

# Passengers

## Table Of Contents

- [1 Vehicle](#)
  - [1.1 In General](#)
  - [1.2 Set up seats](#)
  - [1.3 Set up doors](#)
  - [1.4 Set up further path points](#)
  - [1.5 Connect path points](#)
  - [1.6 Inside lighting](#)
- [2 Doors in scripts](#)
- [3 MapEditor](#)

## 1 Vehicle

### 1.1 In General

The configuration of the [passengers](#) in the vehicle is done in the Object&Vehicle tool in the section "Passengers". There are two lists for this: The list of path points that will be connected later and along which [passengers](#) can walk (this includes doors!) and the list of seats and standing places where [passengers](#) place themselves during the journey:



The operation of the two lists is done "as usual" with the corresponding buttons and also the binding to the respective carriage parts or other animations is done as usual with the buttons "Bind to animation".

For the sake of simplicity, however, it is also possible here to click on the "Add path points graphically" or "Add places graphically" button and thus activate it and then click on the mesh at the desired places in each case.

### 1.2 Set up seats

It is now recommended to define the seats first. For standing places the foot point is defined, for seats the "rearmost" point of the seating area... so to speak there, where the sun never comes... ;-):



The [passengers](#) sit with their backs relatively straight. The rear edge of the back is about 10cm behind the seat reference point.

Whether a seat is a sitting or standing seat and in which direction it should be oriented is set directly in the seat properties.

### 1.3 Set up doors

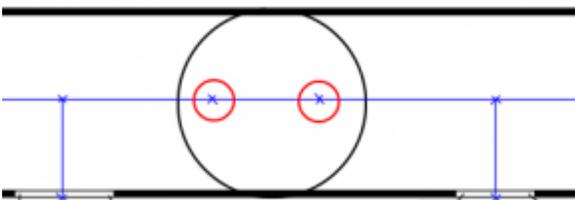
Next, the door points are defined (more precisely: the middle of the door sill, where the passenger changes from "outside" to "inside"). Door points are quite "normal" path points that can be configured as door points in the properties. These should be explicitly assigned a specific name so that they can be easily found in the table.

LOTUS automatically detects whether it is a left or right door by simply checking whether the X coordinate is greater than (right) or less than (left) 0.

### 1.4 Set up further path points

Now all corner points are set which are needed for the paths to be defined next. Corner points are to be provided everywhere where either the direction of travel changes, from where [passengers](#) can move to their place by a direct route or where they change the carriage part.

When passing between different parts of the carriage, the following placement of corner points is recommended in the area of the transition:



## 1.5 Connect path points

Finally, the path points must be connected to each other to form paths. The procedure for this is:

- Click on the button "Connect path points graphically" (will then be colored).
- Click with the mouse on the path points in the 3D window (!) that are to be connected with each other continuously.
- Click the button "Connect path points graphically" again.
- Repeat this process for another continuous chain.

Faulty connections can currently only be disconnected by deleting one of the path points involved.

## 1.6 Inside lighting

To ensure that the [passengers](#) in the interior are also illuminated, both the path points and the seats (in their properties) must be assigned the desired light room that is to "apply" to the corresponding [passengers](#).

Attention: At the moment these light rooms may only contain a maximum of about 4 light sources (ideally only 2, for people who turn the graphics settings all the way down)!

## 2 Doors in scripts

First of all: The script control regarding [passengers](#) has nothing to do with the remote controlled doors!

For each door, the following public variables can be created, where the index to be specified corresponds to the order of the doors in the list of path points, where the first door is designated "0" and the path points that are not doors are not counted:

- `DoorOpen_#`: boolean: This is a writable variable that tells the simulator or the [passengers](#) whether the respective door is currently open. If no variable is assigned, the door is always considered "open".
- `DoorReqIn_#`: boolean: This variable is constantly set to "true" by each passenger as soon as he or she has made his or her way to the door for the purpose of exiting and/or is waiting to exit - it thus corresponds to the stop request from inside the vehicle. The variable must be reset manually to "false"!
- `DoorReqOut_#`: boolean: This variable is constantly set to "true" by each passenger who is outside the door and wants to board - it thus corresponds to the outer door buttons. Accordingly, it is also only set to "true" when the respective passenger is near the door. The variable must be reset manually to "false"!
- `DoorOccupied_#`: boolean: This variable is set to "true" if there is a passenger in the respective door, so it corresponds to a virtual light barrier.
- `DoorEntryAvailable_#`: boolean and `DoorExitAvailable_#`: boolean: If these variables are set to false, then the associated door will be completely ignored by the boarding and/or exiting [passengers](#) (so they won't walk to it either). If one of these variables is not declared, the associated door is interpreted as available.
- `DoorEntryReleased_#`: boolean and `DoorExitReleased_#`: boolean: By setting the variable to "true", the associated door is marked as "released". If a passenger stands unsuccessfully in front of a closed and not released door, he will then not only look for an already opened door, but also for a released one, if necessary, in order to then open it. If the release is revoked, the associated

variable must also be set to "false" again.

### 3 MapEditor

Passengers can basically leave the vehicle anywhere and once they have expressed the wish to stop, they will do so as soon as the doors are open ;-). However, in order for them to then continue walking, they need to find a suitable sidewalk path nearby and on the exit side, otherwise they will simply become invisible once they have exited the vehicle.

In order for [passengers](#) to board, there must be a group polygon near a track occupied by the train, to which the track has been previously assigned in the MapEditor. For this purpose, after selecting the "People" section (left), the polygon must be marked. Then, one after the other, the tracks to be assigned can be clicked with the right mouse button and "Assign track to mark. Group polygon" can be selected.