Manual to Plug-in 1 for EEP 17.1

Dear customers, first of all we would like to heart-fully thank you for purchasing EEP 17.1 Plug-in 1.

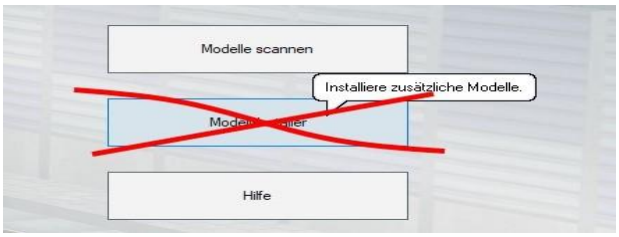
This plug-in includes both improvements and new functions. For this reason, we invite you to read the following information carefully and in particular the installation instructions.

Glossary:

Manual to Plug-in 1 for EEP 17.1

Installation instructions

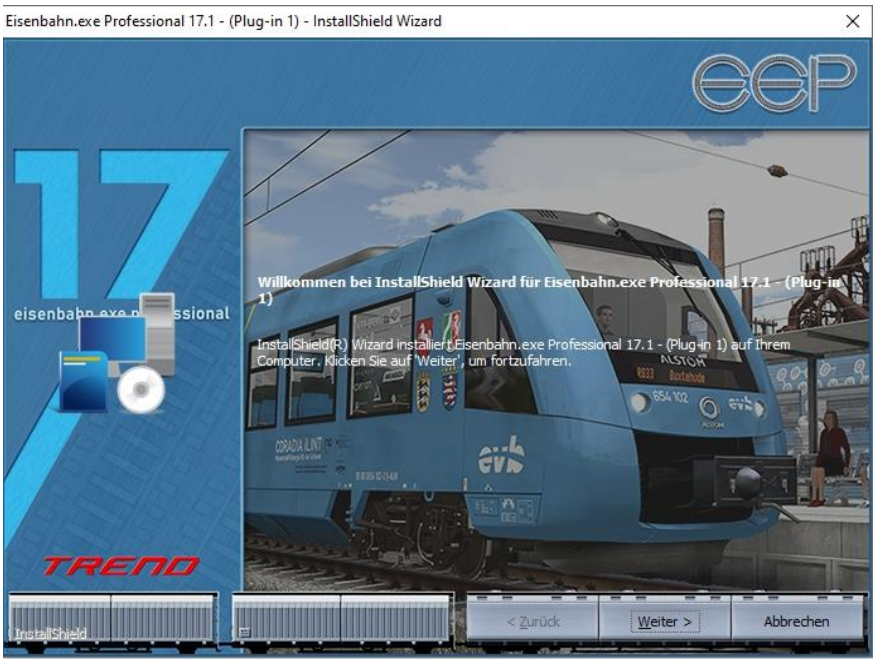
Please make sure that update #1 of EEP 17 is installed. This is the prerequisite for this plug-in. You can check this in the lower left part of the loading.



The Plug-in needs its own installer to upgrade, since it brings new features besides the new models. For this reason it is not possible to open this file directly via the "Model-Installer of EEP 17.1.

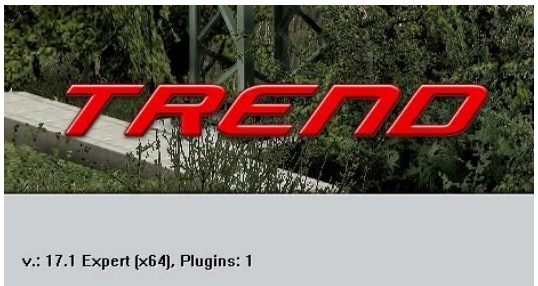
If necessary, please leave your EEP 17.1!

Please start installing the Plug-in by double clicking on the file V17TSP10057. A message informs you that this program will operate changes to your computer. Please allow this by clicking Yes to proceed with the installation. The installation window then appears and prompts you to accept the user license and once these steps are completed, the program then extends the functionality of your EEP 17.1 with new functions and models.



Important note
The window disappears temporarily during the installation. Please wait until it is visible again and you can press the "OK" button. Only then is the installation complete!

If the installation was successful, then you will see in the lower left part of the launch window the EEP 17.1 version followed by the mention „**Plugins: 1“**.



Manual to Plug-in 1 for EEP 17.1

New models contained in Plug-in 1 for EEP 17.1

Tracks (splines)->Roads->Other:
Road tunnel - side wall (SM2)
Road tunnel - side wall - raling (SM2)

Tracks (splines)->Roads->Streets:
Road tunnel - sidewalk only 2m L (SM2)
Road tunnel - sidewalk only 2m R (SM2)

Tracks (splines)->Roads->Tunnels:
Road tunnel - single lane L (SM2)
Road tunnel - single lane R (SM2)
Road tunnel - middle (SM2)
Road tunnel - left side (SM2)
Road tunnel - right side (SM2)

Static structures ->Equipment->Perimeters:
Road tunnel - concrete block (SM2)
Road tunnel - concrete triangle (SM2)
Road tunnel - concrete plate (SM2)

Static structures ->Traffic->Tunnels:
Road tunnel - pillar (SM2)
Road tunnel - portal - round L (SM2)
Road tunnel - portal - round R (SM2)
Road tunnel - portal - rect. L (SM2)
Road tunnel - portal - rect. R (SM2)

Static structures-> Traffic ->Traffic signs and traffic lights:
Road tunnel - clearance height (SM2)
Road tunnel - clearance height WP (SM2)
Road tunnel - lights on (SM2)
Road tunnel - lights on WP (SM2)
Road tunnel - sign pole (SM2)
Road tunnel - tunnel distance (SM2)
Road tunnel - tunnel distance WP (SM2)
Road tunnel - tunnel name nad length (SM2)
Road tunnel - tunnel name nad length WP (SM2)

Signals->Signal posts:
Road tunnel - lane indicator (SM2)
Road tunnel - lane indicator (int) (SM2)
Road tunnel - speed limit 1 L (SM2)
Road tunnel - speed limit 1 L (stretch) (SM2)
Road tunnel - speed limit 1 R (SM2)
Road tunnel - speed limit 1 R (stretch) (SM2)
Road tunnel - speed limit 2 L (SM2)
Road tunnel - speed limit 2 L (stretch) (SM2)
Road tunnel - speed limit 2 R (SM2)
Road tunnel - speed limit 2 R (stretch) (SM2)
Road tunnel - speed limit M (SM2)
Road tunnel - speed limit M (stretch) (SM2)

Signals->Other:

Road tunnel - firefighting eq. (SM2)
Road tunnel - evacuation route (SM2)
Road tunnel - camera (SM2)
Road tunnel - loudspeaker (SM2)
Road tunnel - emergency bay sign (SM2)
Road tunnel - direction sign short (SM2)
Road tunnel - direction sign long (SM2)

Street objects->Bridges and Tunnels->Tunnels:

Road tunnel - 1 lane - split L a (SM2)
Road tunnel - 1 lane - split L a ln (SM2)
Road tunnel - 1 lane - split L b (SM2)
Road tunnel - 1 lane - split L b ln (SM2)
Road tunnel - 1 lane - split R a (SM2)
Road tunnel - 1 lane - split R a ln (SM2)
Road tunnel - 1 lane - split R b (SM2)
Road tunnel - 1 lane - split R b ln (SM2)
Road tunnel - 1 lane - single lane LR swap (SM2)
Road tunnel - 1 lane - single lane RL swap (SM2)
Road tunnel - 1 lane - emergency exit (SM2)
Road tunnel - 1 lane - telephone (SM2)
Road tunnel - 2 lanes - split L (SM2)
Road tunnel - 2 lanes - split R (SM2)
Road tunnel - 2 lanes - lane change (SM2)
Road tunnel - 2 lanes - fans (SM2)
Road tunnel - 3 lanes - split L (SM2)
Road tunnel - 3 lanes - split R (SM2)
Road tunnel - 3 lanes - lane change (SM2)
Road tunnel - 3 lanes - fans (SM2)

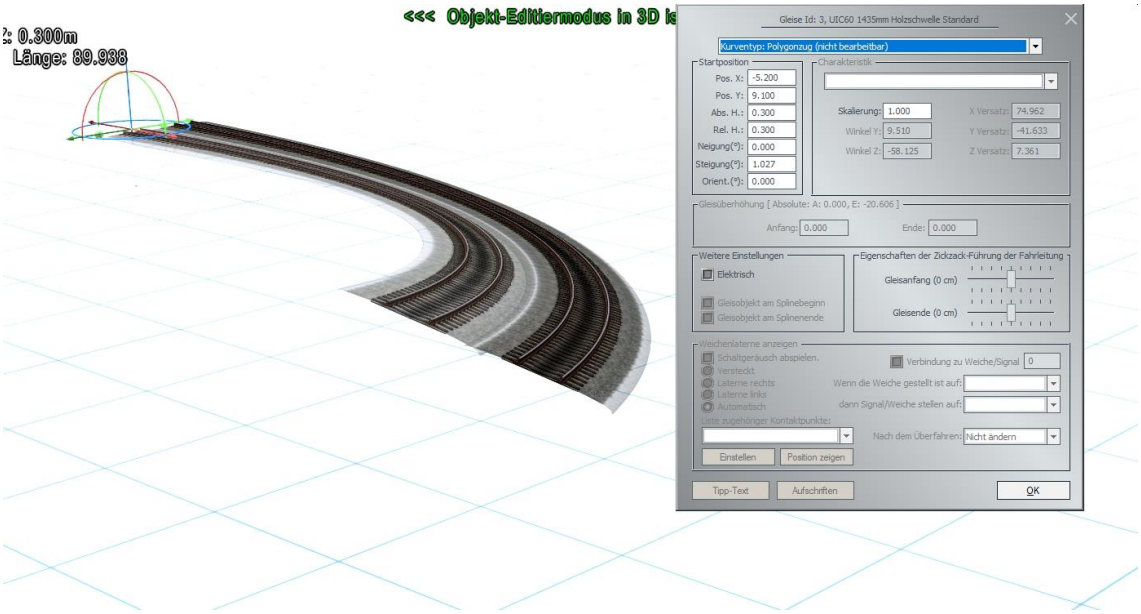
An included demo layout and 2 block files round off this model compilation.

Manual to Plug-in 1 for EEP 17.1

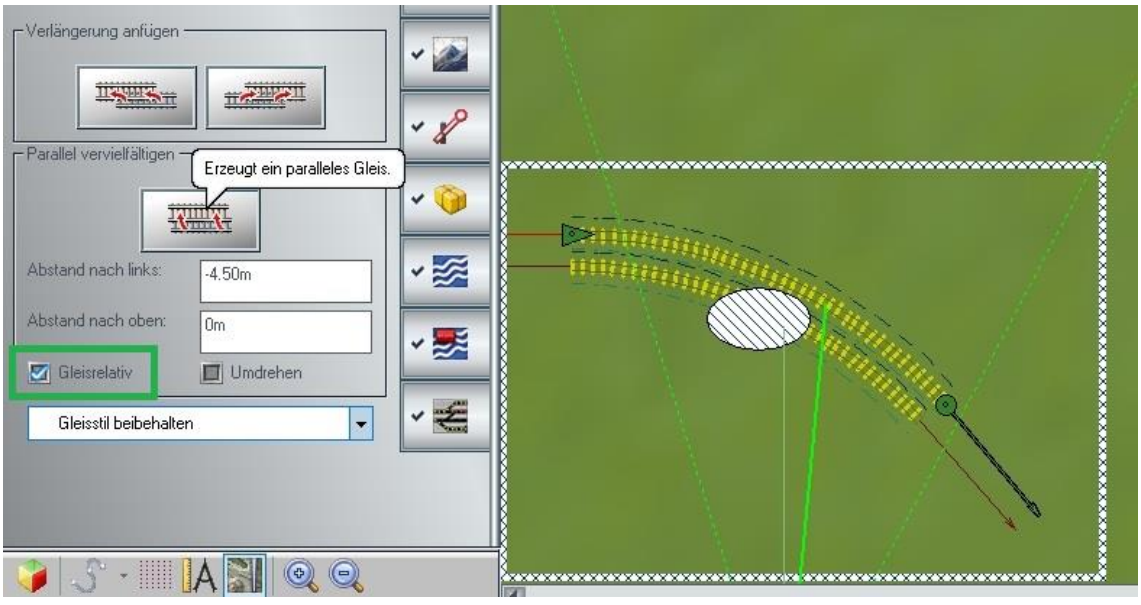
New features allowed by Plug-in 1 of EEP 17.1

Improving the parallel duplication of curve types "Clothoide" and "Cubic"

Until now, EEP used the start and end vector to calculate parallel tracks, roads, paths etc. This led to incorrectness with the curve types "Clothoide" and "Cubic", especially if the duplication was "Track-relative". With the plug-in 1 to EEP 17.1, EEP automatically creates the parallel track with the newly created curve type "Polygonal chain" for these two track types. Here, individual points are used for the calculation and shifted in parallel. This curve type is characterised by a much better track appearance, even if - as in the case shown - the displacement is "Track-relative".

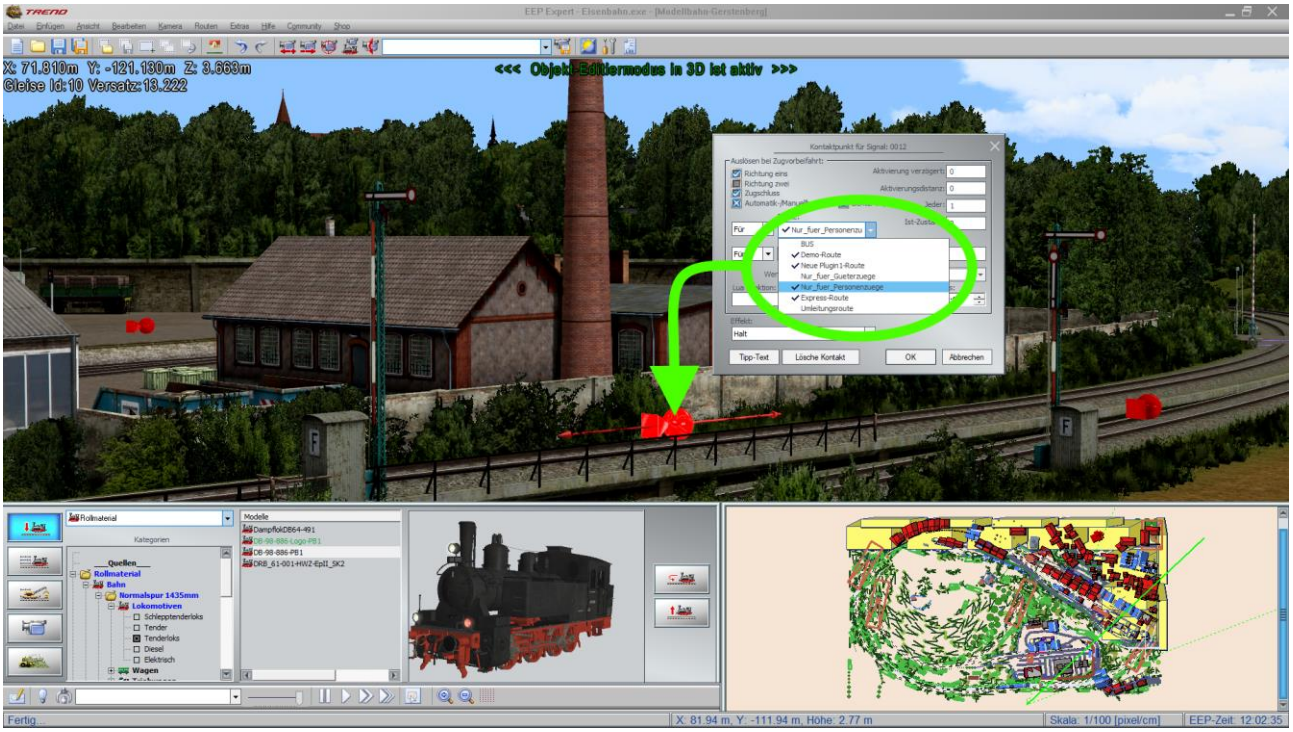


Tracks with the new curve type " Polygonal chain " can no longer be edited afterwards.
The curve type " Polygonal chain " is also not available for general selection. It is only reserved for internal calculation.



Multiple route assignment in contact points

With Plug-in 1 to EEP 17.1 it is possible to assign not only one but several routes to a contact point. This is done by clicking on the corresponding route(s) in the selection box. A tick is automatically placed in front of each selected route. Clicking again removes the route from the selection and the tick disappears.

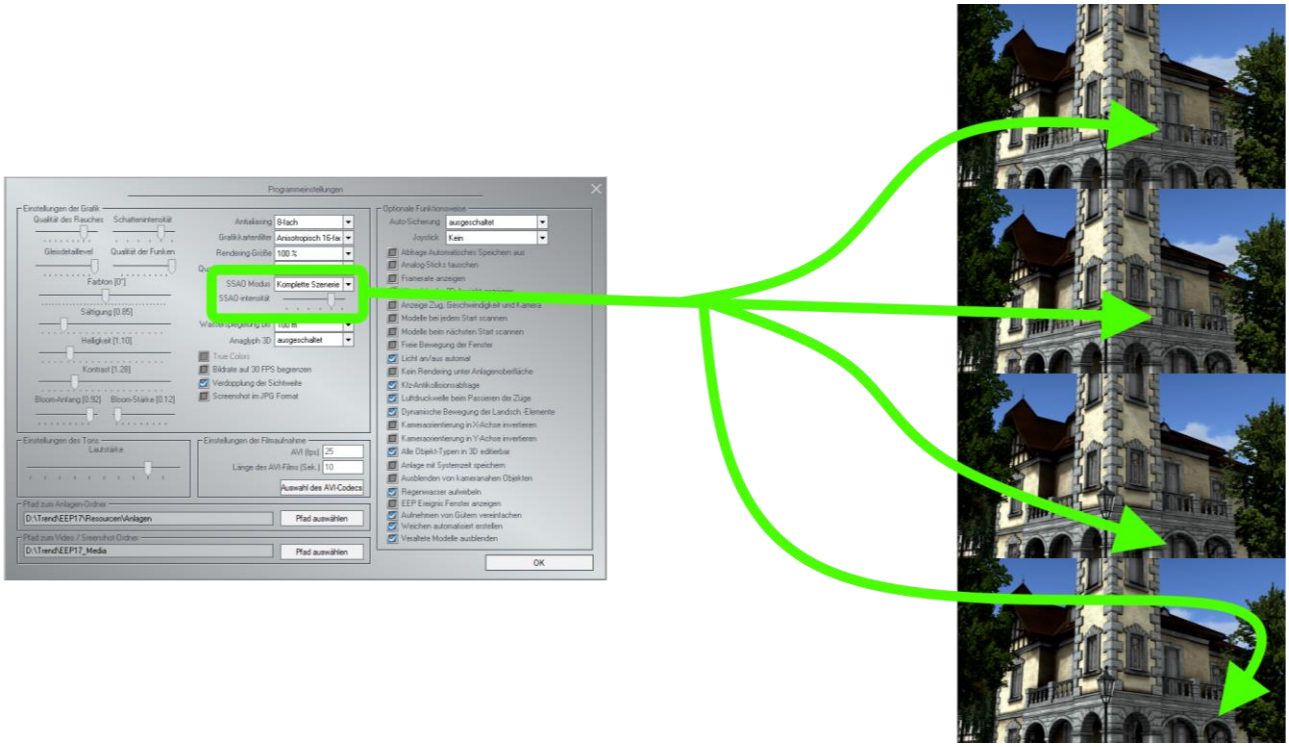


If no route is selected - i.e. if all routes are unchecked - the contact point applies to all routes. You do not have to "tick" all routes. There is no longer an explicit "Any" display.

SSAO- intensity can be changed via slider in future

The SSAO mode increases the plastic impression of 3D objects. This can be activated in the EEP programme settings for various scenarios or generally switched off. With Plug-in 1 for EEP 17.1 it is also possible to change the SSAO intensity individually via a slider.

Since not all models are suitable for this effect, it can of course still be switched on or off individually in their object properties.

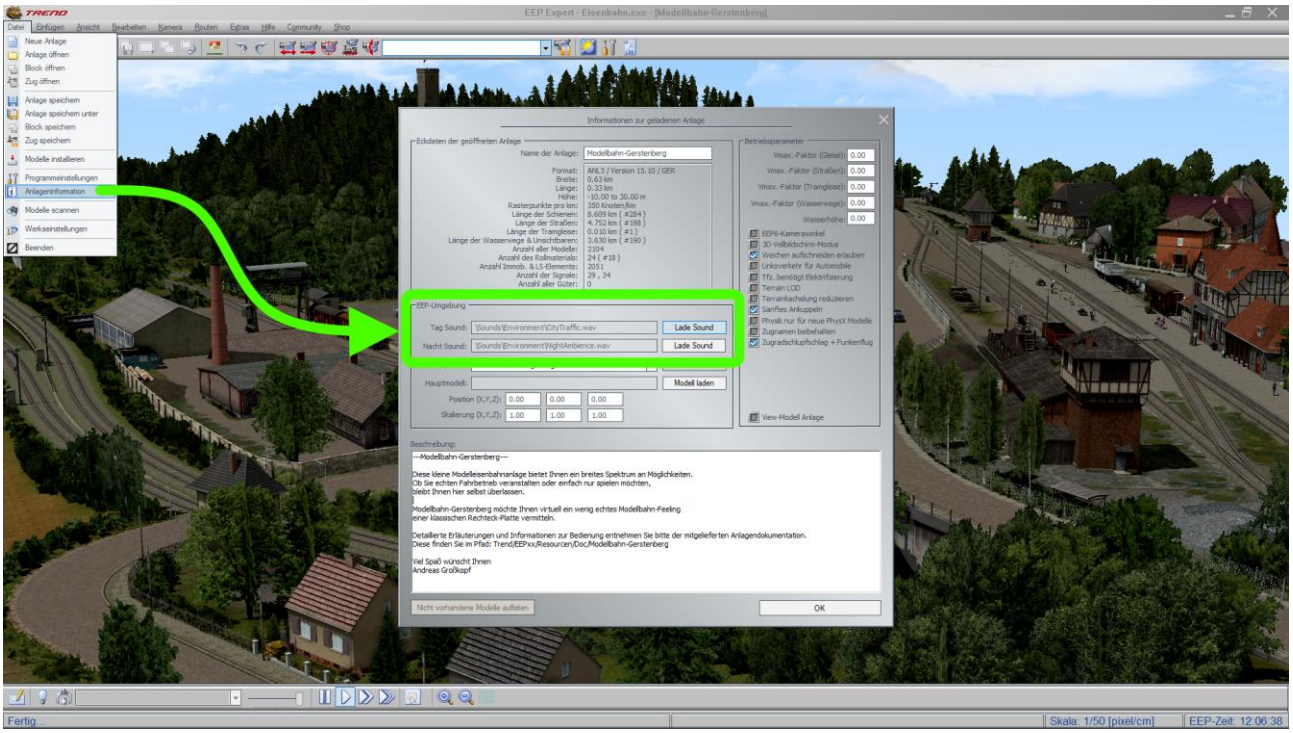


Manual to Plug-in 1 for EEP 17.1

New ambient sounds for EEP

With the plug-in 1 for EEP 17.1 you can now set individual WAV files for a day sound as well as a night sound in the dialogue window "Layout Information" in the area "EEP environnement" via the two buttons "Load Sound".

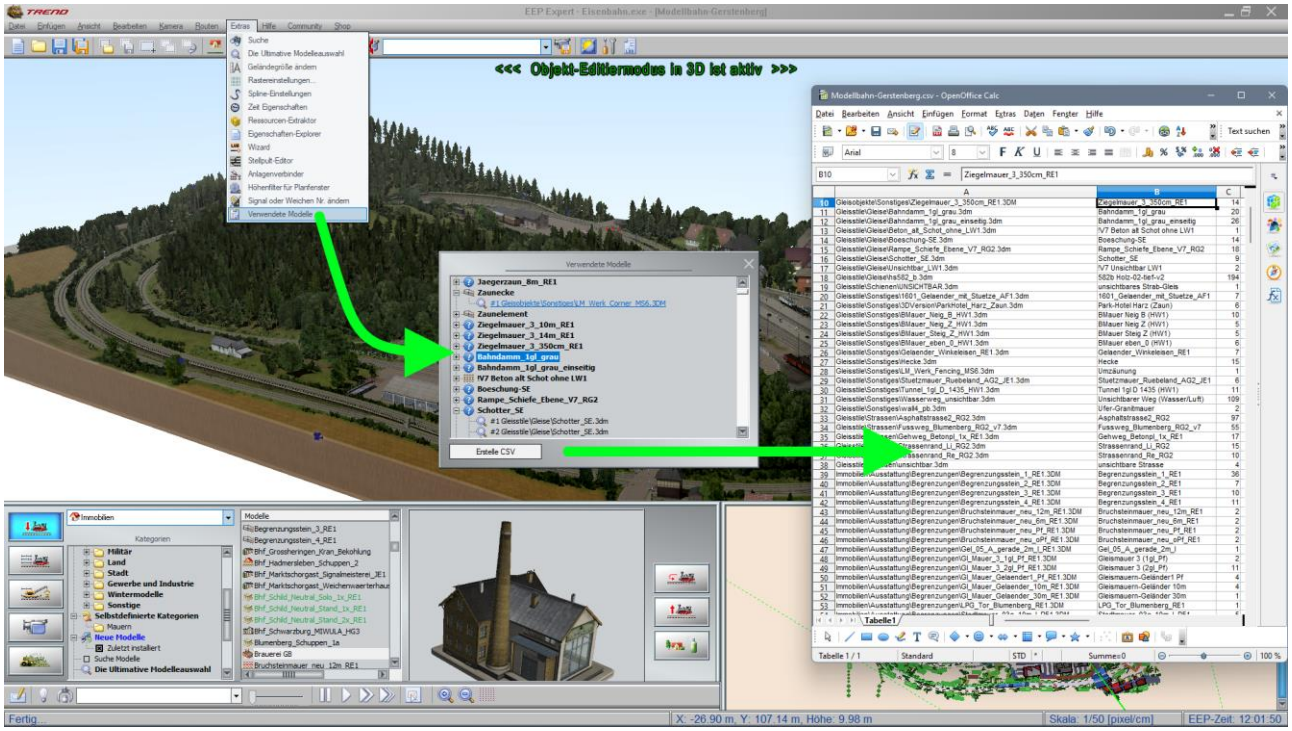
5 new sound files are made available to them in the newly created folder Resources\Sounds\Environment for this purpose. However, you can also load other WAV files from other folders.



Manual to Plug-in 1 for EEP 17.1

Creation of the model list of a layout by the system

Under the menu item "Extras", a new category "Models used" has been created. By clicking on it, you have the option of displaying and saving a list of all models of the layout at any time.



This new possibility can be useful in many ways.

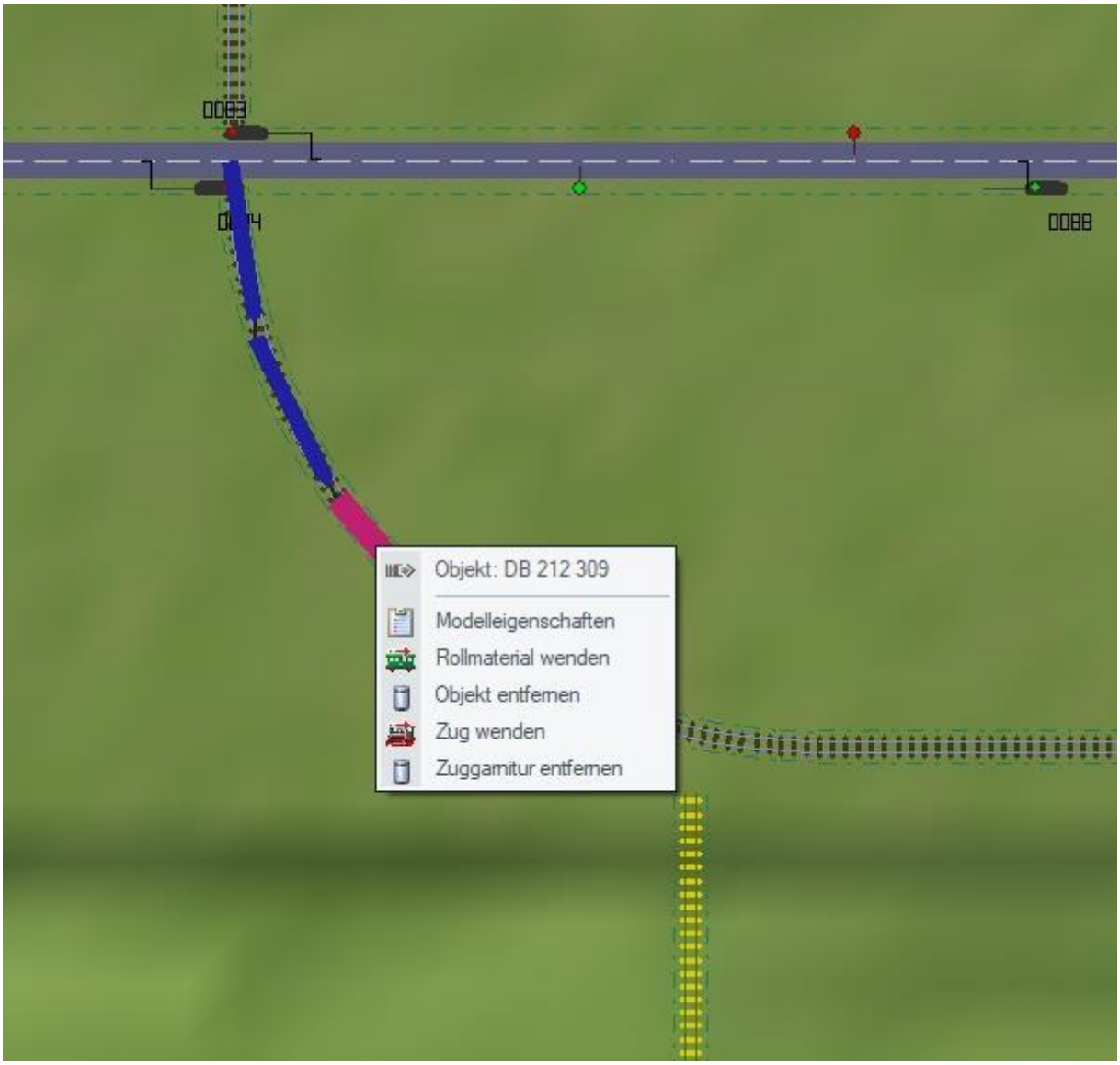
If you have an unknown layout, you can immediately see which models are missing and, thanks to the integrated search function, you can also directly see where the model you are looking for is located.

If you want to pass on a layout yourself, you can easily create a list of all models used without having to use any other tools.

The file created by clicking on the "Generate CSV" button is saved in the same folder as the EEP layout currently in use.

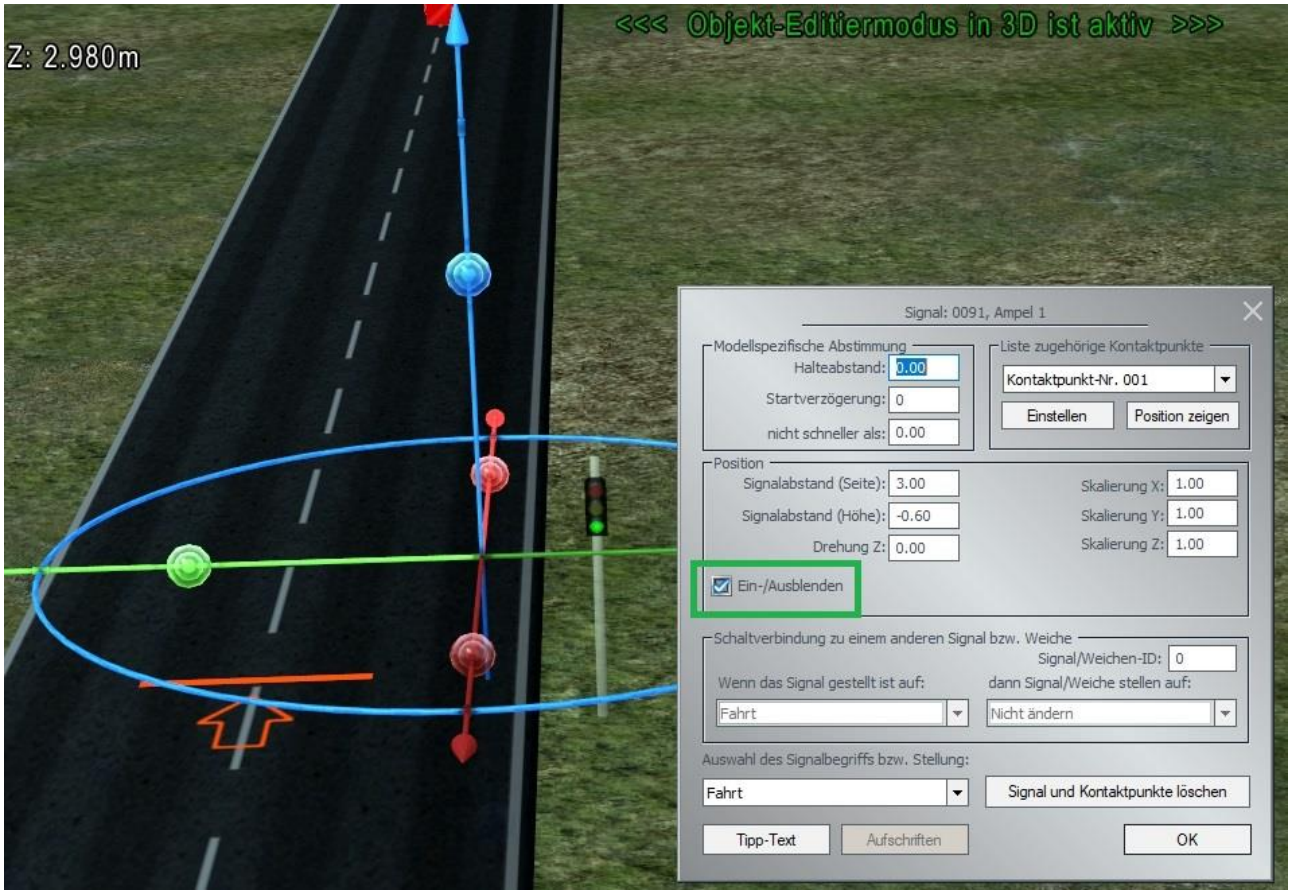
New possibility to remove or rotate rolling stock in 2D mode

With the plug-in 1 to EEP 17.1 it is now possible to turn rolling stock as well as whole trains in 2D mode in addition to 3D mode. The corresponding menu opens by right-clicking in 2D mode on the desired train or rolling stock.



New possibility to hide signals in 3D mode

When working with EEP, it can sometimes be helpful to hide certain models because they are not needed for the current work steps.



With Plug-in 1 for EEP 17.1, this function has now been created for signals.

For this purpose, an additional button was added to the signal menu with the option of hiding a signal by setting a check mark.

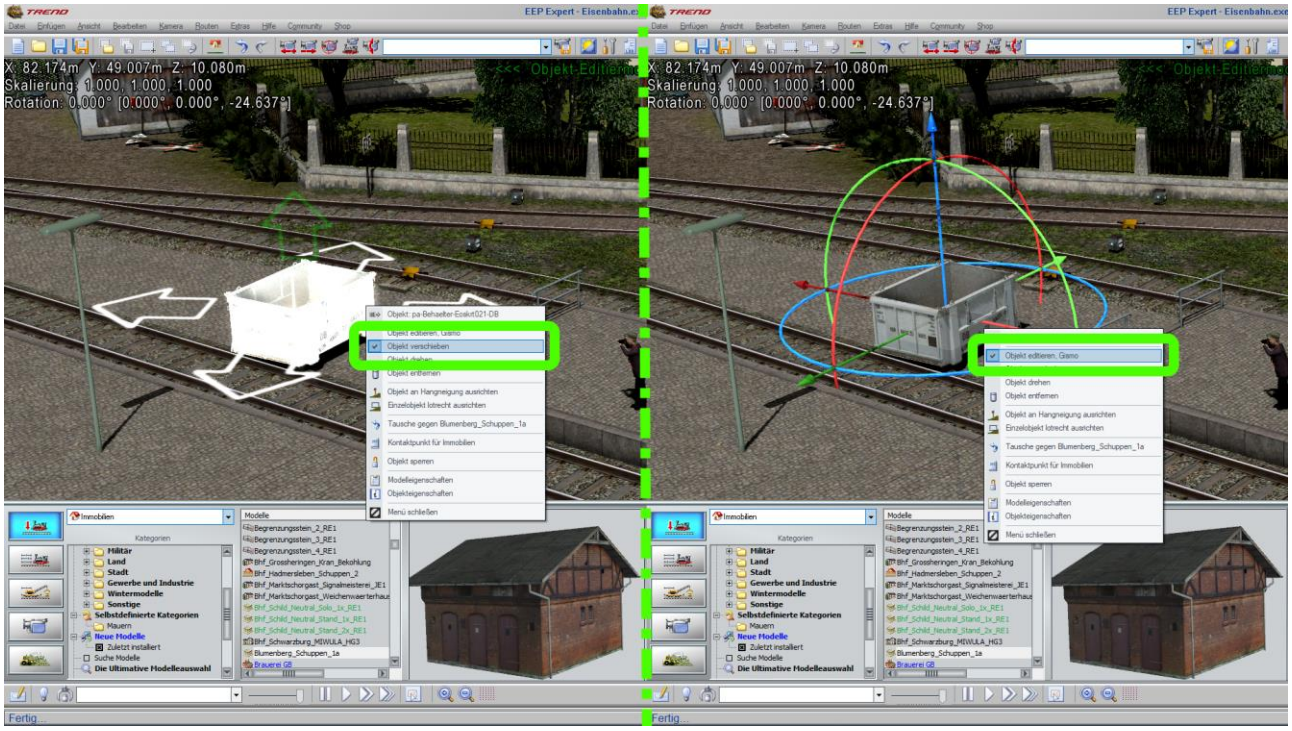
After removing the tick, the signal is visible again.

Manual to Plug-in 1 for EEP 17.1

Possibility to activate the Gizmo in 3D editing mode

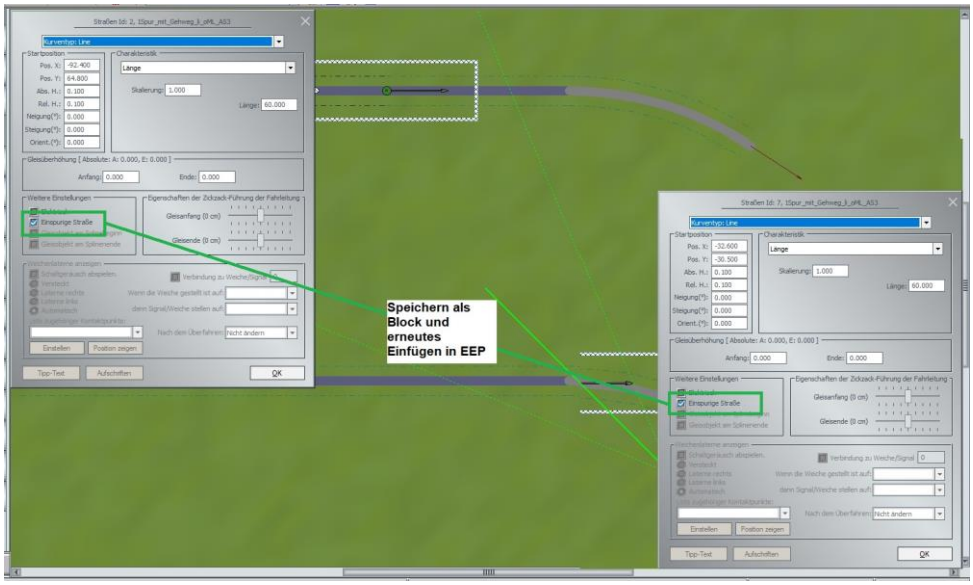
Previously, when moving models, the gizmo was automatically hidden. This was subsequently a hindrance when editing this model in a different way.

This has now been considered by inserting an additional button in the object menu. By clicking on "Edit objects with Gismo" the Gizmo immediately becomes visible and usable again.

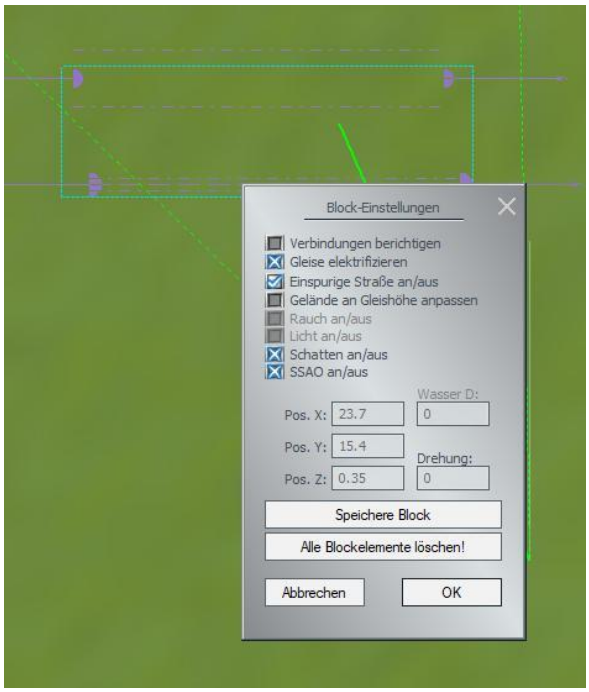


The "One-way traffic" function when copying blocks

Until now, it was not intended that when copying blocks with one-way traffic roads, this function was also saved and transferred in the blocks.
With Plug-in 1 for EEP 17.1, this useful option has now also been introduced. This means that from now on, one-way traffic roads are saved as blocks with this additional function and retain this functionality when they are reinserted elsewhere.

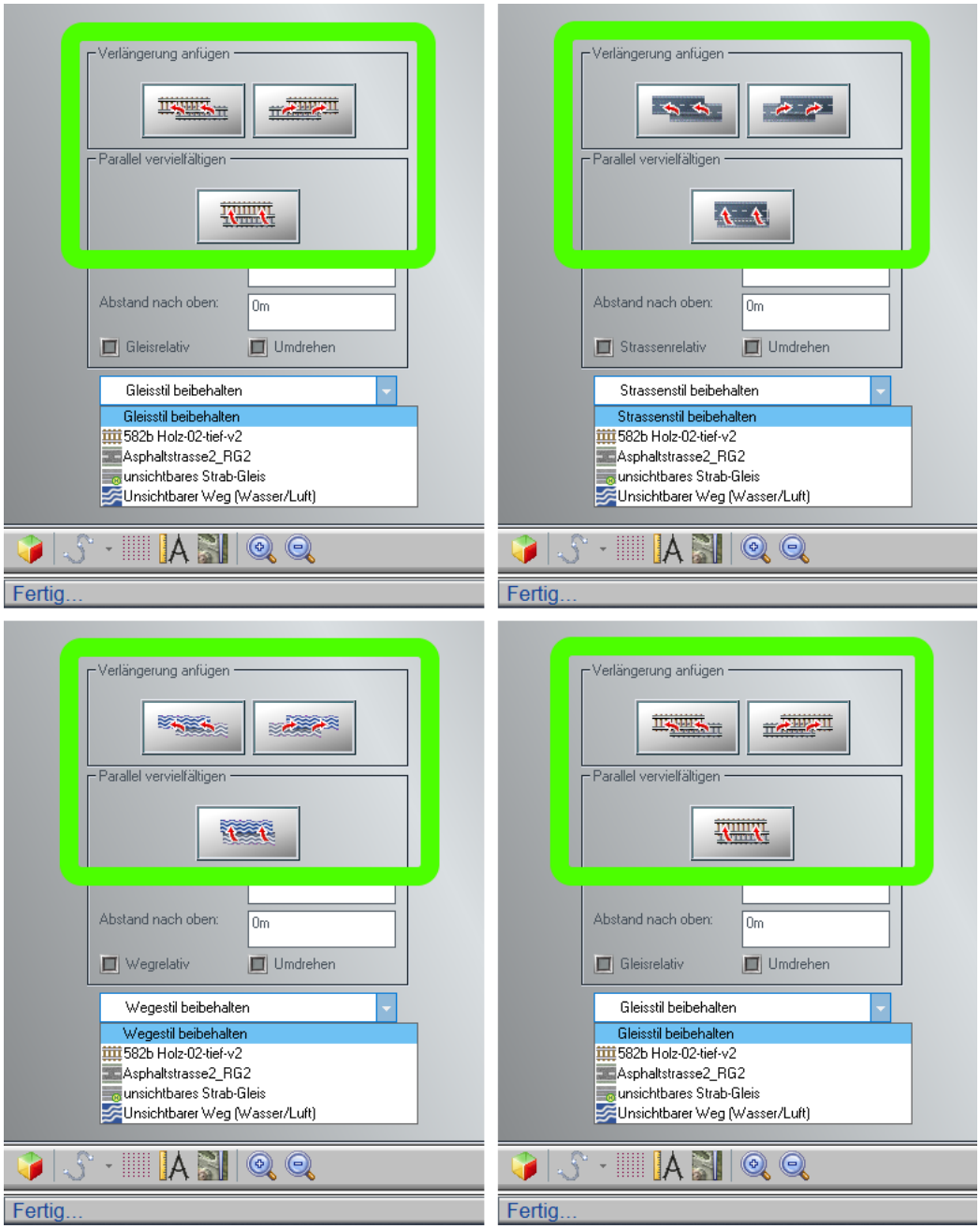


In addition, when copying blocks, it is now also possible to define in the block settings in 2D both one-way and two-way traffic roads as "one-way traffic road".



New properties icons for individual layers in 2D mode

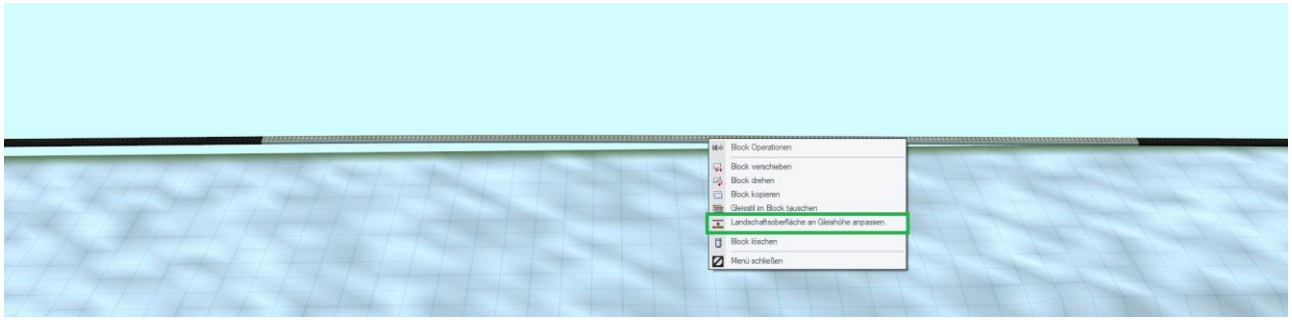
In line with the individual layers in EEP, the visual representation of the associated icons was also changed with Plug-in 1 for EEP 17.1.
This means that it is now possible to see directly from the icons in which layer the spline that is currently being processed is located.



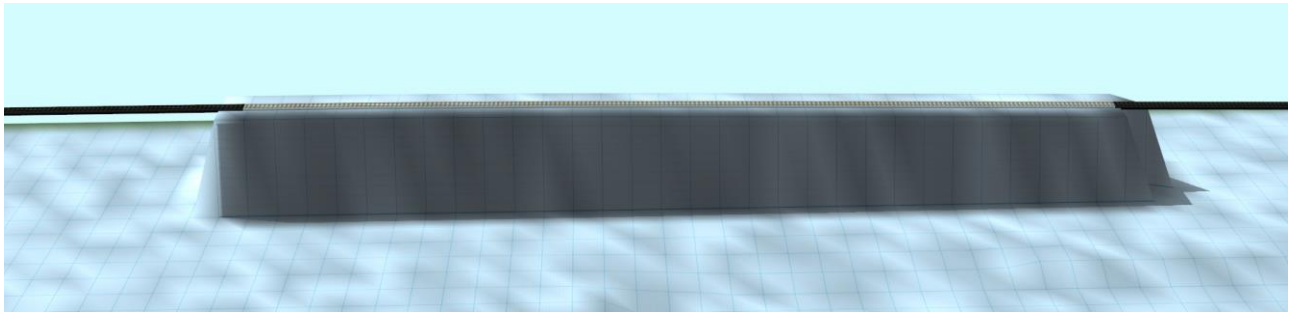
The function "Match surface elevation to track level" is also possible for blocks in 3D

To do this, select the tracks (roads, waterways, etc.) for which you want to adjust the surface elevation with the key combination Alt+left mouse button (function - select several objects).

Then click with the right mouse button on one of the selected objects and the following menu appears:



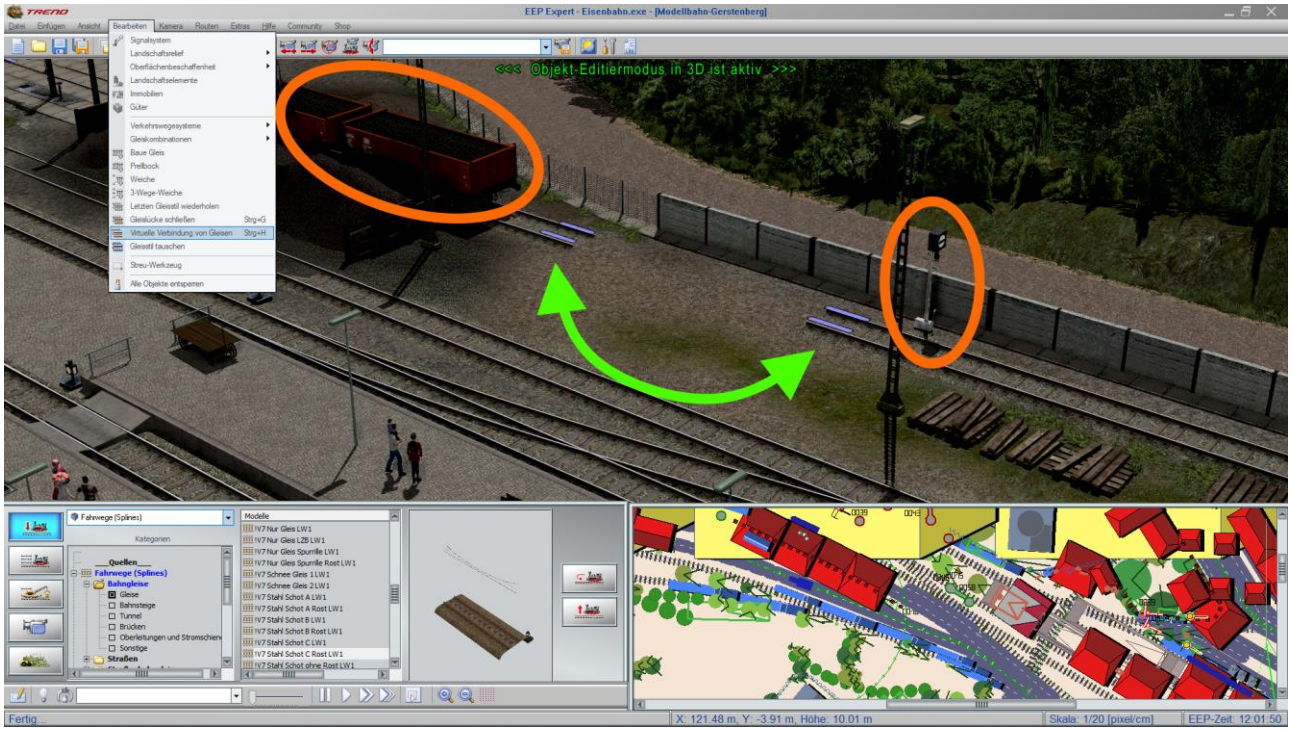
A click on the function "Match surface elevation to track level" leads to the desired result.



Manual to Plug-in 1 for EEP 17.1

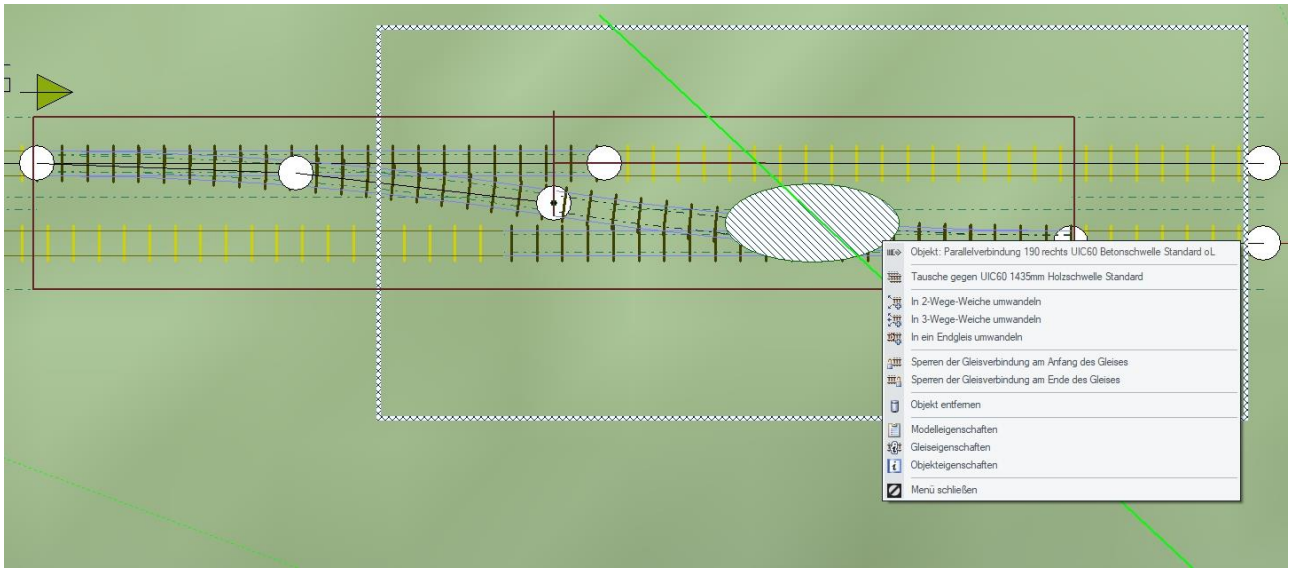
Editing of virtual connections for splines also possible with occupied splines (e.g. by signals)

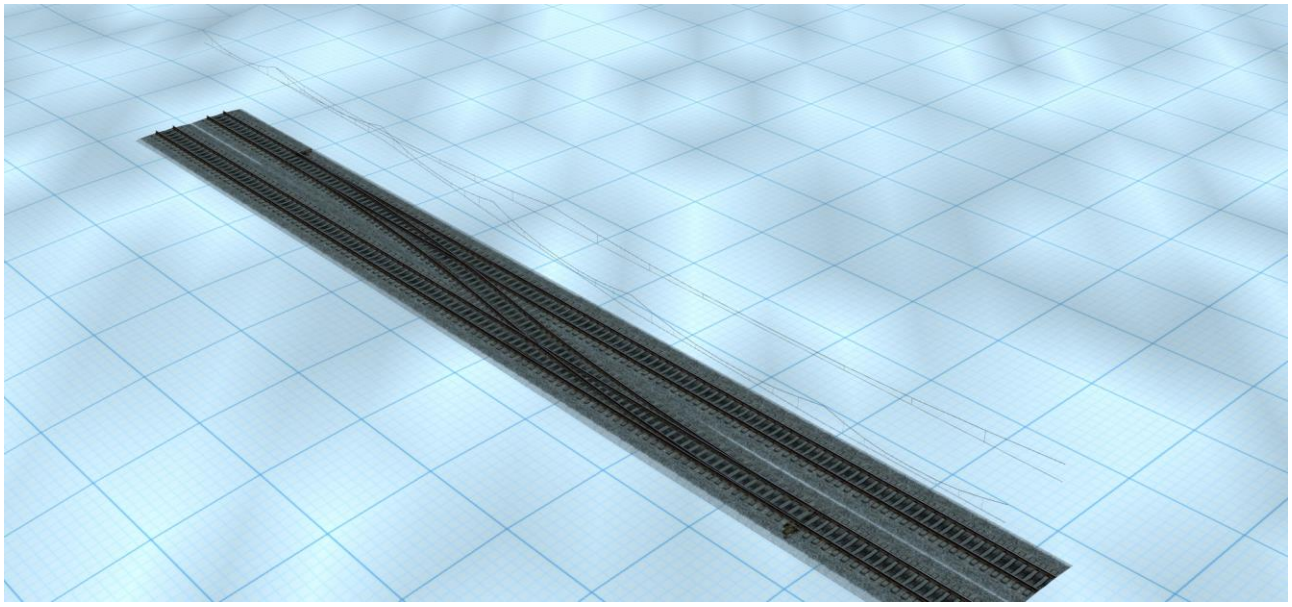
Until now, the problem arose again and again when signals had to be positioned shortly before virtual connections or trains came to a standstill on the last track before a virtual connection, that afterwards an editing of the virtual connection was no longer possible.
This problem has now been solved in Plug-in 1 for EEP 17.1 by clicking again on the menu item "Virtual connection of tracks".



Insertion of a dialogue for tracks in track objects makes it possible to directly address and change individual tracks

This new feature in EEP 17.1 Plug-in 1 makes it possible to edit tracks in track objects individually.
For this purpose, an additional field "Track properties" has been inserted in all track objects.
This allows editing after a click with the right mouse button on the corresponding track.
For example, individual tracks of a track object can be electrified.





News about Lua in EEP

With Plug-in 1 to EEP 17.1 the following new Lua functions are available:

EEPSetZonePos()		**EEPSetZonePos(zone_number, pos_X, pos_Y, pos_Z, radius)**
Parameters	five	`ok = EEPSetZonePos(3, 250, -160, 0, 300)`
Returns	one	
Requires	EEP 17.1 - Plug-in 1	
Purpose	Moves the named weather zone to a new position and/or changes its radius.	
Notes	• First parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window. • Second parameter is the position of the zone centre in X-direction. • Third parameter is the position of the zone centre in X-direction. • Fourth parameter is the position of the zone centre in X-direction. • Fifth parameter is the radius of the weather zone. • **Be aware that** the positioning of the centre of the weather zone may only be done within the system boundaries minus 1 pixel in each direction. The function is not executed beyond this. There is no error message! • Return value is true if the zone exists, false if not.	

EEPGetZonePos ()		**EEPGetZonePos(zone_number)**
Parameters	one	`ok, pos_X, pos_Y, pos_Z, radius = EEPGetZonePos(3)`
Returns	five	
Requires	EEP 17.1 - Plug-in 1	
Purpose	Returns the current position of a weather zone and its radius.	
Notes	• Parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window. • First return value is true when the execution was successful, false if not. • Second return value is the position of the zone centre in X-direction. • Third. return value is the position of the zone centre in Y-direction. • Fourth return value is the position of the zone centre in Z-direction. • Fifth return value is the radius of the zone. • **Be aware that** once the zone has been moved by using *EEPSetZonePos()* you'll have to wait until a new *EEPMain()* cycle has begun to get the new mode thanks to the EEPGetZonePos() function.	

EEPSetZoneWindIntensity()		EEPSetZoneWindIntensity(zone_number, wind_intensity)
Parameters	two	ok = EEPSetZoneWindIntensity(2, 60)
Returns	one	
Requires	EEP 17.1 Plug-in 1	
Purpose	Sets wind intensity in a weather zone between 10 % and 100 % (corresponding to the settings under its properties window).	
Notes	• First parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • Second parameter is wind intensity in percent. If the value is below 10 % then the wind intensity will be reduced to 0. This setting will immediately be applied. • Return value is true when the execution was successful false if not	

EEPGetZoneWindIntensity()		EEPGetZoneWindIntensity(zone_number)
Parameters	one	ok, wind_intensity = EEPGetZoneWindIntensity(2)
Returns	two	
Requires	EEP 17.1 Plug-in 1	
Purpose	Returns wind intensity in a weather zone.	
Notes	• Parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • First return value is true when the execution was successful, false if not. • Second return value is wind intensity in percent. • **Be aware that** once the wind intensity has been set by using *EEPSetZoneWindIntensity()* you'll have to wait until a new *EEPMain()* cycle has begun to get the new intensity thanks to the EEPGetZoneWindIntensity() function.	

EEPSetZoneRainIntensity()		EEPSetZoneRainIntensity(zone_number, rain_intensity)
Parameters	two	ok = EEPSetZoneRainIntensity(1, 50)
Returns	one	
Requires	EEP 17.1 Plug-in 1	
Purpose	Sets rain intensity in a weather zone between 10 % and 100 % (corresponding to the settings under its properties window)	
Notes	• First parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • Second parameter is wind intensity in percent. If the value is below 10 % then the wind intensity will be reduced to 0. The new value is set immediately, but rain and clouds change gradually. • Return value is true when the execution was successful false if not	

EEPGetZoneRainIntensity()		EEPGetZoneRainIntensity(zone_number)
Parameters	one	ok, rain_intensity = EEPGetZoneRainIntensity(1)
Returns	two	
Requires	EEP 17.1 Plug-in 1	
Purpose	Returns rain intensity in a weather zone.	
Notes	• First return value is true when the execution was successful, false if not. • Second return value is rain intensity in percent. • **Be aware that** once the rain intensity has been set by using *EEPSetZoneRainIntensity()* you'll have to wait until a new *EEPMain()* cycle has begun to get the new intensity thanks to the EEPGetZoneRainIntensity() function.	

EEPSetZoneSnowIntensity()		EEPSetZoneSnowIntensity(zone_number, snow_intensity)
Parameters	two	ok = EEPGetZoneSnowIntensity(3, 35)
Returns	one	
Requires	EEP 17.1 Plug-in 1	
Purpose	Sets snow intensity in a weather zone between 10 % and 100 % (corresponding to the settings under its properties window)	
Notes	• First parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • Second parameter is snow intensity in percent. If the value is below 10 % then the snow intensity will be reduced to 0. The new value is set immediately, but snow and clouds change gradually. • Return value is true when the execution was successful false if not	

EEPGetZoneSnowIntensity()		EEPGetZoneSnowIntensity(zone_number)
Parameters	one	ok, snow_intensity = EEPGetZoneSnowIntensity(3)
Returns	two	
Requires	EEP 17.1 Plug-in 1	
Purpose	Returns snow intensity in a weather zone.	
Notes	• Parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • First return value is true when the execution was successful, false if not. • Second return value is snow intensity in percent. • **Be aware that** once the snow intensity has been set by using *EEPSetZoneSnowIntensity()* you'll have to wait until a new *EEPMain()* cycle has begun to get the new intensity thanks to the EEPGetZoneSnowIntensity() function.	

EEPSetZoneHailIntensity()		EEPSetZoneHailIntensity(zone_number, hail_intensity)
Parameters	two	ok = EEPSetZoneHailIntensity(4, 55)
Returns	one	
Requires	EEP 17.1 Plug-in 1	
Purpose	Sets hail intensity in a weather zone between 10 % and 100 % (corresponding to the settings under its properties window)	
Notes	• First parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • Second parameter is hail intensity in percent. If the value is below 10 % then the hail intensity will be reduced to 0. The new value is set immediately, but hail and clouds change gradually. • Return value is true when the execution was successful false if not	

EEPGetZoneHailIntensity()		EEPGetZoneHailIntensity(zone_number)
Parameters	one	ok, hail_intensity = EEPGetZoneHailIntensity(4)
Returns	two	
Requires	EEP 17.1 Plug-in 1	
Purpose	Returns hail intensity in a weather zone.	
Notes	• Parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • First return value is true when the execution was successful, false if not. • Second return value is hail intensity in percent. • **Be aware that** once the hail intensity has been set by using *EEPSetZoneHailIntensity()* you'll have to wait until a new *EEPMain()* cycle has begun to get the new intensity thanks to the EEPGetZoneHailIntensity() function.	

EEPSetZoneFogIntensity()		EEPSetZoneFogIntensity(zone_number, fog_intensity)
Parameters	two	ok = EEPSetZoneFogIntensity(2, 40)
Returns	one	
Requires	EEP 17.1 Plug-in 1	
Purpose	Sets fog intensity in a weather zone between 10 % and 100 % (corresponding to the settings under its properties window)	
Notes	• First parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • Second parameter is fog intensity in percent. If the value is below 10 % then the fog intensity will be reduced to 0. This setting will immediately be applied. • Return value is true when the execution was successful false if not	

EEPGetZoneFogIntensity()	EEPGetZoneFogIntensity(zone_number)
Parameters	one
Returns	two
Requires	EEP 17.1 Plug-in 1
Purpose	Returns fog intensity in a weather zone
Notes	• Parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window • First return value is true when the execution was successful, false if not. • Second return value is fog intensity in percent. • **Be aware that** once the fog intensity has been set by using *EEPSetZoneFogIntensity()* you'll have to wait until a new *EEPMain()* cycle has begun to get the new intensity thanks to the EEPGetZoneFogIntensity() function.

```
ok, fog_intensity = EEPGetZoneFogIntensity(2)
```

EEPSetZoneClouds ()	EEPSetZoneClouds(zone_number, mode)
Parameters	two
Returns	one
Requires	EEP 17.1 - Plug-in 1
Purpose	Defines if there are clouds in a weather zone and what type they are.
Notes	• First parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window. • Second parameter is the clouds mode of the weather zone: 0 = No clouds, 1 = Clouds, 2 = Dark clouds. • Return value is true when the execution was successful, false if not.

```
ok = EEPSetZoneClouds(1, 2)
```

EEPGetZoneClouds()	EEPGetZoneClouds(zone_number)
Parameters	one
Returns	two
Requires	EEP 17.1 - Plug-in 1
Purpose	Returns whether there are clouds in a weather zone and what kind they are.
Notes	• Parameter is the number of the existing weather zone as a numeric number. It is in the top line of its properties window. • Second return value is the clouds mode of the weather zone: 0 = No clouds, 1 = Clouds, 2 = Dark clouds. • Return value is true when the execution was successful, false if not. • **Be aware that** once the mode has been set by using *EEPSetZoneClouds()* you'll have to wait until a new *EEPMain()* cycle has begun to get the new mode thanks to the EEPGetZoneClouds() function.

```
ok, mode = EEPGetZoneClouds(1)
```

EEPGetCloudsMode()		EEPGetCloudsMode()
Parameters	none	`mode = EEPGetCloudsMode()`
Returns	one	
Requires	EEP 17.1 - Plug-in 1	
Purpose	Returns whether there are clouds in the sky globally (outside any weather zones) and what kind they are.	
Notes	• This function is called by EEP without any parameters. • Return value is the global clouds mode: 0 = No clouds, 1 = Clouds, 2 = Dark clouds. • **Be aware that** once the mode has been set by using *EEPSetCloudsIntensity()* respectively. *EEPSetDarkCloudsIntensity()* you'll have to wait until a new *EEPMain()* cycle has begun to get the new mode thanks to the EEPGetCloudsMode() function.	

EEPSignalSetTagText()		EEPSignalSetTagText("Name","Text")
Parameters	two	`ok = EEPSignalSetTagText(87, "Route_C")`
Returns	one	
Requires	EEP 17.1 - Plug-in 1	
Purpose	Changes the tag-text of a signal. Each signal can carry an arbitrary string of maximal 1024 characters. These strings are saved and loaded with the layout. As these strings are individually assigned to each signal, they won't get lost.	
Notes	• First parameter is the signal's ID. • Second parameter is the desired text to be saved. • Return value is true when the execution was successful, false if not.	

EEPSignalGetTagText()		EEPSignalGetTagText("Name")
Parameters	one	`ok, Text = EEPSignalGetTagText(87)`
Returns	two	
Requires	EEP 17.1 - Plug-in 1	
Purpose	Returns the tag text of a signal. By means of tag texts, e.g. information on preset routes or level crossings can be stored directly in the signals instead of in data slots.	
Notes	• The parameter is the signal's ID. • First return value is true when the execution was successful, false if not. • Second return value is the tag text that was assigned to the signal. • **Be aware that:** The EEPSignalGetTagText() function will only return the new state of a signal set by using *EEPSignalSetTagText()* once a new *EEPMain()* cycle has begun.	

EEPGetAnlName()		EEPGetAnlName()
Parameter	none	`Name = EEPGetAnlName()`
Rückgabewerte	one	
Voraussetzung	EEP 17.1 - Plug-in 1	`print(Welcome to the ", Name, "layout")`
Zweck	Returns the name with which the layout was last saved.	
Bemerkungen	• This function is called by EEP without any parameters. • Return value is the name of the layout as string.	

Zudem wurde mit dem Plug-in1 zu EEP 17.1 der mit jeder neuen Anlage produzierte Startcode in der Datei `Anlagenname.lua` auf das unbedingt Notwendige reduziert:

```
1. clearlog()
2.
3. function EEPMain()
4.   return 1
5. end
```

In the future, the font size of text textures can be scaled higher than previously possible

With an increase of the font size, further effects are possible in EEP, where text textures can be used.

Optimisation of the use of the graphics card when displaying in the 3D window

With plug-in 1 to EEP 17.1, the display in the 3D window has been optimised so that the preview window is supported slightly less by the graphics card, giving the 3D window full performance. This has a positive effect on the performance of weaker computers.

Other improvements in the system

In the nostructor - with transfer to EEP - it is now possible to set the couplings separately for vehicles, for example to set the coupling at the front of the tractor and at the rear of the trailer to "push off" and at the rear of the tractor and at the front of the trailer to "connect".

In the block function, the float precision was increased (more decimal places when calculating blocks) in order to be able to use blocks more precisely in EEP 17.

Closing remarks

Many wishes of experienced EEP users and designers have been incorporated into the development of **Plug-In 1 for EEP 17.1**. The innovations facilitate both the construction and the operation of your layouts. The Lua extensions also open up completely new possibilities to automate your layouts "intelligently".

We hope you enjoy this first plug-in to EEP 17.1.

Your EEP-Team of Trend Verlag