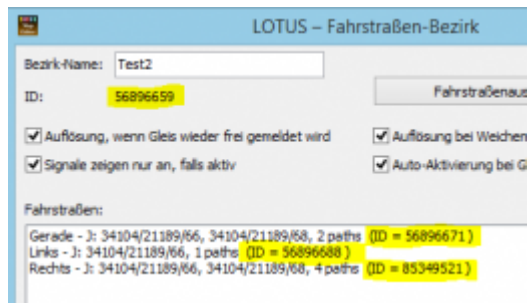
## 9.4 Request By Scenery Object

Scripts in scenery objects can request or resolve [routes](#), so that e.g. corresponding key switches at [traffic lights](#) etc. can be implemented.

The following procedures are available for this purpose:

- procedure `SRT_Request(self: integer; districtID, routeID: integer)` requests a route, but this route is then only inserted if the conditions for this are given (e.g. no blockage by other [routes](#) or occupied tracks).
- procedure `SRT_Deactivate(self: integer; districtID, routeID: integer)` dissolves one or all [routes](#), regardless of whether there are still trains on the track sections assigned to the route(s). To resolve all [routes](#) of the route district, `routeID` must be set to -1.

The identification is done via the IDs of the route districts and the [routes](#), here marked in yellow:



To ensure that a scenery object can be used universally, start variables should be used for all IDs so that they can be set to concrete values in the MapEditor.

If, on the other hand, the IDs in the scenery object are hard-coded to fixed values, the scenery object can only be used on a specific [routes](#) on a specific map.