

Manual to Plug-in 2 for EEP 17.2

Dear customers, first of all we would like to heart-fully thank you for purchasing EEP 17.2 Plug-in 2.

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:

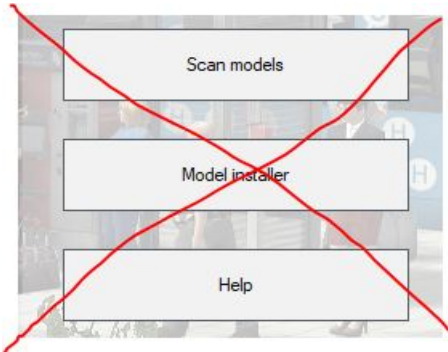
| | |
|---|----|
| Glossary | 1 |
| Installation instructions | 2 |
| New models contained in Plug-in 2 for EEP 17.2 | 3 |
| New features allowed by Plug-in 2 for EEP 17.2 : | 7 |
| • Additional customisable light source for real estate and landscape elements | 7 |
| • New 'Align to ground' option for animated landscape elements. | 13 |
| • New contact point settings for landscape elements or real estate structures | 14 |
| • The 3D model properties window now displays the model's registration path. | 17 |
| • Leaf fall adapts to the season..... | 17 |
| • Increased randomness of air displacement effects on vegetation when trains pass by. --- | 20 |
| • Lighting on and off adjusted to sunrise and sunset times | 21 |
| • Menu for controlling visual effects and adding tag text | 21 |
| • Dynamic sound management based on the camera position..... | 24 |
| • Storing vehicle axle settings even after deletion..... | 24 |
| • Display of signal and turnout numbers using a combination of keys..... | 25 |
| • New function for multiple selection of blocks in 2D mode using a shortcut..... | 27 |
| • Improving the distinction of the Gizmo at the end of the tracks..... | 28 |
| • Differentiated colours to distinguish invisible splines in 3D mode. | 29 |
| • Simplified exchange of tracks of the same type in the 3D editor. | 29 |
| • Improved representation of the connection lock at the end of a track. | 30 |
| • Optimising loading and saving times in EEP | 31 |
| • Two features that were not yet documented in EEP. | 31 |
| • Lua offers new EEP-specific functions | 34 |
| Closing remarks | 47 |

Manual to Plug-in 2 for EEP 17.2

Installation instructions

Please make sure that update #2 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.

v.: 17.2 Expert (x64),

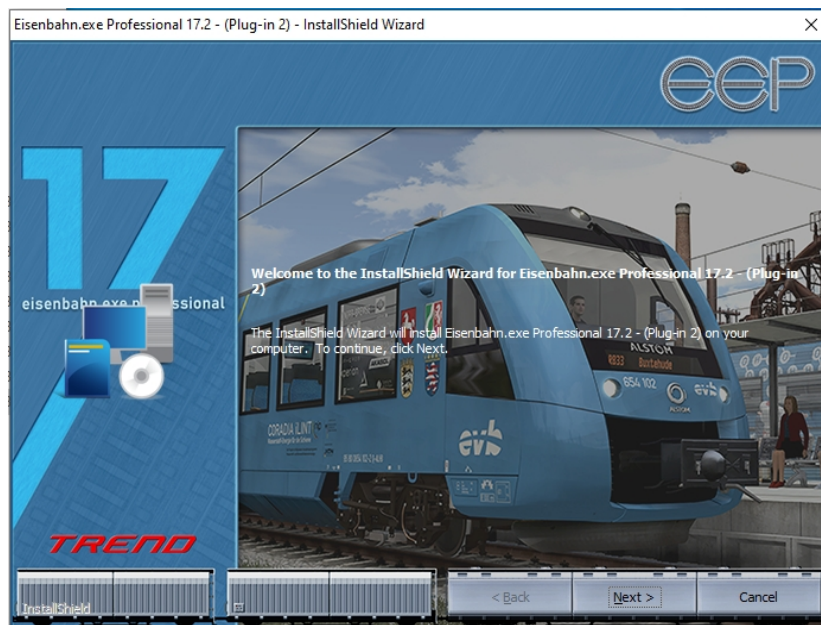


The Plug-in needs its own installer to upgrade, since it brings new features besides the new models. For this reason it is not possible to open this file directly via the "Model-Installer of EEP 17.2.

If necessary, please leave your EEP 17.2!

Please start installing the Plug-in by double clicking on the file **V17TSP10059**.

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 17.2



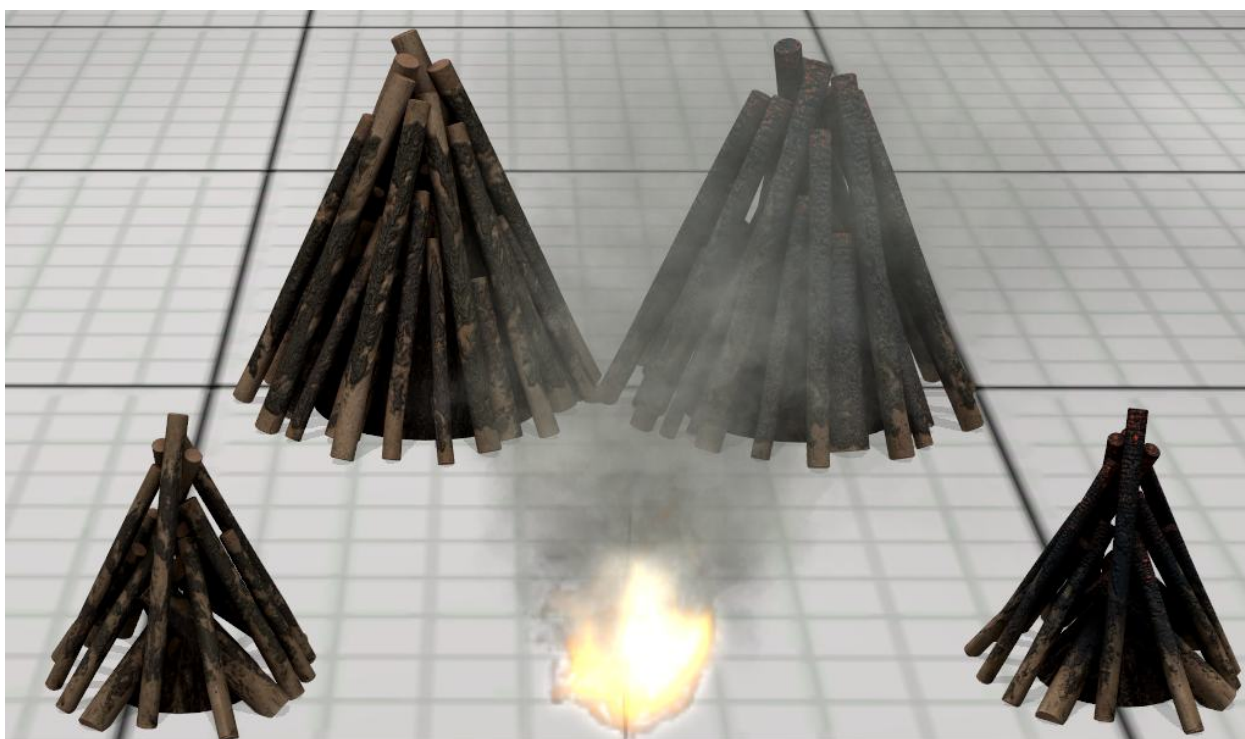
If the installation was successful, then you will see in the lower left part of the launch window the EEP 17.2 version followed by the mention „Plugins: 2“. (Here 1,2 since I have both Plug-in 1 and Plug-in 2 installed)

Manual to Plug-in 2 for EEP 17.2



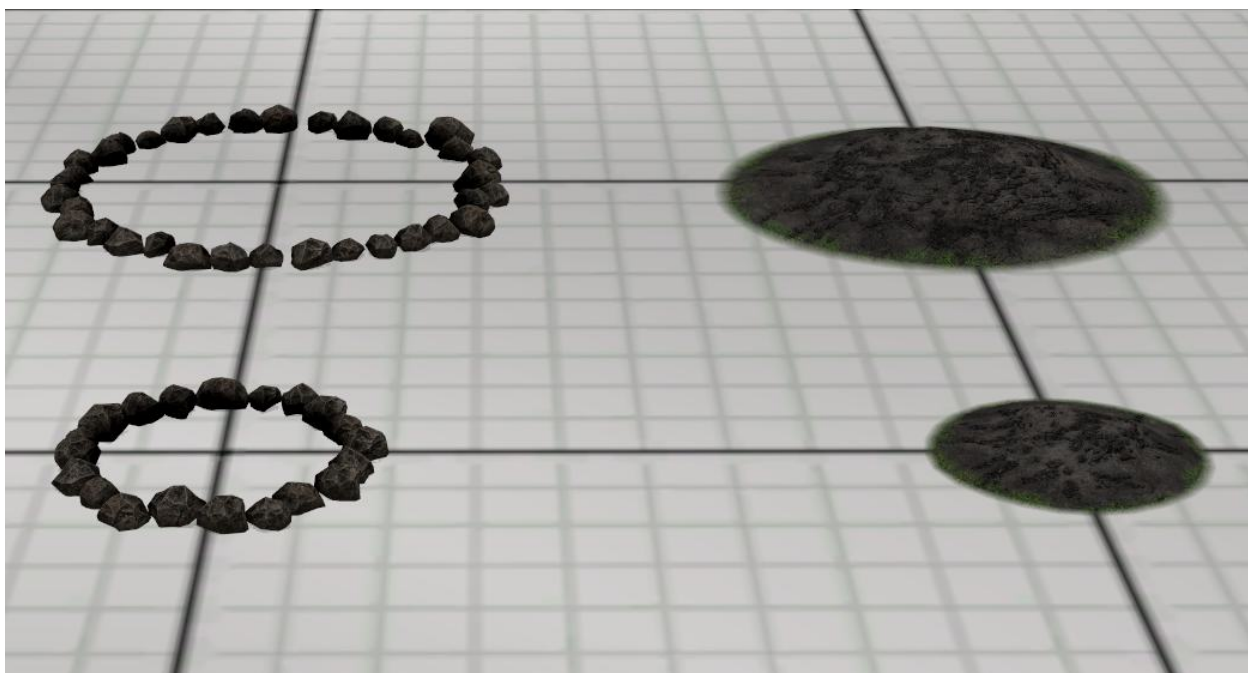
New models contained in Plug-in 2 for EEP 17.2 :

4 different campfires with fireplaces.



Manual to Plug-in 2 for EEP 17.2

2 stone circles and 2 ash heaps



4 pieces of charred wood of different lengths

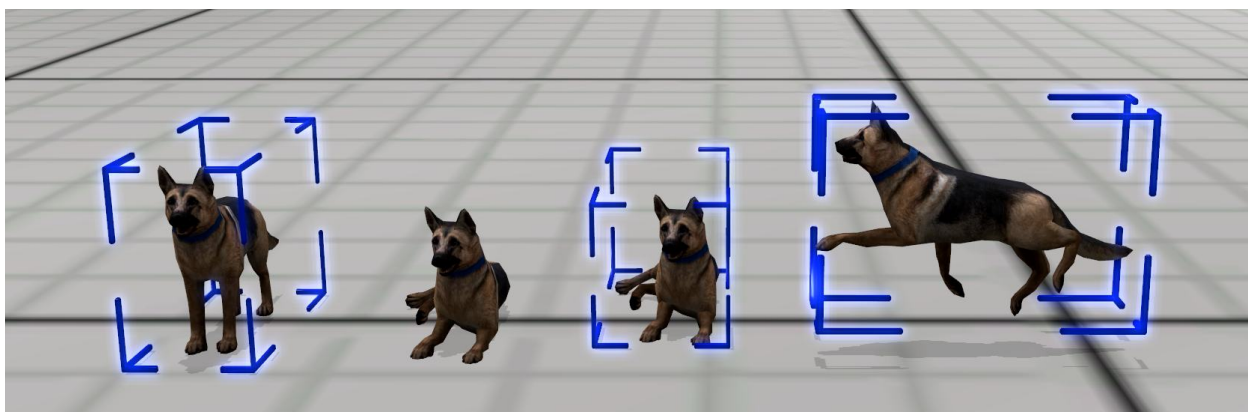


Manual to Plug-in 2 for EEP 17.2

3 animated men in different poses

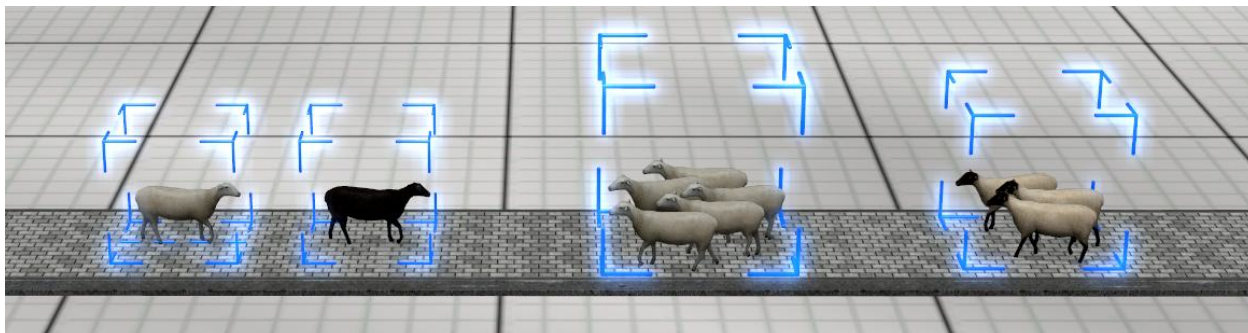


4 animated German shepherds with different poses

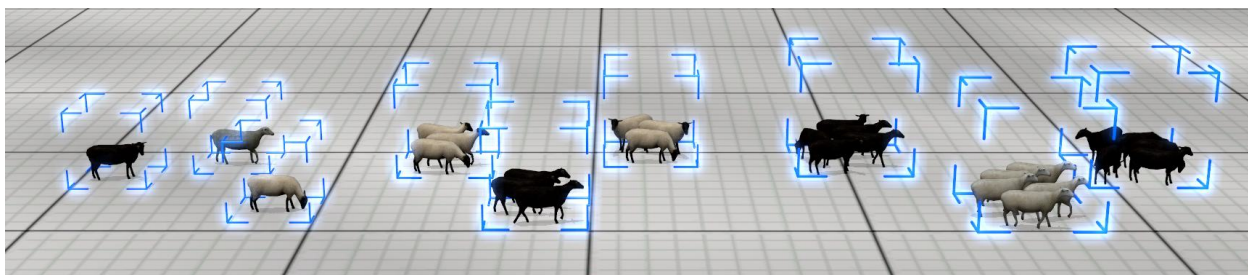


Manual to Plug-in 2 for EEP 17.2

4 sheep on their own or in groups as rolling stock



9 sheep, alone or in groups, as animated landscape elements.



EEP 17

Plug-in 2

Manual to Plug-in 2 for EEP 17.2

New features allowed by plug-in 2 for EEP 17.2 :

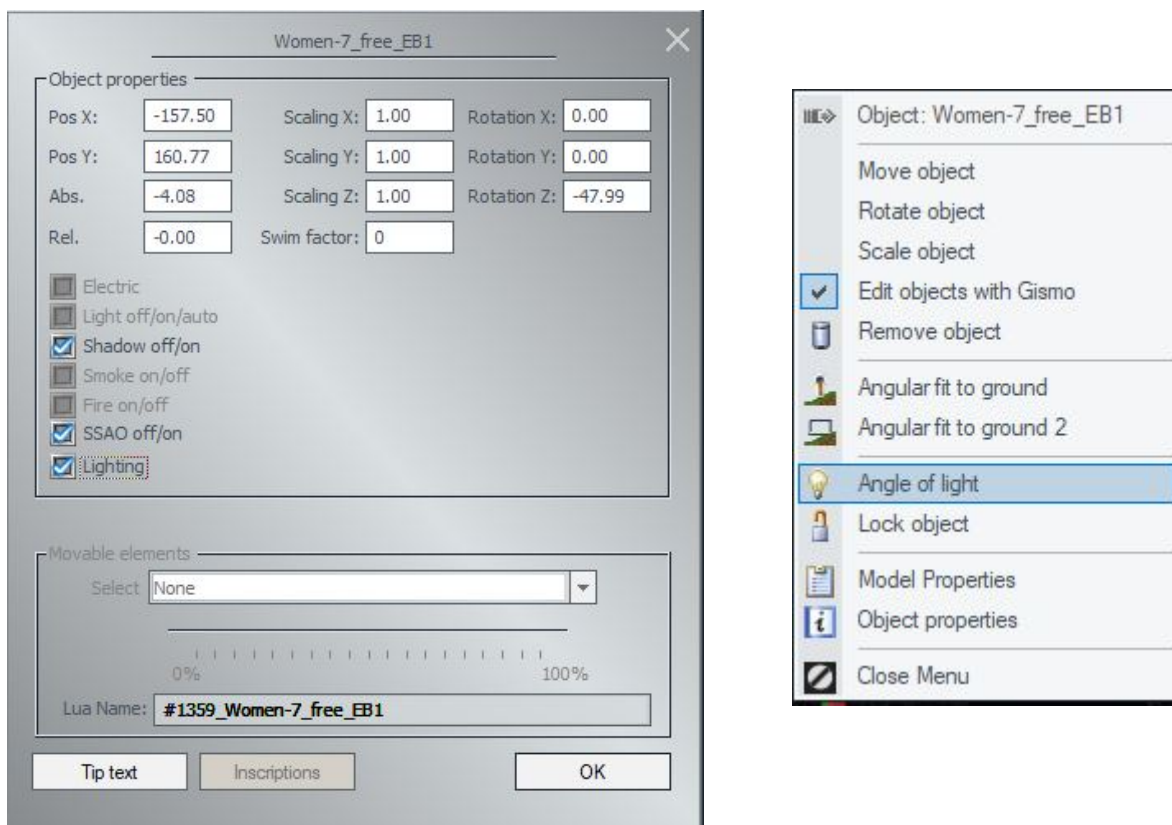
Additional customisable light source for real estate and landscape elements:

The plug-in 2 for EEP 17.2 allows the individualized lighting of real estate structures and landscape elements.

To do this, an additional light source specific to the real estate structure or landscape element can be activated.

Here's how to proceed:

In 3D display mode, bring up the context menu of the real estate structure or landscape element and open the object's properties dialog box.



Check the "Lighting" box in the object properties.

Then, right-click on the object again to bring up the context menu, but this time select the "Angle of light" entry.

Manual to Plug-in 2 for EEP 17.2

Using the gizmo that appears, change the angle of this new object-specific light source. The yellow arrow represents the angle of the light.



The results obtained allow for much more realistic lighting in nighttime scenes. This enables you to adjust the lighting on characters standing under a street lamp, correct the lack of lighting on building facades, trees, etc., even if they are supposed to be affected by a light source.

It should be noted that while this additional lighting feature is available for most landscape elements or buildings, some models, often older ones created by certain designers, may have had the "lighting" function disabled during their design process. These specific models will not be able to benefit from this additional lighting feature.

Below, we show you some examples of lighting corrections.

Manual to Plug-in 2 for EEP 17.2



- with additional lighting -



- without additional lighting -

Manual to Plug-in 2 for EEP 17.2



- with additional lighting -



- without additional lighting -

Manual to Plug-in 2 for EEP 17.2



- with additional lighting -



- without additional lighting -

Manual to Plug-in 2 for EEP 17.2



- with additional lighting -



- without additional lighting -

Manual to Plug-in 2 for EEP 17.2

New 'Align to ground' option for animated landscape elements.

Right-click on the desired animated landscape element and select "object properties" from the context menu.



Above « automatic avoidance », you'll find a new option which, when ticked, allows animated landscape elements to automatically adjust their height and inclination to the terrain encountered (see cows in background).



Manual to Plug-in 2 for EEP 17.2



Here without the option activated. The cows in the background sink into the relief until they disappear.

New contact point settings for landscape elements or real estate structures.

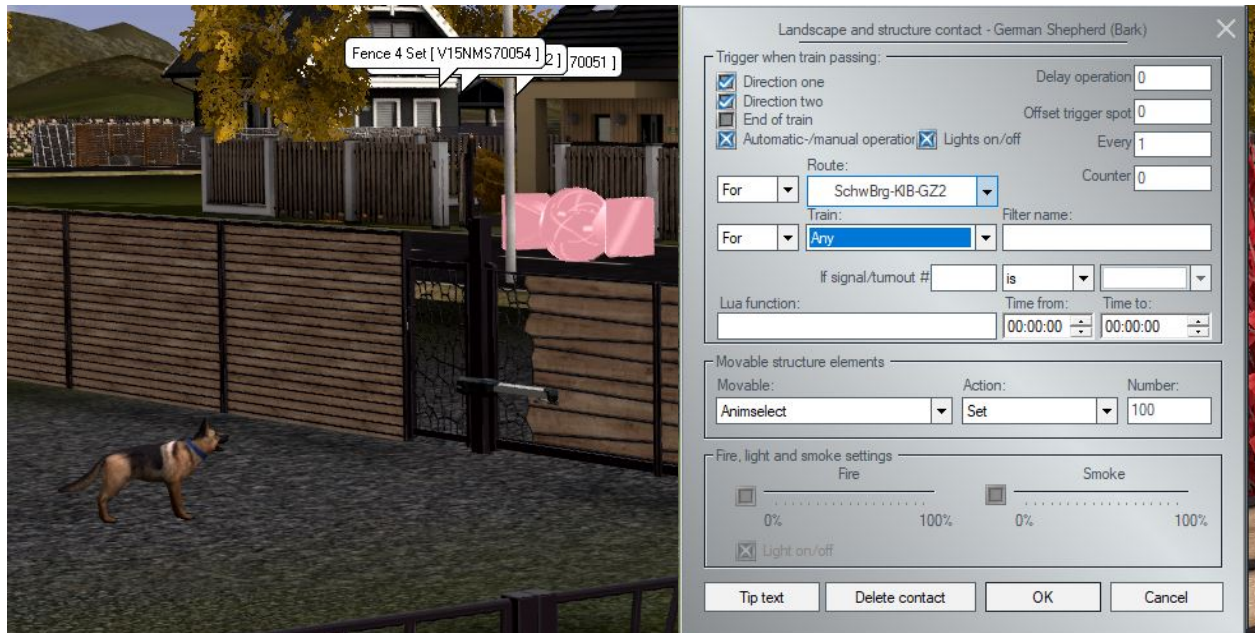
Now, it is possible to define the exact position of the axis of a real estate structure or landscape element by triggering a contact point, rather than transitioning from position A to position B as before. This is particularly useful for 3D models with different animations where you would like to switch immediately from one specific animation to another.

You will now be able to modify the position of the German Shepherd dog provided with the plug-in, which will start barking when the DHL delivery employee approaches.

The German Shepherd dog (Barking) has two positions:

- 0% = lying down (1st position)
- From 1% to 99%, it starts barking while still lying down (sound animation)
- 100% = it stands up and barks (2nd position)

Manual to Plug-in 2 for EEP 17.2



In 3D mode, right-click on the desired landscape element or real estate structure, then select "Structure and LS element contact."

In 2D mode, activate the signal editor and in the contact menu, select "Structure or LS elements." Then, click on the landscape element or real estate structure to place a contact point.

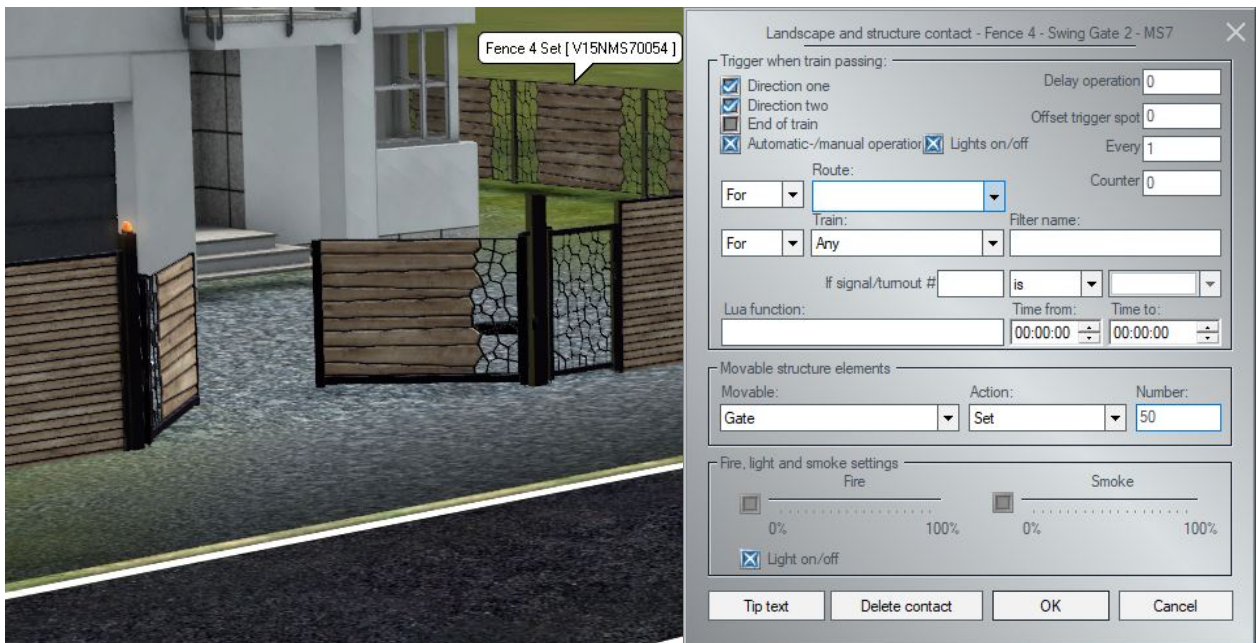


- before triggering the contact point -

Manual to Plug-in 2 for EEP 17.2



- after triggering the contact point -



When the supra contact point is activated, the gate opens directly at 50%; there is no transition from 0 to 50%.

Manual to Plug-in 2 for EEP 17.2

The 3D model properties window now displays the model's registration path

The header of the 3D model properties dialogue box now displays the storage path of the corresponding 3dm file.



Leaf fall adapts to the season:

In the landscape elements => Flora => Other => 'Flying Leaves' with a range of 50 m, changes have been made to the shape and colour of the leaves to reflect the seasons.

Leaves will have colours appropriate to the season (spring, summer and autumn) and in winter will be replaced by snowflakes. This feature adds an extra touch of realism.

Manual to Plug-in 2 for EEP 17.2



– spring –



– summer –



-- autumn --

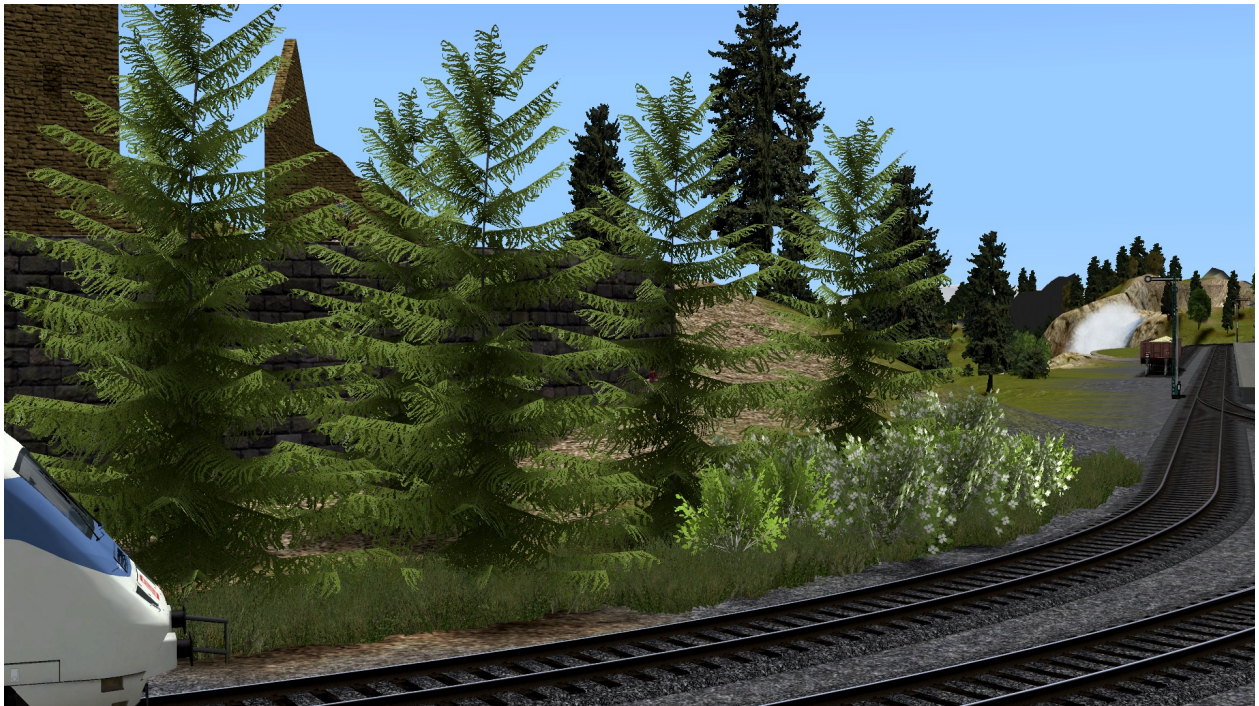


-- winter --

Manual to Plug-in 2 for EEP 17.2

Increased randomness of air displacement effects on vegetation when trains pass by.

Train passages generate an air displacement that influences the movement of the surrounding vegetation.



In order to further enhance the realism achieved so far, it has been decided to increase the randomness of these air displacement effects.

Manual to Plug-in 2 for EEP 17.2

Lighting on and off adjusted to sunrise and sunset times.

Another point for a better representation of the seasons in EEP is an adapted switching on of the lighting at sunset and switching off at dawn. (Later in summer, earlier in winter).



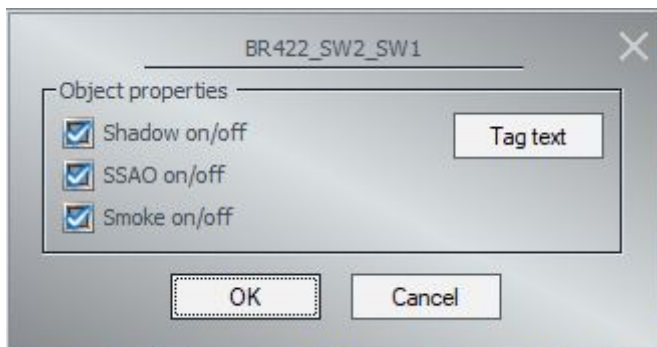
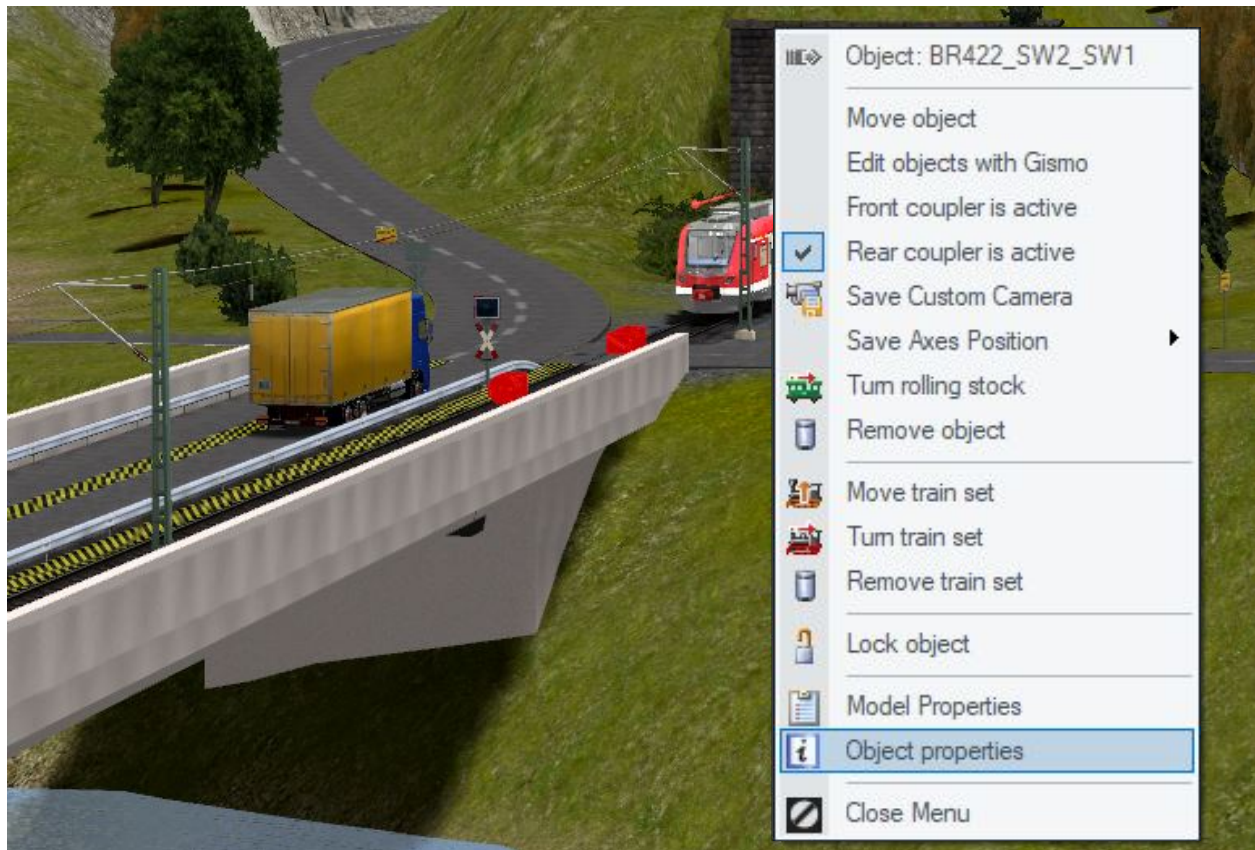
Menu for controlling visual effects and adding tag text.

The plug-in 2 for EEP 17.2 provides you with a new menu that allows you to enable or disable smoke, shadows, and SSAO (Screen Space Ambient Occlusion) effect for all rolling stock, both in 2D mode and 3D mode. You can also add a tag text (up to 1024 characters) to each rolling stock.

Here's how to proceed:

In 3D mode, select the rolling stock, then right-click to bring up the context menu and open the object properties dialog box.

Manual to Plug-in 2 for EEP 17.2

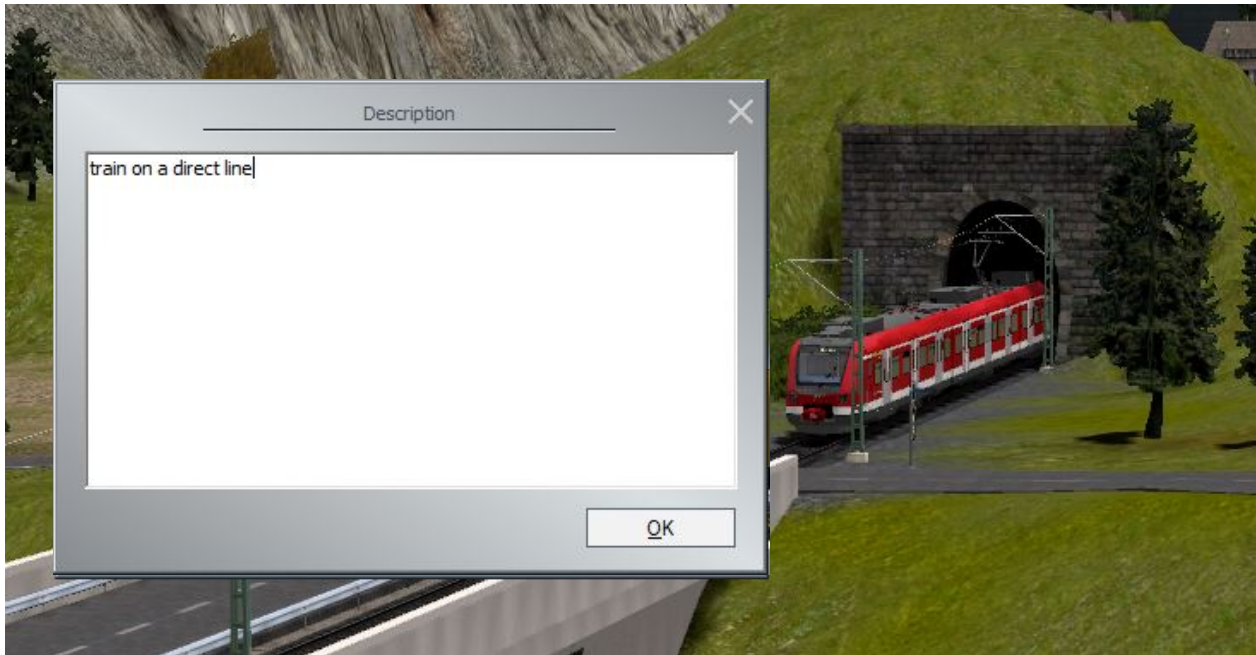


By clicking on "Tag text" a new window opens where you can define a specific tag for the selected rolling stock.

This tag can later be queried using the following Lua function :

```
EEPRollingstockGetTagText("Name").
```


Manual to Plug-in 2 for EEP 17.2



In 2D mode, you can bring up the same menu. To do this, first make sure that the "System overview" tab is selected. Then, right-click on the rolling stock to bring up the context menu and launch the "Object Properties" dialog box.



Manual to Plug-in 2 for EEP 17.2

Dynamic sound management based on the camera position.

This new feature was developed in response to user feedback, as they wanted the sound of subway trains running beneath the surface of their layout to be audible.

With Plug-in 2 for EEP 17.2, if the active camera is positioned below the surface, only the sound of rolling stock below the surface will be audible. Conversely, if the active camera is on the surface, only the sound of rolling stock on the surface will be audible.



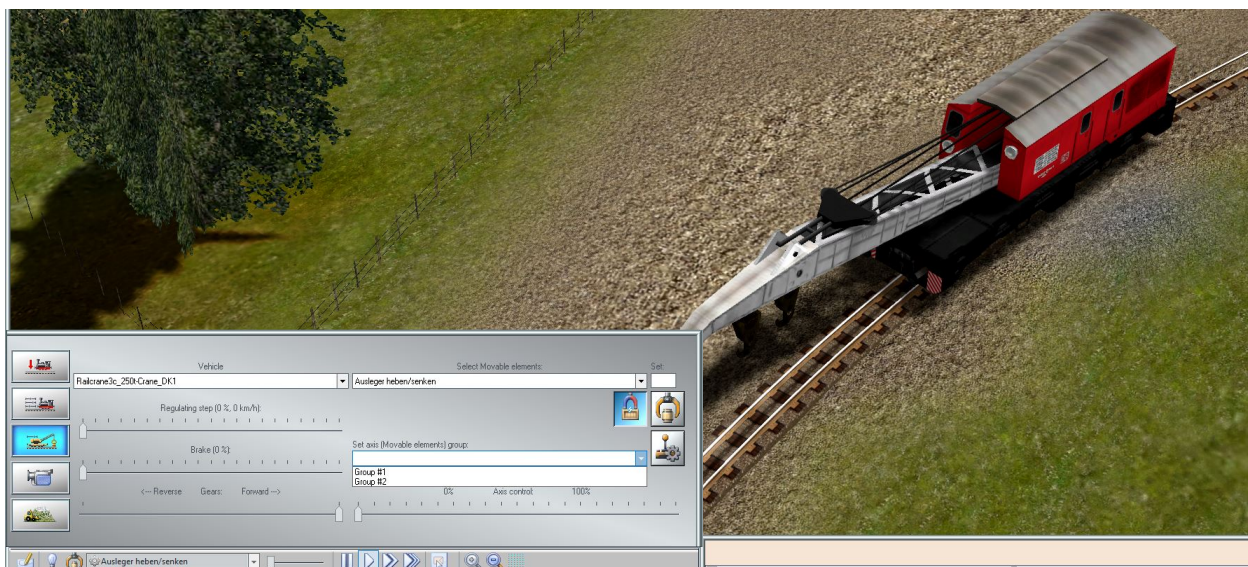
Storing vehicle axle settings even after deletion.

EEP17 makes it possible to store the settings of the various axles making up a piece of rolling stock directly in a group of axles called « Save axes position ».

However, the settings contained in these groups were not saved when the rolling stock to which they were attached was removed from the layout.

From now on, cancelling the deletion of the rolling stock using the 'Undo' function will allow the rolling stock and the various settings saved in the « Save axes position » groups to be recovered.

Manual to Plug-in 2 for EEP 17.2



Display of signal and turnout numbers using a combination of keys

When you click on a signal or a turnout while holding down the left mouse button and the right ALT key [ALTGR], the signal or turnout displays its respective number, here ID=0053 (in game mode). This is extremely useful when you want to change the position of the turnout or signal using the shortcut [Alt GR + ID].

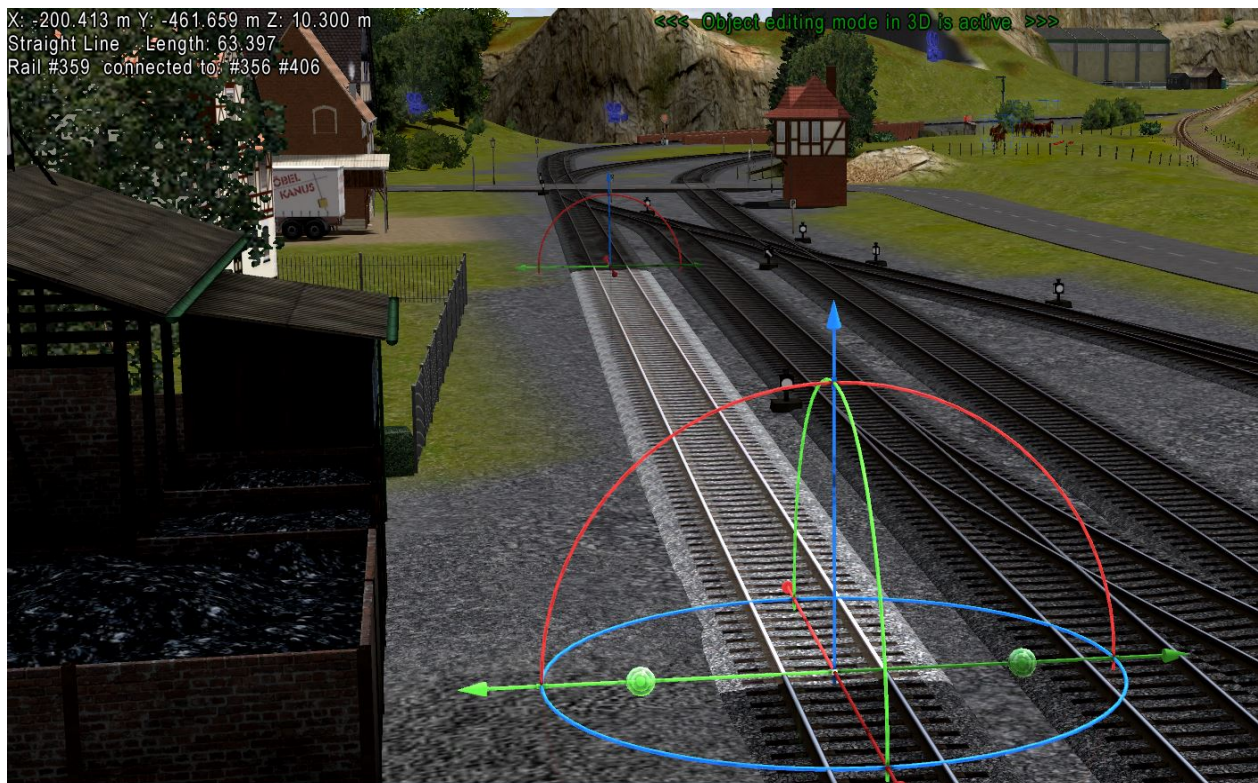


This feature is also available when the control window is active (3D editing mode) and you click on the signal or turnout. In addition to the number, additional information such as the x, y, and z values, the ID of the track on which the turnout or signal is located, the offset from the beginning of the track, and the signal number are also displayed. This allows for more precise identification and tracking of signals and turnouts on the layout.

Manual to Plug-in 2 for EEP 17.2



The rails indicate their location, type and length, as well as the IDs of the connected rails.



Manual to Plug-in 2 for EEP 17.2

New function for multiple selection of blocks in 2D mode using the CTRL + A key combination

A new feature is now available in 2D mode to easily select blocks for later modification (save, copy, delete, move). The CTRL + A functionality, known in Windows for selecting all, can now be used in EEP to select all elements of the same category displayed in the current 2D window. However, this selection is limited to 1000 objects. If exceeded, an error message will appear, and it will be necessary to resize the 2D window accordingly.

This feature simplifies the management and manipulation of multiple blocks in a single operation in the virtual environment.

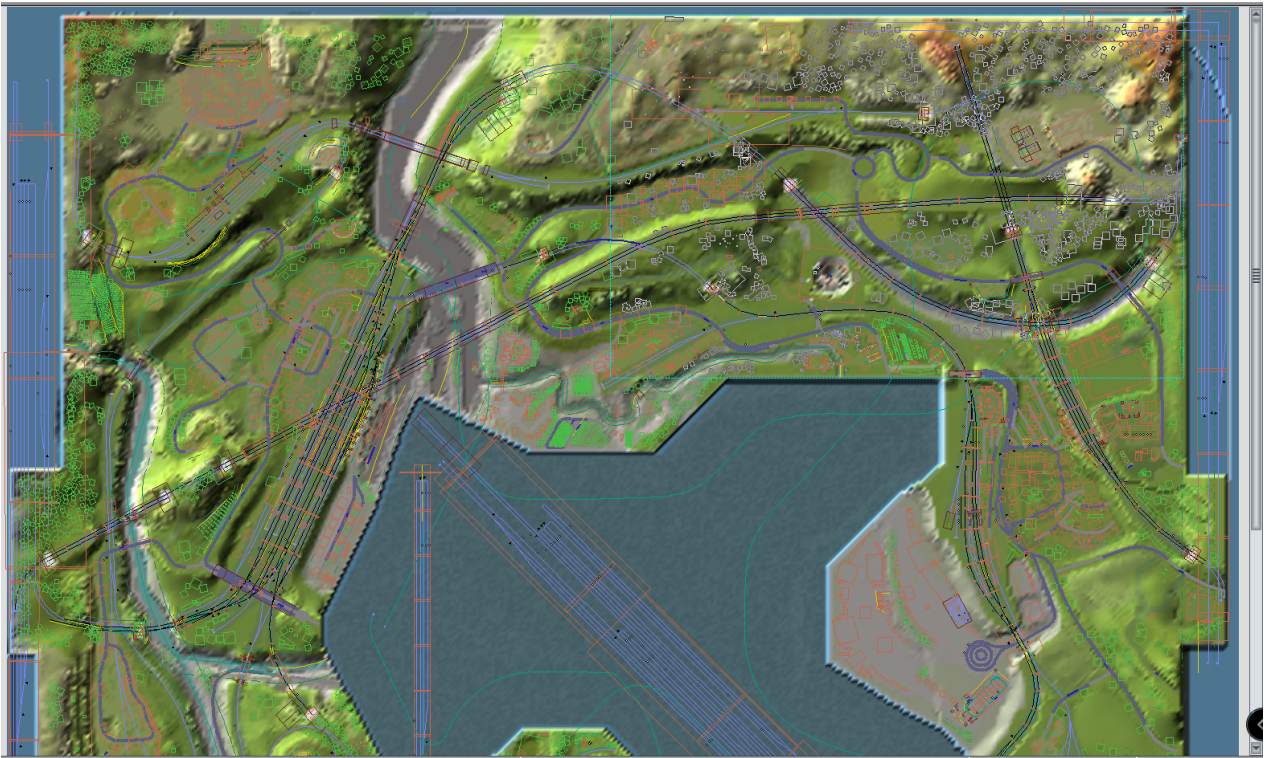


In the example above, we activated the landscape editor and selected all corresponding models in the displayed area.

A simple Ctrl + A was enough. Now you can save the block, move it, duplicate it, delete it, or perform other actions.

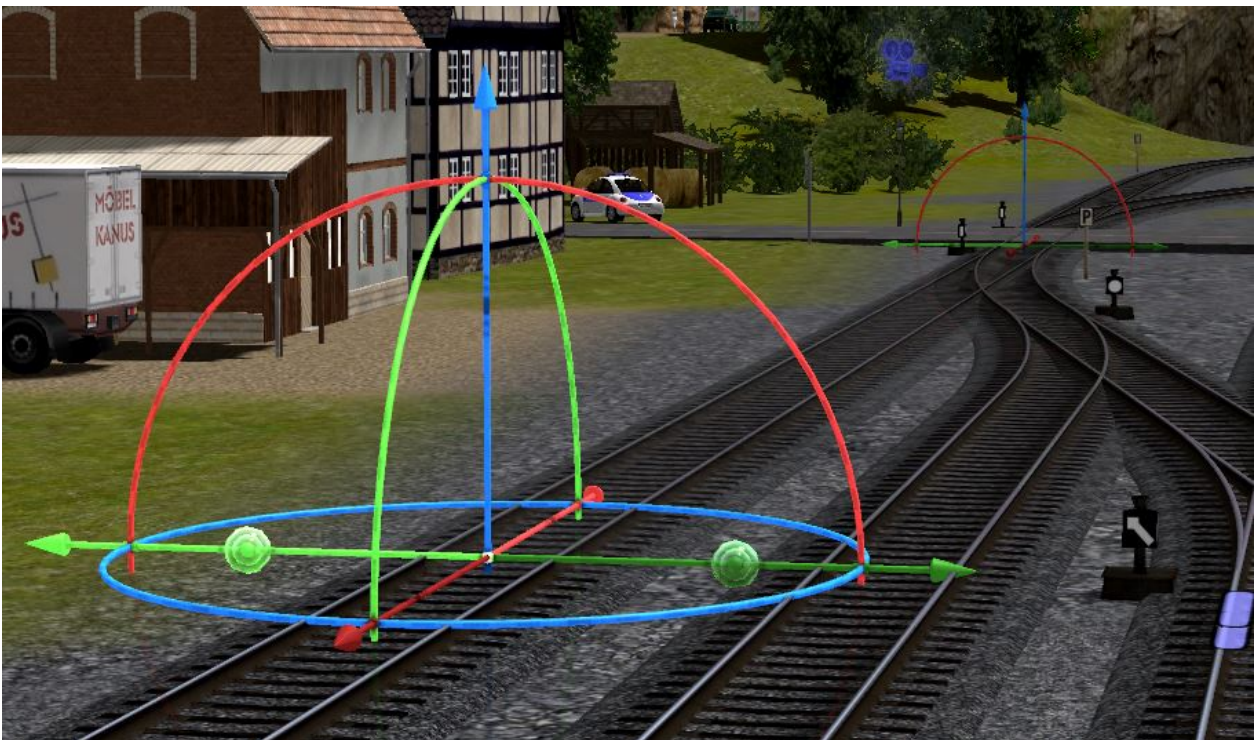
This functionality can also be used to save road layouts, railway tracks, and more. Just activate the corresponding editor and apply the Ctrl + A shortcut.

Manual to Plug-in 2 for EEP 17.2



Zooming out reveals the selected elements, which differ in colour.

Improving the distinction of the Gizmo at the end of the tracks



In response to user demand, this feature makes it easier to differentiate between the Gizmo at the start of a spline and the Gizmo at the end of the spline.

Manual to Plug-in 2 for EEP 17.2

With EEP 17.2 Plugin 2, the Gizmo at the end of the spline has been reduced in size and its colour has been slightly darkened. This makes it easier to see and distinguish the start and end points of splines in edit mode.

Differentiated colours to distinguish invisible splines in 3D mode

For greater visual distinction, Plug-in 2 has introduced individual colouring for invisible splines to make them easier to identify in the 3D environment.



Yellow/black for railway tracks.

Green/black for roadways.

Orange/black for tramways.

Blue/black for waterways.

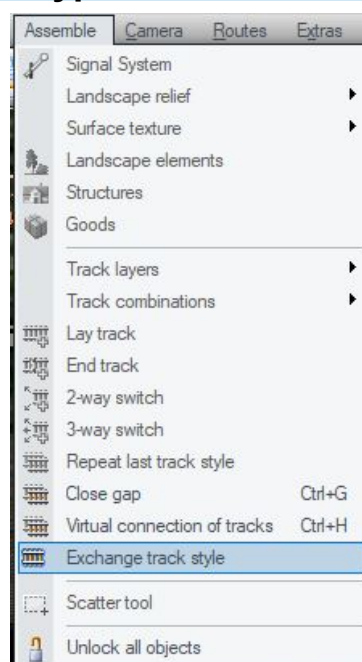
This makes it easier to use these different splines, without confusing them.

Simplified exchange of tracks of the same type in the 3D editor.

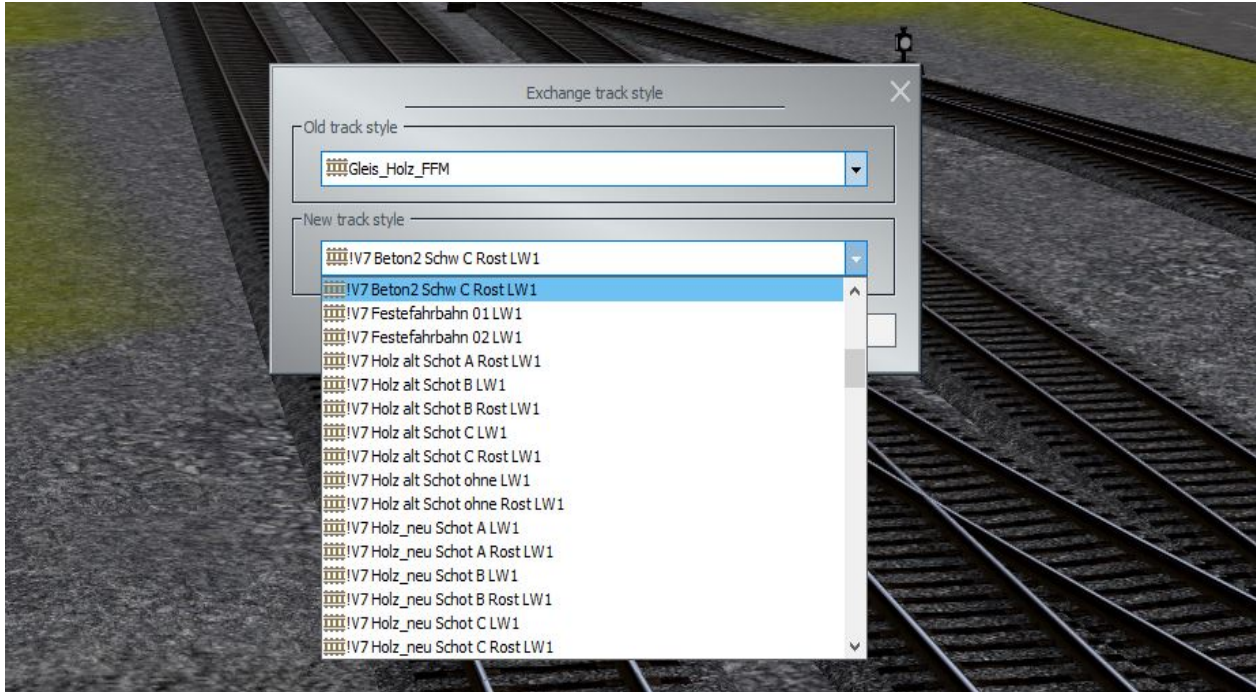
With Plugin 2 for EEP 17, the process of exchanging track styles within the same layer has been further simplified. Now, you can simply select the track you want to exchange with a single click and go to the "Assemble" menu to choose the "Exchange Track Style" function.

The selected track style will be displayed in the upper part of the dialog box, and a dropdown menu will list the other available track styles in the same category.

Confirm your selection by clicking on "Exchange" to apply the new track style.



Manual to Plug-in 2 for EEP 17.2



Improved representation of the connection lock at the end of a track.

In response to user feedback, the representation of the locking of a track connection has been improved. The locking elements have been reduced in size and made more transparent, making it easier to distinguish the exact end of the track. This enhances the precise identification of track endpoints and improves the visualization of locked connections.



Manual to Plug-in 2 for EEP 17.2

Optimising loading and saving times in EEP

Various elements of the program have been optimized to improve loading and saving times in EEP. These optimizations apply to the rendering of 3D models, the functioning of 2D mode, sound processing, and other operations. However, the speed improvements may vary depending on the models used, the size of the installation, and the computer's processing power. These optimizations aim to provide a smoother and more responsive experience when using EEP, reducing waiting times and enhancing the overall efficiency of the software.

Two features that were not yet documented in EEP.

These are features that already exist but have not yet been documented in EEP. These are no new features introduced with the EEP 17 Plugin 2.

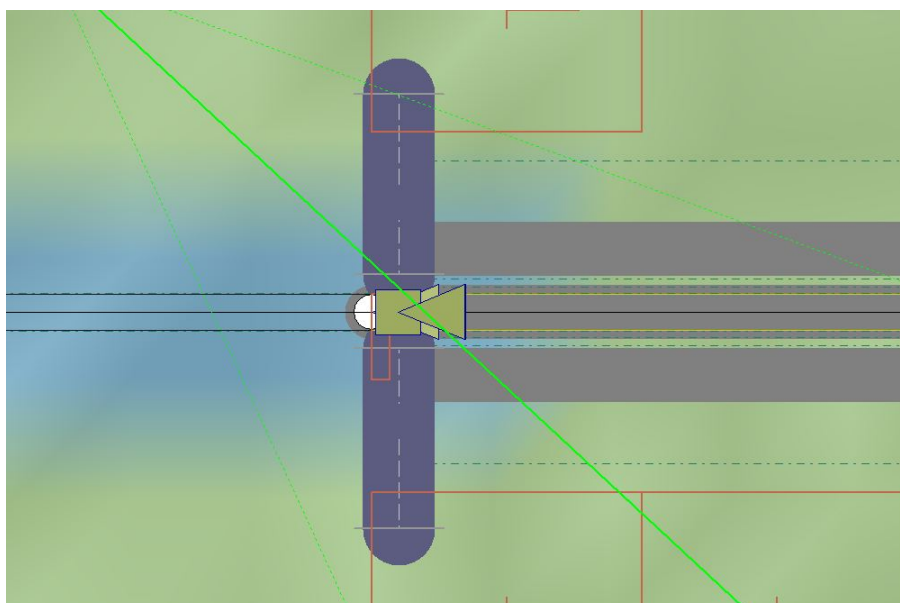
That's why we're using this manual to introduce them and explain how they work.

Switching from one type of track to another using invisible train depots

The two types of tracks with different characteristics are overlapped in such a way that their respective lengths are sufficient to accommodate the entire vehicle or train. This ensures that the individual components of the rolling stock are not compressed upon entry or exit from the depot.

The two depots, the entry depot and the exit depot, are placed side by side at the same location.

In EEP, there has been a frequent request to be able to combine different types of tracks. This capability exists by using "intercalated" invisible depots.



Manual to Plug-in 2 for EEP 17.2



Finally, the rolling stock entered one of the depots and "disappeared" for a short while.

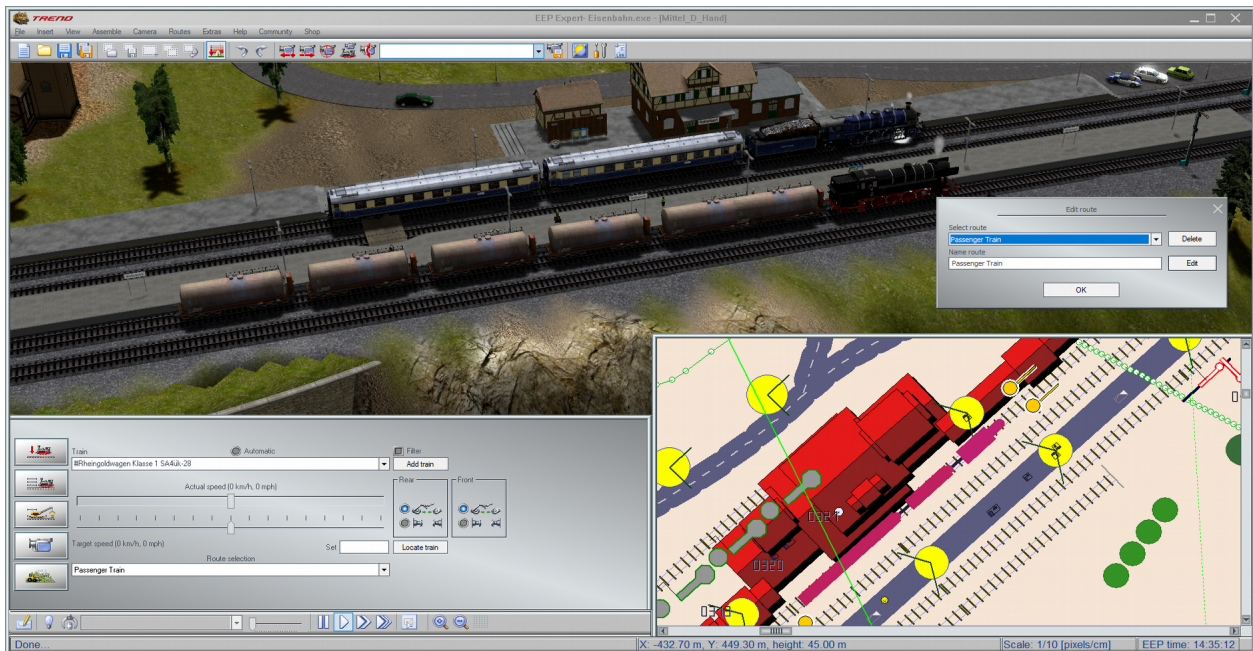


and reappears in the same place as it leaves the second depot to continue its journey.

Manual to Plug-in 2 for EEP 17.2

Filter trains according to the route assigned to them:

Another previously undocumented feature is the ability to filter the displayed trains based on the assigned route. This feature allows you to selectively show or hide trains based on the specific routes they are assigned to.



Prerequisites:

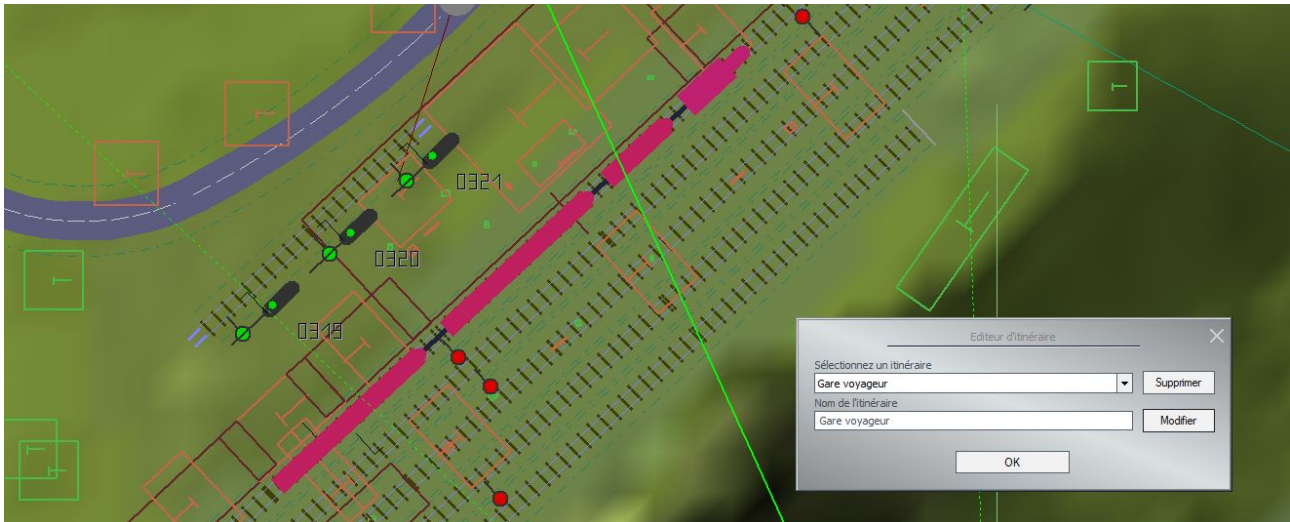
- Have created routes
(Route Menu => Route Editor, give a name, then click on "Add")
- Assign routes to trains (or rolling stock)
(In the command window, under the dropdown "Route Selection")

How it works:

To only display trains in the radar window that have been assigned a specific route, go to the "Routes" menu and select the Route Editor. In the dialog box that opens, select the route you want to filter and click OK.

The radar window will now only display trains or rolling stock that have a matching destination.

Manual to Plug-in 2 for EEP 17.2



The same result can be achieved in 2D editing mode using the same procedure.

Lua offers new EEP-specific functions

Plug-in 2 for EEP 17.2 provides the following 12 new Lua functions:

| EEPStructureGetTextureText() | | EEPStructureGetTextureText ("#Lua-Name", Area_number) | |
|------------------------------|---|--|--|
| Parameters | two | ok, Text = EEPStructureGetTextureText("#147", 1) | |
| Returns | two | | |
| Requires | EEP 17.2 - Plug-in 2 | | |
| Purpose | Returns the text being displayed on the writable area of a landscape or structure element. | | |
| Notes | <ul style="list-style-type: none">• First parameter is the Lua name of the structure or landscape element. The difference between the Lua name and the model name is the presence of the ID in the header. The ID preceded by # is already sufficient for the argument.• Second parameter is the number of the area that is supposed to display the text. Some models have more than just one writable area.• First return value is true when the execution was successful, false if not.• Second return value is the text being displayed on the specified area. | | |

Manual to Plug-in 2 for EEP 17.2

| EEPSignalGetTextureText() | | EEPSignalGetTextureText(<i>Signal-ID</i> , <i>Area_number</i>) |
|---------------------------|---|---|
| Parameters | two | ok, Text = EEPSignalGetTextureText(88, 1) |
| Returns | two | |
| Requires | EEP 17.2 - Plug-in 2 | |
| Purpose | Returns the text being displayed on the writable area of a signal. | |
| Notes | <ul style="list-style-type: none">• First parameter is the ID of the signal.• Second parameter is the number of the area that is supposed to display the text. Some models have more than just one writable area.• First return value is <i>true</i> when the execution was successful, <i>false</i> if not.• Second return value is the text being displayed on the specified area. | |

| EEPGoodsGetTextureText() | | EEPGoodsGetTextureText("#Lua-Name", Area_number) |
|--------------------------|--|---|
| Parameters | two | ok, Text = EEPGoodsGetTextureText("#137", 1) |
| Returns | two | |
| Requires | EEP 17.2 - Plug-in 2 | |
| Purpose | Returns the text being displayed on the writable area of cargo. | |
| Notes | <ul style="list-style-type: none">• First parameter is the Lua name of the cargo as a string. The difference between the Lua name and the model name is the presence of the ID in the header. The ID preceded by # is already sufficient for the argument.• Second parameter is the number of the area that is supposed to display the text. Some models have more than just one writable area.• First return value is true when the execution was successful, false if not.• Second return value is the text being displayed on the specified area. | |

| EEPRailTrackGetTextureText() | | EEPRailTrackGetTextureText (Gleis-ID, Area_number) |
|------------------------------|---|--|
| Parameters | two | ok, Text = EEPRailTrackGetTextureText(67, 1) |
| Returns | two | |
| Requires | EEP 17.2 - Plug-in 2 | |
| Purpose | Returns the text being displayed on the writable area of rail track. | |
| Notes | <ul style="list-style-type: none">• First parameter is the ID of the rail section.• Second parameter is the number of the area that is supposed to display the text. Some models have more than just one writable area.• First return value is true when the execution was successful, false if not.• Second return value is the text being displayed on the specified area. | |

Manual to Plug-in 2 for EEP 17.2

| EEPRoadTrackGetTextureText() | | EEPRoadTrackGetTextureText(Strassen-ID , Area_number) |
|------------------------------|--|--|
| Parameters | two | ok, Text = EEPRoadTrackGetTextureText(128, 1) |
| Returns | two | |
| Requires | EEP 17.2 - Plug-in 2 | |
| Purpose | Returns the text being displayed on the writable area of road. | |
| Notes | <ul style="list-style-type: none">• First parameter is the ID of the road section• Second parameter is the number of the area that is supposed to display the text. Some models have more than just one writable area.• First return value is true when the execution was successful, false if not.• Second return value is the text being displayed on the specified area. | |

| EEPTramTrackGetTextureText() | | EEPTramTrackGetTextureText (Strassenbahngleis-ID, Area_number) |
|------------------------------|---|--|
| Parameters | two | ok, Text = EEPTramTrackGetTextureText(213, 1) |
| Returns | two | |
| Requires | EEP 17.2 - Plug-in 2 | |
| Purpose | Returns the text being displayed on the writable area of a tram track. | |
| Notes | <ul style="list-style-type: none">• First parameter is the ID of the tram track.• Second parameter is the number of the area that is supposed to display the text. Some models have more than just one writable area.• First return value is true when the execution was successful, false if not.• Second return value is the text being displayed on the specified area. | |

Manual to Plug-in 2 for EEP 17.2

| EEPAuxiliaryTrackGetTextureText() | | EEPAuxiliaryTrackGetTextureText(<i>Spline-ID</i> , <i>Area_number</i>) |
|-----------------------------------|---|---|
| Parameters | two | ok, Text = EEPAuxiliaryTrackGetTextureText(197, 1) |
| Returns | two | |
| Requires | EEP 17.2 - Plug-in 2 | |
| Purpose | Returns the text being displayed on the writable area of an auxiliary track. | |
| Notes | <ul style="list-style-type: none">• First parameter is the ID of the auxiliary track category element.• Second parameter is the number of the area that is supposed to display the text. Some models have more than just one writable area.• First return value is true when the execution was successful, false if not.• Second return value is the text being displayed on the specified area. | |

| EEPGetSeason() | | EEPGetSeason () |
|----------------|--|---------------------------------------|
| Parameters | none | <code>Season = EEPGetSeason ()</code> |
| Returns | one | |
| Requires | EEP 17.2 - Plug-in 2 | |
| Purpose | Returns the season set in the layout. | |
| Notes | <ul style="list-style-type: none">• This function is called by EEP without any parameters.• The return value is the season set in the layout as a number: 1 = Spring, 2 = Summer, 3 = Autumn, 4 = Winter. | |

| EEPGetCurrentFrame () | | EEPGetCurrentFrame () |
|-----------------------|---|-------------------------------------|
| Parameters | none | FrameNumber = EEPGetCurrentFrame () |
| Returns | one | |
| Requires | EEP 17.2 Plug-in 2 | |
| Purpose | Returns the current value of the frames counter since system start without the images during editing and pause mode. | |
| Notes | <ul style="list-style-type: none">• This function is called by EEP without any parameters.• The return value is the current value of the frames counter starting with the loading of the layout file (.anl3) as a number, whereby the images in edit and pause mode were not counted. | |

Manual to Plug-in 2 for EEP 17.2

| EEPGetFramesPerSecond() | | EEPGetFramesPerSecond() |
|-------------------------|--|-------------------------------|
| Parameters | none | fps = EEPGetFramesPerSecond() |
| Returns | one | |
| Requires | EEP 17.2 Plug-in 2 | |
| Purpose | Returns the current frame rate (fps). | |
| Notes | <ul style="list-style-type: none">• This function is called by EEP without any parameters.• The return value is the current frame rate (number of frames per second) as a number. | |

| EEPGetCurrentRenderFrame () | | EEPGetCurrentRenderFrame () |
|-----------------------------|--|---|
| Parameters | none | FrameNumber = EEPGetCurrentRenderFrame () |
| Returns | one | |
| Requires | EEP 17.2 Plug-in 2 | |
| Purpose | Returns the total number of images rendered since the start of the layout, including images rendered during edit and pause modes. | |
| Notes | <ul style="list-style-type: none">• This function is called by EEP without any parameters.• The return value is the current value of the frames counter starting with the loading of the layout file (.anl3) as a number, whereby the images in edit and pause mode were also counted. | |

| EEPGetTimeLapse() | | EEPGetTimeLapse () |
|-------------------|--|---------------------------------------|
| Parameters | none | Time-lapse_factor = EEPGetTimeLapse() |
| Returns | one | |
| Requires | EEP 17.2 Plug-in 2 | |
| Purpose | Returns the time-lapse factor currently set in EEP. | |
| Notes | <ul style="list-style-type: none">• This function is called by EEP without any parameters.• The return value is the time-lapse factor currently set in EEP, 1, 5 or 10 as a number. | |

Manual to Plug-in 2 for EEP 17.2

9 other Lua functions have also been extended with the arrival of plug-in 2 for EEP17.2

| EEPSetTrainSpeed() | | EEPSetTrainSpeed("#Name", Speed [, false true]) |
|--------------------|---|---|
| Parameters | two or three | |
| Returns | one | ok = EEPSetTrainSpeed("#Passenger train", 80) |
| Requires | EEP 11.0 EEP 17.2 - Plug-in 2 | ok = EEPSetTrainSpeed("#Rheingold", 80, false) ok = EEPSetTrainSpeed("#Rheingold", 80, true) |
| Purpose | Assigns a new speed to a specified "vehicles composite" (e.g. a train). | |
| Notes | <ul style="list-style-type: none"> First parameter is the name of the "vehicles composite" as a string preceded by #. Second parameter is the target speed. A negative value will result in a reverse gear. Since EEP 17.2 Plug-in 2, an optional third parameter can be entered with true or false respectively 1 or 0. <ul style="list-style-type: none"> If false or 0 or non-existence, a new target speed is assigned - as before - whereby any signal influence is cancelled (i.e. a "vehicles composite" that is currently being held up by a signal sets off). If true or 1, a "cruising" speed is assigned to the "vehicles composite" while maintaining a signal influence (i.e. a "vehicles composite" currently stopped by a signal will continue to stop). Return value is true if the targeted "vehicles composite" exists, or false if it doesn't. | |

| EEPGetTrainSpeed() | | EEPGetTrainSpeed("#Name" [, false true]) |
|--------------------|---|---|
| Parameters | one or two | |
| Returns | two | ok, TargetSpeed = EEPGetTrainSpeed("#VT98;001") ok, TargetSpeed = EEPGetTrainSpeed("#VT98;001", false) |
| Requires | EEP 11.0 EEP 17.2 - Plug-in 2 | ok, CruisingSpeed = EEPGetTrainSpeed("#VT98;001", true) |
| Purpose | Returns the target or cruising speed of a specified "vehicles composite" (e.g. a train) | |
| Notes | <ul style="list-style-type: none"> First parameter is the name of the "vehicles composite" as a string preceded by #. Since EEP 17.2 Plug-in 2, an optional second parameter can be entered with true or false respectively 1 or 0. <ul style="list-style-type: none"> If false or 0 or non-existence, the current actual speed is returned - as before. If true or 1, the "cruising" speed is returned (even if it is waiting in front of a signal!). First return value is true if the targeted "vehicles composite" exists, or false if not. Second return value is the determined speed of the "vehicles composite". Attention: After EEPSetTrainSpeed() EEPGetTrainSpeed() returns the new value in the next cycle of EEPMain() at the earliest. | |

Manual to Plug-in 2 for EEP 17.2

| EEPSetTrainAxis() | | <code>EEPSetTrainAxis("#Name", "Axis", Position [, false true])</code> |
|-------------------|---|--|
| Parameters | three or four | <code>ok = EEPSetTrainAxis("#Rail_crane", "raise/lower the boom", 100)</code> |
| Returns | one | |
| Requires | EEP 11.2 Plug-in 2 EEP 17.2 Plug-in 2 | <code>ok = EEPSetTrainAxis("#DustyGoodsTrain", "Dome cover", 100, true)</code> |
| Purpose | Animates one or more selected axes of the specified "vehicles composite" (e.g. a train). | |
| Notes | <ul style="list-style-type: none"> First parameter is the name of the "vehicles composite" as a string preceded by #. Second parameter is the name of the axis as a string. Third parameter is the target position of the axis. Since EEP 17.2 Plug-in 2, an optional fourth parameter can be entered with true or false respectively 1 or 0. <ul style="list-style-type: none"> If true or 1, the term entered as the 2nd parameter serves as a filter for all axis names that begin with the filter fragment. If, for example, "dome cover" is entered as the second parameter, the function sets the axes "dome cover 1", "dome cover 2", "dome cover 3", etc. (if they exist) to the position entered as the third parameter. If false or if the 4th parameter does not exist, the complete axis name must always be entered as the second parameter. Return value is true if the targeted "vehicles composite" and the axis exist, false if not. | |

| EEPRollingstockSetAxis() | | <code>EEPRollingstockSetAxis("Name", "Axis", Position [, true false])</code> |
|--------------------------|--|--|
| Parameters | three or four | <code>Name = "CoalPortalCrane" Axis = "Turn left" ok = EEPRollingstockSetAxis(Name, Axis, 50)</code> |
| Returns | one | |
| Requires | EEP 11.0 EEP 17 2 - Plug-in 2 | <code>ok = EEPRollingstockSetAxis("Dispolok SBB Cargo", "Pantograph", 0, true)</code> |
| Purpose | Moves the specified axis/axes of the specified rollingstock to the desired position. | |
| Notes | <ul style="list-style-type: none"> First parameter is the name of the rollingstock as a string. Second parameter is the name of the axis as a string. Third parameter is the target position of the axis. Since EEP 17.2 Plug-in 2, an optional fourth parameter can be entered with true or false respectively 1 or 0. <ul style="list-style-type: none"> If true or 1, the term entered as the second parameter serves as a filter for all axis names that begin with the filter fragment. If, for example, " Pantograph " is entered as the second parameter, the function sets the axes " Pantograph 1", " Pantograph 2", " Pantograph 3", etc. (if they exist) to the position entered as the third parameter. If false or if the 4th parameter does not exist, the complete axis name must always be entered as the second parameter. Return value is either true, if the rollingstock and the axis exist, false if at least one is missing. | |

Manual to Plug-in 2 for EEP 17.2

| EEPIsRailTrackReserved() | | EEPIsRailTrackReserved(ID [,true number]) |
|--------------------------|---|--|
| Parameters | one or two | |
| Returns | two or three | ok, Occupied = EEPIsRailTrackReserved(123) |
| Requires | EEP 11.3 - Plug-in 3 | ok, Occupied, Name = EEPIsRailTrackReserved(123, true) |
| | EEP 13.2 - Plug-in 2 EEP 17.2 - Plug-in 2 | ok, Occupied, Name = EEPIsRailTrackReserved(123, 2) |
| Purpose | Determines whether a track element is occupied or whether it is occupied with the xth "vehicles composite" and then returns its name. | |
| Notes | <ul style="list-style-type: none"> Parameter is the ID of the rail track to be checked. Since EEP 13.2 Plug-in 2, an optional second parameter true or 1 can be given so that the function returns the name of the foremost vehicle formation in the track direction as the third return value. Since EEP 17.2 Plug-in 2, numbers greater than 1 can also be entered so that the function outputs the name of the train formation on the corresponding rear position as the third return value. First return value is true if the targeted rail track exists and has been registered beforehand, false if not. Second return value is true without the second parameter if the track is occupied, otherwise false. With the second parameter it returns true if there is a "vehicles composite" at the location designated in the second parameter (where true = 1 there), otherwise false. Third (optional) return value is the name of the "vehicles composite" occupying the track at the position specified as the second parameter (where true = 1). If a number is entered as the third parameter which is greater than the number of "vehicles composites" on the track, the third return value is an empty character string. ATTENTION: The third return value is correct as long as the trains are not moving! But if the vehicles are moving, the Lua interpreter can deliver wrong names, because the list with the names changes e.g. with 60 fps, but Lua runs asynchronously in another thread (CPU) in order not to slow down EEP. Remember to register the track before using this function. | |

Manual to Plug-in 2 for EEP 17.2

| EEPIsRoadTrackReserved() | | EEPIsRoadTrackReserved(ID [,true number]) |
|--------------------------|--|---|
| Parameters | one or two | |
| Returns | two or three | ok, Occupied = EEPIsRoadTrackReserved(211) |
| Requires | EEP 11.3 - Plug-in 3 | ok, Occupied, Name = EEPIsRoadTrackReserved(211, true) |
| | EEP 13.2 - Plug-in 2 EEP 17.2 - Plug-in 2 | ok, Occupied, Name = EEPIsRoadTrackReserved(211, 3) |
| Purpose | Determines whether a road track element is occupied or whether it is occupied with the xth "vehicles composite" and then returns its name. | |
| Notes | <ul style="list-style-type: none"> Parameter is the ID of the road track to be checked. Since EEP 13.2 Plug-in 2, an optional second parameter true or 1 can be given so that the function returns the name of the foremost "vehicles composite" in the road direction as the third return value. Since EEP 17.2 Plug-in 2, numbers greater than 1 can also be entered so that the function outputs the name of the "vehicles composite" on the corresponding rear position as the third return value. First return value is true if the targeted road exists and has been registered beforehand, false if not. Second return value is true without the second parameter if the road track is occupied, otherwise false. With the second parameter it returns true if there is a "vehicles composite" at the location designated in the second parameter (where true = 1 there), otherwise false. Third (optional) return value is the name of the "vehicles composite" occupying the road track at the position specified as the second parameter (where true = 1). If a number is entered as the third parameter which is greater than the number of "vehicles composites" on the road track, the third return value is an empty character string. ATTENTION: The third return value is correct as long as the trains are not moving! But if the vehicles are moving, the Lua interpreter can deliver wrong names, because the list with the names changes e.g. with 60 fps, but Lua runs asynchronously in another thread (CPU) in order not to slow down EEP. Remember to register the road before using this function. | |

Manual to Plug-in 2 for EEP 17.2

| EEPIsTramTrackReserved() | | EEPIsTramTrackReserved(ID [,true number]) |
|--------------------------|---|--|
| Parameters | one or two | |
| Returns | two or three | ok, Occupied = EEPIsTramTrackReserved(187) |
| Requires | EEP 11.3 - Plug-in 3 | ok, Occupied, Name = EEPIsTramTrackReserved(187, true) |
| | EEP 13.2 - Plug-in 2 EEP 17.2 - Plug-in 2 | ok, Occupied, Name = EEPIsTramTrackReserved(187, 2) |
| Purpose | Determines whether a tramway track element is occupied or whether it is occupied with the xth "vehicles composite" and then returns its name. | |
| Notes | <ul style="list-style-type: none"> Parameter is the ID of the tramway track to be checked. Since EEP 13.2 Plug-in 2, an optional second parameter true or 1 can be given so that the function returns the name of the foremost "vehicles composite" in the tram track direction as the third return value. Since EEP 17.2 Plug-in 2, numbers greater than 1 can also be entered so that the function outputs the name of the "vehicles composite" on the corresponding rear position as the third return value. First return value is true if the targeted tramway track exists and has been registered beforehand, false if not. Second return value is true without the second parameter if the tram track is occupied, otherwise false. With the second parameter it returns true if there is a "vehicles composite" at the location designated in the second parameter (where true = 1 there), otherwise false. Third (optional) return value is the name of the "vehicles composite" occupying the tram track at the position specified as the second parameter (where true = 1). If a number is entered as the third parameter which is greater than the number of "vehicles composites" on the tram track, the third return value is an empty character string. ATTENTION: The third return value is correct as long as the trains are not moving! But if the vehicles are moving, the Lua interpreter can deliver wrong names, because the list with the names changes e.g. with 60 fps, but Lua runs asynchronously in another thread (CPU) in order not to slow down EEP. Remember to register the tramway track before using this function. | |

Manual to Plug-in 2 for EEP 17.2

| EEPIsAuxiliaryTrackReserved() | | EEPIsAuxiliaryTrackReserved(ID [,true number]) |
|-------------------------------|---|---|
| Parameters | one or two | |
| Returns | two or three | ok, Occupied = EEPIsAuxiliaryTrackReserved(321) |
| Requires | EEP 11.3 - Plug-in 3 | ok, Occupied, Name = EEPIsAuxiliaryTrackReserved(321,true) |
| | EEP 13.2 - Plug-in 2 EEP 17.2 - Plug-in 2 | ok, Occupied, Name = EEPIsAuxiliaryTrackReserved(321,3) |
| Purpose | Determines whether an auxiliary track element is occupied or whether it is occupied with the xth "vehicles composite" and then returns its name. | |
| Notes | <ul style="list-style-type: none"> Parameter is the ID of the auxiliary track to be checked. Since EEP 13.2 Plug-in 2, an optional second parameter true or 1 can be given so that the function returns the name of the foremost "vehicles composite" in the track direction as the third return value. Since EEP 17.2 Plug-in 2, numbers greater than 1 can also be entered so that the function outputs the name of the "vehicles composite" on the corresponding rear position as the third return value First return value is true if the targeted auxiliary track exists and has been registered beforehand, false if not. Second return value is true without the second parameter if the auxiliary track is occupied, otherwise false. With the second parameter it returns true if there is a "vehicles composite" at the location designated in the second parameter (where true = 1 there), otherwise false. Third (optional) return value is the name of the "vehicles composite" occupying the auxiliary track at the position specified as the second parameter (where true = 1). If a number is entered as the third parameter which is greater than the number of "vehicles composites" on the auxiliary track, the third return value is an empty character string. ATTENTION: The third return value is correct as long as the trains are not moving! But if the vehicles are moving, the Lua interpreter can deliver wrong names, because the list with the names changes e.g. with 60 fps, but Lua runs asynchronously in another thread (CPU) in order not to slow down EEP. Remember to register the auxiliary track before using this function. | |

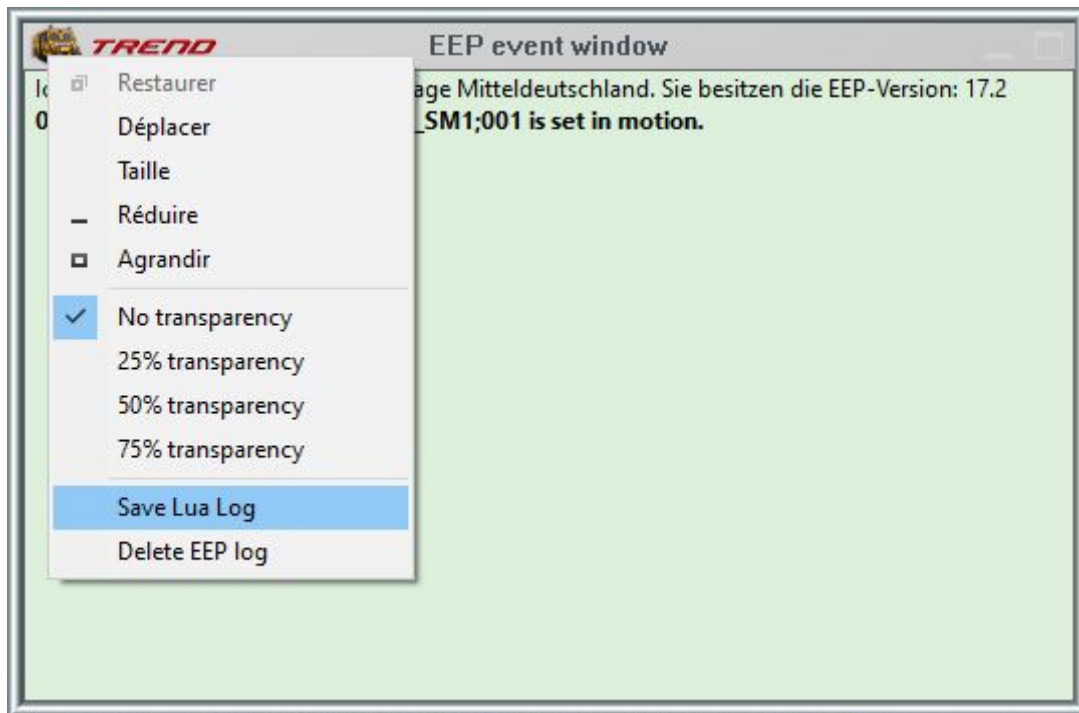
Manual to Plug-in 2 for EEP 17.2

| EEPIsControlTrackReserved() | | EEPIsControlTrackReserved(ID [,true number]) |
|-----------------------------|---|---|
| Parameters | one or two | |
| Returns | two or three | ok, Occupied = EEPIsControlTrackReserved(333) |
| Requires | EEP 11.3 - Plug-in 3 EEP 13.2 - Plug-in 2 EEP 17.2 - Plug-in 2 | ok, Occupied, Name = EEPIsControlTrackReserved(333, true) ok, Occupied, Name = EEPIsControlTrackReserved(333, 1) |
| Purpose | Determines whether a control track element is occupied or whether it is occupied with the xth "vehicles composite" and then returns its name | |
| Notes | <ul style="list-style-type: none"> Parameter is the ID of the section of the control track system to be checked. Since EEP 13.2 Plug-in 2, an optional second parameter true or 1 can be given so that the function returns the name of the foremost "vehicles composite" in the control track direction as the third return value. Since EEP 17.2 Plug-in 2, numbers greater than 1 can also be entered so that the function outputs the name of the "vehicles composite" on the corresponding rear position as the third return value First return value is true if the targeted control track system exists and has been registered beforehand, false if not. Second return value is true without the second parameter if the control track is occupied, otherwise false. With the second parameter it returns true if there is a "vehicles composite" at the location designated in the second parameter (where true = 1 there), otherwise false. Third (optional) return value is the name of the "vehicles composite" occupying the control track at the position specified as the second parameter (where true = 1). If a number is entered as the third parameter which is greater than the number of "vehicles composites" on the control track, the third return value is an empty character string. ATTENTION: The third return value is correct as long as the trains are not moving! But if the vehicles are moving, the Lua interpreter can deliver wrong names, because the list with the names changes e.g. with 60 fps, but Lua runs asynchronously in another thread (CPU) in order not to slow down EEP. Remember to register the control track system before using this function. | |

Manual to Plug-in 2 for EEP 17.2

In addition to the new features and changes introduced, there are two other changes regarding the use of Lua:

- Previously, when you closed the Lua script editor by clicking on the "Close Window" cross in the top right corner of the window, the Lua script would still be executed once, even if it remained unchanged. As requested by users, this no longer occurs starting from EEP 17.2 Plugin 2.
- With the Plugin 2 for EEP 17.2, the number of visible lines in the event window has been increased to 1,024 lines instead of 500. This allows for a larger range of events to be displayed. It is now also possible to save these events through the menu in the event window. Simply right-click with the mouse to bring up the context menu. The saved log file is named "MyProject.eep.log" and can be found in the same folder as the "MyProject.anl3" file. Here, "MyProject" refers to the name under which your layout has been saved.



Manual to Plug-in 2 for EEP 17.2

Closing remarks

Many wishes of experienced EEP users and designers have been incorporated into the development of Plug-In 2 for EEP 17.2. 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 17.2.

Your EEP-Team of Trend Verlag