



Scenario authors guide

Part II

Advanced topics scenario creation for Trainsimulator

Rudolf Heijink

Version 0.2, May 2016

Copyright © 2014/2015 Rudolf Heijink.



This work is licensed under the *Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License*. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:



Attribution — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



NonCommercial — You may not use the material for [commercial purposes](#).



ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](#) as the original.

No additional restrictions — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.

Notices:

- You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable [exception or limitation](#).
- No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as [publicity, privacy, or moral rights](#) may limit how you use the material.
- Dovetail Games has been granted a non-exclusive license for commercial use of this guide.

Preface

Introduction

It started when I got interested in creating scenario scripts and decided to create a scripting guide, because nothing like that existed. During writing the guide I managed to get in touch with DTG and this all ended with the request to create tutorial guide on creating scenarios.

The final result will be a three part Scenario authors guide:

- ✓ Part I: The starters guide, which introduces scenario creation.
- ✓ Part II, this guide, will cover advanced topics. You are now reading version 0.2
- ✓ Part III, covering

Where Part I enables you to create freeroam and standard scenarios for your own use, this part will contain advanced topics, like publishing scenarios, quick drive, using scenery items, the cinematic camera etcetera.

Images

Each chapter is introduced with a screenshot. They represent my favourite routes, rolling stock and developers. It's a mix between payware, donationware and freeware, often combined in a single picture and is meant as a thank you to the TS2016 development community.

Acknowledgements

- All community members for sharing information and answering questions.
- [UKTS community site](#)
- [maxtedrw](#)
- [Matt Peddlesden](#)

Frontpage image:

Screenshot of the fabulous Albula line.

Disclaimer

This guide is provided "as is" the author cannot accept any consequences from the use of this guide.

The contents is the sole responsibility of the author.

Contact

Comments are welcome at trainsimulator@hollandhiking.nl.

But please be aware that I cannot provide you help with you scenario creating problems. If you have any questions, please use one of the regular community forums.

Enjoy reading!

Rudolf Heijink

Contents

| | |
|--|-----------|
| Preface | 3 |
| 1 Introduction | 9 |
| 2 Track settings | 10 |
| 2.1 Marker types | 10 |
| 2.2 Creating a destination marker in a scenario..... | 10 |
| 2.3 Create a platform marker..... | 13 |
| 2.4 Create a stopping point | 13 |
| 2.5 Problem editing the marker | 15 |
| 2.6 Custom track properties..... | 15 |
| 3 Consists | 18 |
| 3.1 Saving consists | 18 |
| 3.2 Using saved consists | 20 |
| 3.3 Consist operations for AI trains | 20 |
| 3.4 Using favorites in the scenario editor | 22 |
| 4 Scenario settings..... | 24 |
| 4.1 Setting switches..... | 24 |
| 4.2 Rolling start..... | 24 |
| 4.3 Force cab camera | 26 |
| 4.4 Expert controls or simple controls | 26 |
| 5 Workorder tweaks | 28 |
| 5.1 Empty passenger coaches | 28 |
| 5.2 Advanced coupling | 28 |
| 5.3 Advanced marshalling | 29 |
| 5.4 Instruction order..... | 29 |
| 5.5 Hidden instructions | 30 |
| 6 More about scenario types | 31 |
| 6.1 Change the scenario type | 31 |
| 6.2 Scenario marker location..... | 31 |
| 6.3 Timetabled scenarios..... | 31 |
| 6.4 Career scenarios | 32 |
| 6.5 Quick drive scenarios..... | 34 |
| 6.5.1 Basic principles | 34 |
| 6.5.2 Planning | 35 |
| 6.5.3 Creating a Quick Drive Scenario basics..... | 36 |
| 6.5.4 Setting up The Player Spawn | 37 |
| 6.5.5 Setting the Route for your train | 39 |
| 6.5.6 The Via Command | 39 |
| 6.5.7 Adding more Player Spawns to your Scenario | 39 |
| 6.5.8 User friendly naming scheme | 40 |
| 6.5.9 Static Consists | 40 |
| 6.5.10 Using the tape measure..... | 41 |
| 6.5.11 How to Decide what is Spawned | 41 |
| 6.5.12 AI Spawn Points | 42 |
| 6.5.13 Routing your AI | 43 |
| 6.5.14 Selecting the AI to be used | 43 |

| | | |
|-----------|---|--|
| 6.5.15 | Goods Yards..... | 44 |
| 6.5.16 | Creating Quick Drive consists | 44 |
| 6.6 | Relay scenarios and scenario chaining | 46 |
| 6.7 | Railfan scenarios..... | 46 |
| 6.7.1 | Add invisible track or use a siding | 47 |
| 6.7.2 | Create a destination marker..... | 47 |
| 6.7.3 | Put a player engine at the track | 47 |
| 6.7.4 | Instructions setup for the player engine | 47 |
| 6.7.5 | Insert the LUA script | 49 |
| 6.7.6 | Set up the AI traffic..... | 54 |
| 7 | Changing the weather | 59 |
| 7.1 | Selecting weather | 59 |
| 7.2 | Create your own weather..... | 60 |
| 7.3 | Extended weather | Fout! Bladwijzer niet gedefinieerd. |
| 8 | Add scenery to scenarios | 55 |
| 8.1 | Add scenery | 55 |
| 8.2 | Make scenery objects available in the scenario editor | 57 |
| 9 | The cinematic camera | 68 |
| 10 | Tools | 69 |
| 10.1 | Utilities..... | 69 |
| 10.2 | LuaCreator | 69 |
| 10.3 | NameMyRoute | 70 |
| 10.4 | Serz | 70 |
| 10.5 | RWTools..... | 70 |
| 10.6 | RW TextEdit | 70 |
| 10.7 | XML Notepad | 70 |
| 11 | Create a scenario for other people | 71 |
| 11.1 | Publication medium..... | 71 |
| 11.2 | Use of rolling stock | 71 |
| 11.3 | Timing and testing | 72 |
| 11.4 | Make sure AI trains are seen | 72 |
| 11.5 | Documentation..... | 72 |
| 11.6 | Localization..... | 72 |
| 11.7 | Translating your scenario to other languages | 73 |
| 12 | Publishing scenarios..... | 74 |
| 12.1 | Where to publish | 74 |
| 12.2 | Create an rwp file | 74 |
| 12.3 | Compressing files..... | 76 |
| 12.4 | Publish in steam workshop..... | 76 |
| 13 | Scripting | 78 |
| 14 | Trouble shooting..... | 79 |
| 14.1 | Editing the scenario properties | 79 |
| 14.2 | Editing scenario.bin file | 79 |
| 14.3 | Follow AI trains | 79 |
| 14.4 | Speed up scenario execution..... | 79 |
| 14.5 | Recalculate timing and show driver list..... | 79 |
| 14.6 | Miscellaneous troubleshooting tips | 80 |

| | | |
|-----------|--|-----------|
| 14.6.1 | Avoid service class Special | 80 |
| 14.7 | Avoid minimum speeds for stop at instructions | 80 |
| 14.7.1 | Avoid testing a scenario directly from the editor..... | 80 |
| 14.7.2 | Check your path..... | 81 |
| 14.8 | Solving problems with tracks and signals..... | 81 |
| A. | Useful links | 85 |
| B. | Availability of this guide..... | 86 |
| | Index | 88 |

List of figures

| | |
|--|----|
| Figure 1. Selecting markers. | 11 |
| Figure 2. Added a destination marker..... | 11 |
| Figure 3. Edit the name of the marker | 12 |
| Figure 4. The Destination marker now can be selected in the marker list of the route | 12 |
| Figure 5 Edit stopping point | 13 |
| Figure 6. Stopping point in the 2D view in the editor. | 14 |
| Figure 7. Stopping point as shown in the instructions list. | 14 |
| Figure 8. Stopping point as shown in the HUD..... | 14 |
| Figure 9. Screen icon in the toolbox..... | 15 |
| Figure 10. Make sure to check 3D visibility of markers and marker areas. | 15 |
| Figure 11. The select tool allows to customize track properties..... | 16 |
| Figure 12. Custom track marker | 16 |
| Figure 13. Custom track properties edit dialog..... | 17 |
| Figure 14. Create a saved consist. | 18 |
| Figure 15. The grey cubes indicate selectable consists | 19 |
| Figure 16. The saved consist properties form | 19 |
| Figure 17. Using saved consists..... | 20 |
| Figure 18. Example of AI train first dropping off a consist and the picking up another one. | 21 |
| Figure 19. The scenario in Figure 18, just before connection to the consist. | 21 |
| Figure 20. Example shows setting switches. | 24 |
| Figure 21. Activate rolling start. | 25 |
| Figure 22. Force cab camera at scenario start. | 26 |
| Figure 23. Forcing simple or expert controls..... | 27 |
| Figure 24. Set the consist as empty in the Driver Instruction. | 28 |
| Figure 25. Set the consist as empty in the Pick up passenger or stop at instruction. | 28 |
| Figure 26. The resulting briefing..... | 29 |
| Figure 27. Define a descriptive text for a coupling instruction. | 29 |
| Figure 28. Uncheck this to get a free order to complete instructions. | 30 |
| Figure 29. Check mark for hidden instruction. | 30 |
| Figure 30. Editing time tabled properties..... | 32 |
| Figure 31. Edit the career rewards and penalties here. | 33 |
| Figure 32. Scenario Properties for a Quick Drive Scenario..... | 36 |
| Figure 33. Player spawn marker which is the minimum requirement for a Quick Drive Scenario. | 37 |
| Figure 34 Player Spawn Point properties dialog | 38 |
| Figure 35. Static Consist Spawn Point | 40 |
| Figure 36. Using the Tape Measure..... | 41 |
| Figure 37. AI Spawn Point dialog showing the first tab..... | 42 |
| Figure 38. Selecting AI train types for Quick Drive Consists..... | 43 |
| Figure 39 Selecting the Railfan Engine | 47 |
| Figure 40 Railfan “engine” with Driver instruction | 48 |
| Figure 41. Make sure to enter “start” as the trigger event value. | 48 |
| Figure 42. Instructions for Railfan scenario..... | 49 |
| Figure 43. Name My Route utility. | 69 |
| Figure 44. Utilities packager screen | 75 |
| Figure 45. Selecting files in the packager | 75 |
| Figure 46. Uploading screenshots to steam..... | 76 |
| Figure 47. Publishing a scenario at steam workshop | 77 |
| Figure 48. Driver instructions list..... | 80 |
| Figure 49. Always restart the scenario after editing | 81 |

1 Introduction

You are reading part II of the Scenario Authors Guide, which appears in three parts. It is assumed that you have sufficient knowledge of all topics covered in Part I. Part I has the character of a tutorial. Part II aims to a systematic coverage of all topics you can use in creating scenarios. There is one exception, creating LUA scripts adds so much additional complexity that it is covered in part III of this guide.

The version you are reading now is not completed. I expect that I will need at least till June 2016 to provide a completed document, because I need to investigate many topics for myself before I can write about them.

Despite these restrictions, I think the content available now is useful for you. I hope you will feel encouraged to create and publish scenarios for other players.

Your tips, comments and additions are welcome. Please mail to trainsimulator@hollandhiking.nl

2 Track settings

In part I you have seen that markers are used as destinations for trains. A route builder will create most markers for you, but sometimes you will like to add a marker while creating a scenario. In principle you can edit the route and add the marker there. This has two disadvantages:

1. If the route is updated, the markers you added will be lost.
2. You cannot publish the scenario, because other players will not see the markers you add.

Fortunately you can add most marker types in a scenario as well. In this chapter you will learn how to do this.

Also, you can overrule the track rules. This also is described in this chapter.

2.1 Marker types

First we repeat the different marker types that were introduced in Part I and add some new ones.

| Marker type | Usage | Remarks |
|----------------------------|--|--|
| Destination marker | Designates a named destination for a consist. Shown as an orange line in the 2D view. | |
| Siding marker | Designates a named siding. Shown as a yellow line in the 2D view | |
| Platform marker | Designates a named platform, allows pick up of passengers. Shown as a green line in the 2D map. | Requires the presence of a platform next to the marker, otherwise pick up passengers will not work. |
| Stopping point | A named point on one of the markers mentioned before, which specifies a specific location where a train must stop. | On some routes used to mark the stop locations for different train lengths. Requires Trainsimulator 2014 or newer. |
| Portal | A named destination. AI trains will end here and be removed from the simulation when they arrive at a portal. | Portals cannot be created in a scenario. Portals are only a destination, not a spawn point. |
| Player spawn point | Location where a player train can start | Only for quickdrive scenarios |
| AI spawn point | Location where an AI train can start | Only for quickdrive scenarios |
| Consist spawn point | Location where a static consist can be placed | Only for quickdrive scenarios |
| Loading point | Equipment for loading or unloading or for fuelling engines. | Must be placed on top of another marker. |

Quickdrive is not covered in this chapter. See chapter 6.6.

2.2 Creating a destination marker in a scenario

You can place scenario markers at any track, but it is not possible to place a marker at places where a marker already exists. To place a marker, select in the editor the signal icon, which represents track infrastructure objects. You now see a list of objects and among them the markers will show up (Figure 1) . It may look different from this example, depending on other objects shown in this list. Pick the marker you want and

drag it to the track. For this example see the scenario demo “RJH Custom Markers Demo”. A destination marker is used here. When you place it on track you see something like in Figure 2.

You can now drag the green blocks at the ends to size the marker. If you don’t see, the green blocks or the marker, follow the instructions in chapter 2.5.

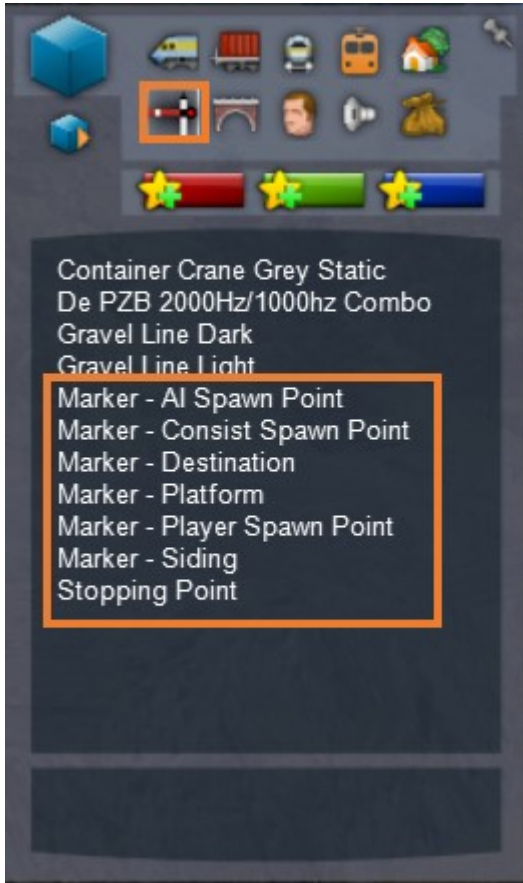


Figure 1. Selecting markers.

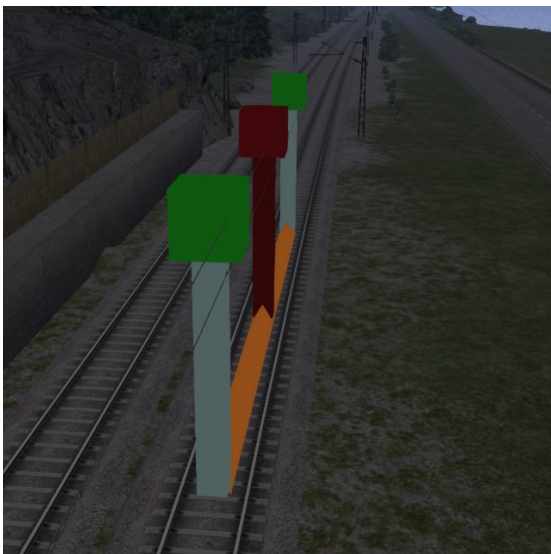


Figure 2. Added a destination marker.

The last step is to give the marker a user friendly name. You do this by clicking on the orange centre block of the marker. You will see a fly out on the right side of the screen. Place the desired name in the two boxes marked orange in Figure 3. In Figure 4 you see the Destination Marker is now visible in the marker list of the route.



Figure 3. Edit the name of the marker

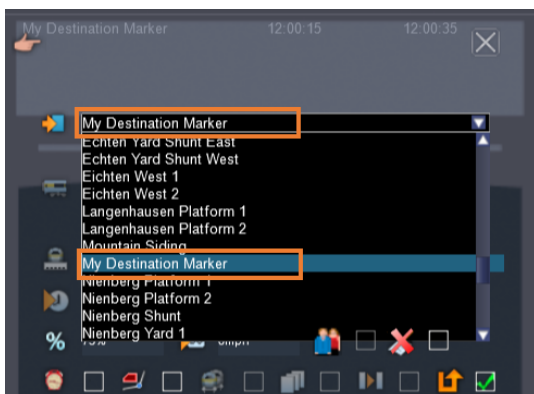


Figure 4. The Destination marker now can be selected in the marker list of the route

2.3 Create a platform marker

Creating a platform marker works the same way as creating a destination marker.

There is a problem here. A platform marker alone is not sufficient for a pick up passenger instruction. A pick-up passenger instruction requires both a platform marker and an invisible platform. Unfortunately you cannot add a platform in the scenario editor. Probably the blueprint does not support this.

2.4 Create a stopping point

A stopping point is a location you can add to a marker to indicate an exact spot where the consist must halt. In career scenarios you can subtract penalty points if the consist stops too far away from the stopping point.

Another application is to connect it to train length dependent platform stops. For each train length, create a

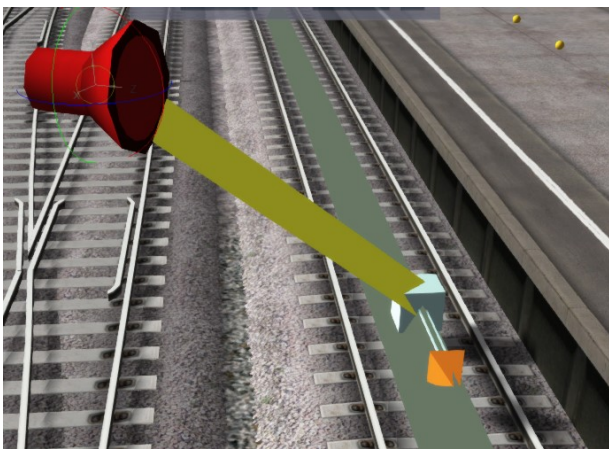


Figure 5 Edit stopping point

In Figure 5 you see what a **stopping point** looks like in the scenario editor. The arrow probably must point into the driving direction (not sure, you may need to test this). You can switch the direction using the **CTRL** or **SHIFT** key and the click with the **left mouse button**. The key you need, seems to depend on the direction,

DTG did for some reason not choose to make it a toggle.

You should give the stopping point a name, using a similar procedure as for the other markers. I did have some trouble making the settings form visible. It looks like double clicking on the top side of the cone does the trick.

I also noted that if you try to change the marker to a stopping point, this is not accepted. It works only if you create a new instruction and then select the stopping point immediately.

The stopping point in the example scenario is called: "Stop here @ Echten Platform 2"

In the editor, you see this description in the marker list, but in the instructions you only see "Stop here" and in gameplay you see "Echten Platform 2". In the HUD you can see the exact location of the stopping point.

This all is very confusing, see also Figure 6, Figure 7 and Figure 8.

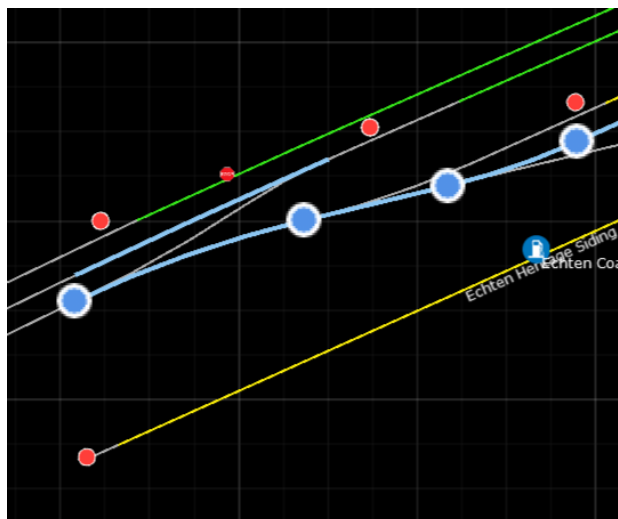


Figure 6. Stopping point in the 2D view in the editor.

| | Destination | Arrival Time | Departure time |
|--------------------------|-----------------------|--------------|----------------|
| | Player | --:-- | 12:00:00 |
| <input type="checkbox"/> | My Destination Marker | 12:00:15 | 12:00:35 |
| <input type="checkbox"/> | My Platform | 12:01:40 | 12:02:00 |
| <input type="checkbox"/> | Stop here | 12:06:42 | 12:07:17 |
| <input type="checkbox"/> | Stop here | 12:07:17 | 12:07:17 |

Figure 7. Stopping point as shown in the instructions list.



Figure 8. Stopping point as shown in the HUD.

2.5 Problem editing the marker

Sometimes it seems you cannot edit the marker you added. If you don't see the marker, or if you cannot change its length, you need to select the screen icon in the editor toolbox (Figure 9). This opens a form on the right hand side of the screen. In this screen Figure 10 make sure the check marks for marker visibility are set.

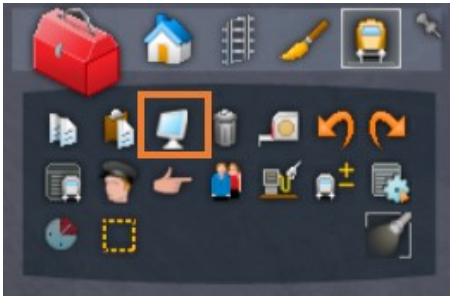


Figure 9. Screen icon in the toolbox.

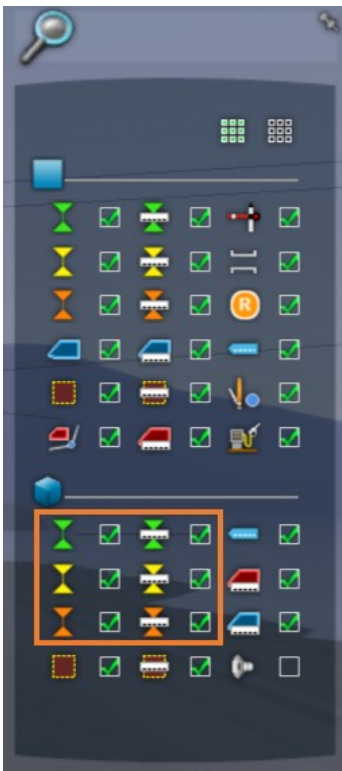


Figure 10. Make sure to check 3D visibility of markers and marker areas.

2.6 Custom track properties

In a scenario you can set custom track properties. You can do a lot with this:

1. Create a temporary speed limit
2. Repair the track type (passenger, freight, mainline etc. ...)
3. Make unidirectional track bidirectional
4. Add track unevenness, which degrades track quality



Figure 11. The select tool allows to customize track properties

To do so, open the scenario editor and pick the Select tool (Figure 11).

1. Click on the track, now a yellow marker will appear (Figure 13).
2. Keep your mouse button pressed, to extend the marker area
3. Release the mouse button and edit the track properties (Figure 12).



Figure 12. Custom track marker

In the track properties form, you can adjust some settings:

1. Speed limits for passenger/freight trains (nr 1)
2. Line type (nr 2)
3. Directionality (nr 3)
4. Unevenness(nr 4)

Note: unfortunately the select tool is not very intuitive in the way it works. You probably will need a lot of perseverance to get it working as you like. Especially, I do not know how to reopen an existing custom track.



Figure 13. Custom track properties edit dialog

3 Consists

In this chapter two topics will be covered.

1. You can save a consist in the scenario editor, and use this later to place the whole consist at once in a scenario.
2. There are (limited) possibilities to perform consist operations with AI trains. I will illustrate this using a short tutorial scenario, named RJH Tutorial 20. It is created for the Workshop version of the Seebergbahn.

3.1 Saving consists

If you want to re-use a consist, you can save the consist in the editor. To do so, select the consist edit icon in the toolbox (nr 1 in Figure 14). In the 3D editor view, you now see grey cubes above the consists (Figure 15). Now click on the grey cube. This causes the consist properties form to appear on the right side of the screen (Figure 16).



Figure 14. Create a saved consist.



Figure 15. The grey cubes indicate selectable consists.

You can set a descriptive name (nr 1) for the consist here, to help you to recognize it when you find it. You also can delete a saved consist by clicking at the delete icon (nr 2).

Note: make sure you have coupled all wagons you want to include in the saved consist, otherwise you may end with a partially saved consist.

Note: you should use this function sparingly. If you create many consists with more or less similar names, this may cause more confusion than it will help you.



Figure 16. The saved consist properties form

3.2 Using saved consists

In Figure 17 select the consist icon (nr 1). This will show a list of saved consists. The consist we just created is shown here at nr 2. If you click on the consist you want to use, the consist properties form will occur again. This allows you to rename the consist or delete it. If you want to use the saved consist, just drag it to the tracks. In principle the editor will generate new wagon numbers reach time you drag a consist into the editor.

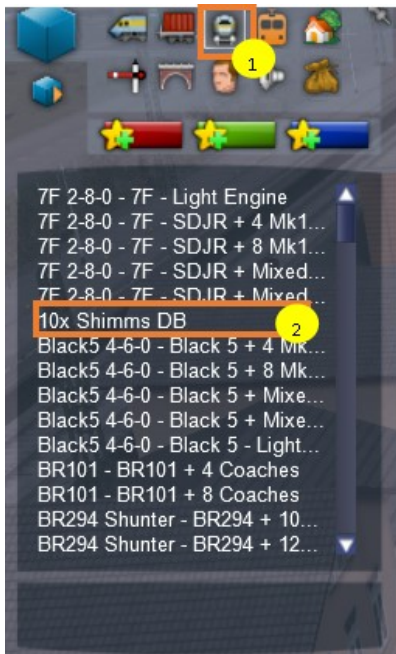


Figure 17. Using saved consists.

In the consist tab a consist only will be shown if for all included stock the provider and product have been selected (the procedure to do this, is described in Part I of the Scenario Authors Guide).

3.3 Consist operations for AI trains

In Part I of the Scenario Authors Guide I showed how to couple and uncouple wagons. You can do this as well with AI trains, provided you respect some limitations:

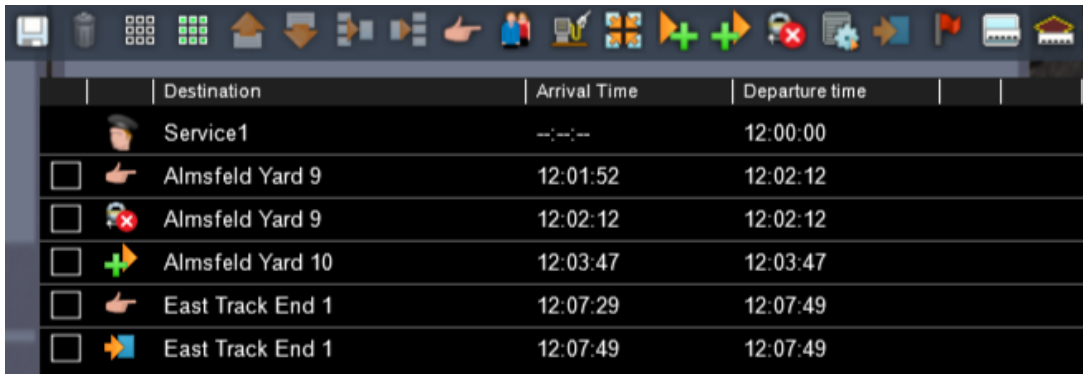
1. You can create an AI consist, drive it to a destination and uncouple the consist.
2. You can place a static consist in a scenario and make an AI train couple to it.
3. You cannot move a consist in a scenario, using the player train or an AI train and the couple it to an AI train.

The reason for this is, that the whole path of an AI train is fixed and calculated when you save an edited scenario. So if you move a consist, the AI dispatch function doesn't know exactly where to find it, causing erroneous path calculations.

Despite these restrictions, you can do some interesting things, e.g. make an AI train bring a consist, drop it off leaving a shunting duty for you. An example is my scenario **RJH Truckee Shunter** for Donner Pass (available in steam workshop, user RudolfJan).

I created a short tutorial, where an AI train brings a freight train with logs to a yard, and then picks up another consist and continues. You see the instructions list in Figure 18. This example does not use a player train, so it is much like a railfanning scenario. Elsewhere in this Guide (chapter 6.8), I will discuss railfanning more extensively. The scenario is called **RJH Tutorial 20** and runs on the Workshop version of Seeburgbahn.

Note: if you play the scenario you may notice that the train does not reverse after coupling, but continues backward to the main track. It is a nice exercise to try to repair this strange behaviour.



The screenshot shows a user interface for a train simulation. At the top is a toolbar with various icons for navigation and settings. Below the toolbar is a table with the following columns: Destination, Arrival Time, and Departure time. The table contains several rows of data, including 'Service1', 'Almsfeld Yard 9', and 'East Track End 1'.

| | Destination | Arrival Time | Departure time |
|--------------------------|------------------|--------------|----------------|
| | Service1 | --:-- | 12:00:00 |
| <input type="checkbox"/> | Almsfeld Yard 9 | 12:01:52 | 12:02:12 |
| <input type="checkbox"/> | Almsfeld Yard 9 | 12:02:12 | 12:02:12 |
| <input type="checkbox"/> | Almsfeld Yard 10 | 12:03:47 | 12:03:47 |
| <input type="checkbox"/> | East Track End 1 | 12:07:29 | 12:07:49 |
| <input type="checkbox"/> | East Track End 1 | 12:07:49 | 12:07:49 |

Figure 18. Example of AI train first dropping off a consist and the picking up another one.



Figure 19. The scenario in Figure 18, just before connection to the consist.

3.4 Using favorites in the scenario editor

If you create complex scenarios with a large number of consists and scenery items and you want to use them repeatedly, the scenario editor allows you to create three lists with favorite items. The first step is to add items to the favorites:

Right click on the item you want to add. Click once to add it to the **red list**, twice to add it to the **blue list** and three times to add it to the **green list**.

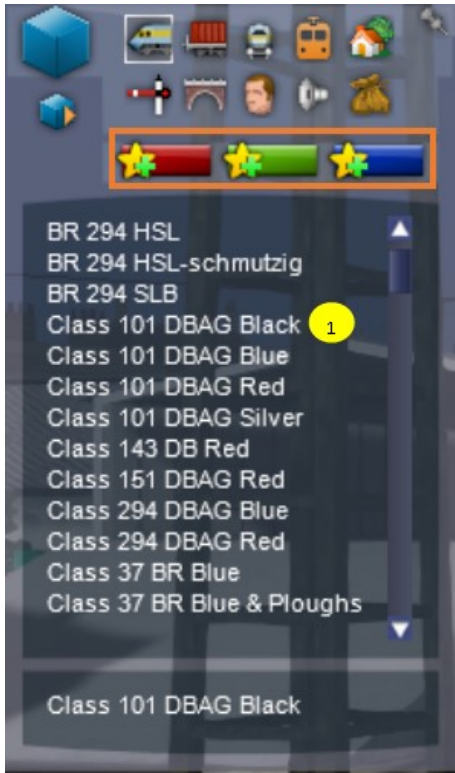


Figure 20. In the scenario editor you have three lists available for favorites.

The use the list, **click left** on the colored button.



Figure 21. Click at the red button to show the favorites list.

To go back and view all items, **click left** at one of the **item categories**. As you can see, there are no restrictions to the items you can add, including markers and consists.

To **delete** a **single item**, **right click** on the item in the list.

To **delete** the **whole list**, click right on the list button. You will be asked to confirm this operation.

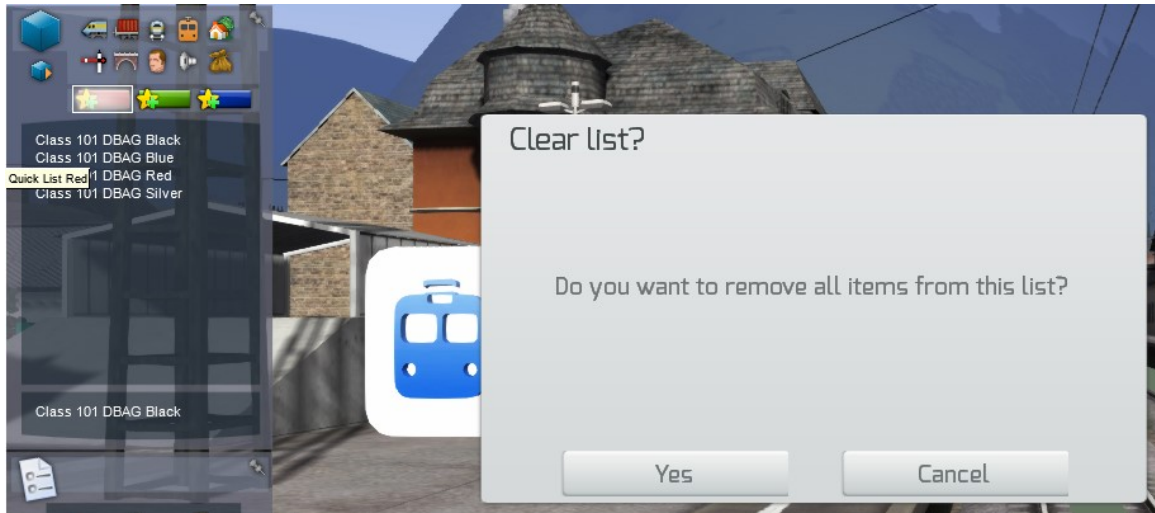


Figure 22. Right click on the red button to delete all items in the list.

The favorite lists are specific for a scenario, it seems the contents is not transferred to another scenario, but its content is preserved for other edit sessions.

3.5 Dead engines

[jalsina](#) 9 hours ago



Originally posted by [dcshoe](#):

For a dead loco, place the loco in the scenario editor, uncoupled and with no driver assigned to the locomotive. Double-click on the locomotive to be 'dead'. In the right hand fly out settings dialogue box, the one used to load loadable rolling stock, you'll see the loco number. Beneath this is an icon of a white outline of an HST-style engine with a red cross over it. Check the box beside it and the loco will remain dead when you couple to it. This procedure only works with uncoupled locomotives. If they're coupled, the lead unit governs. Dead locomotives cannot have drivers nor be selected in the driving scenario.

It works fine with a diesel. With an electric you can also lower the pantograph but who cares as they do not have particles emission (except the hybrid locos).

With a steam loco I think there is no way to "kill" the smoke, other than playing with the engine or the particles xlm (bin). Some have obtained results by extinguishing the fire mass.

4 Scenario settings

In this chapter several more fancy scenario settings will be discussed. For some of them tutorial scenarios will be provided.

4.1 Setting switches

Trainsimulator distinguishes automatic switches and manual switches. In principle the player must set the manual switches to create the desired path. It is possible to set manual switches in the correct position during scenario creation. It works very easy:

In the scenario editor, press **key 9** to enter the 2D view.

Click with the left mouse key at the blue circles representing the manual switch. You see the set path drawn with a slightly thicker line. Also you can see the blue line of the player path that is set.

When you save the scenario, a file called **InitialSave.bin** is created/updated. This file contains the settings for all switches.

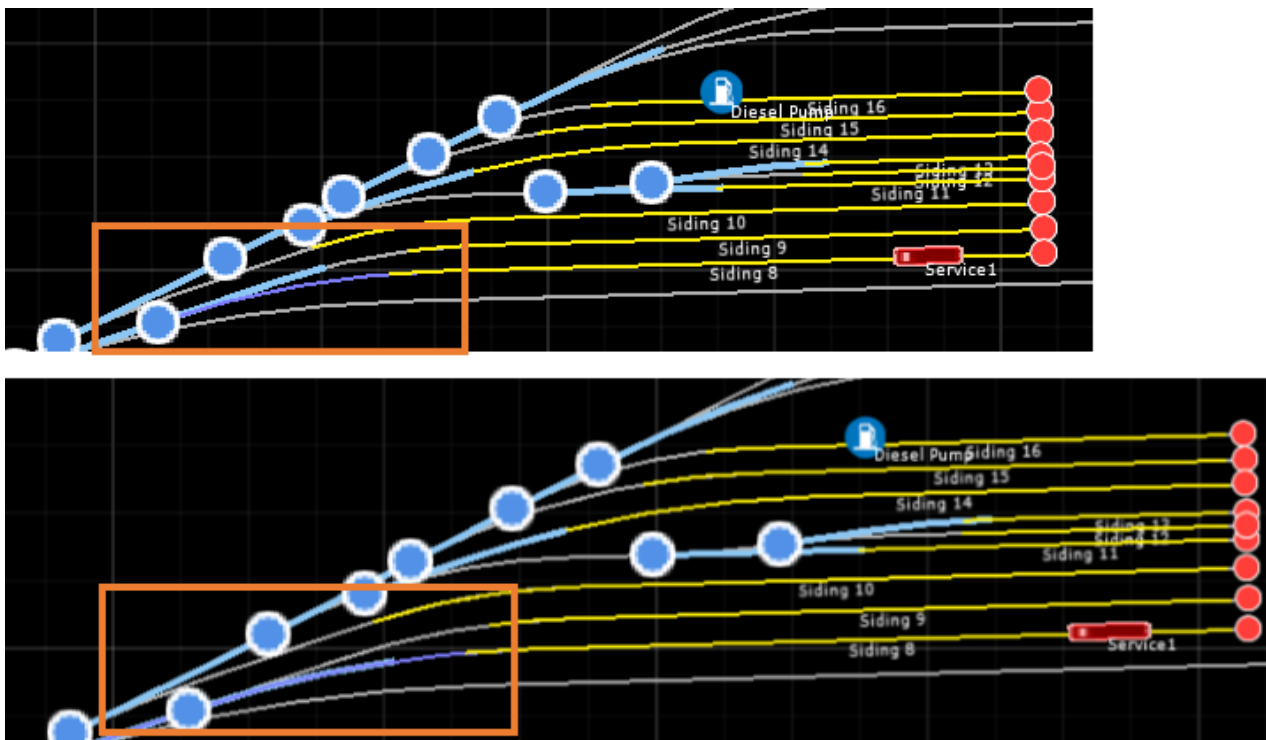


Figure 23. Example shows setting switches.

In the upper figure of Figure 23 the switch is not set for service1. In the lower figure you see the situation after the switch is set properly.

4.2 Rolling start

If you want to set the initial state of player engine, e.g. make the consist depart at a certain speed, create a dead engine with all functions turned off, you can use the Rolling Start feature. The rolling start feature is able to store other engine parameters that just the speed as well. It does not always work, so you need to try and see what works. Some people report it is not portable to other computers, but rolling start just creates another type of save file for the scenario.



Figure 24. Activate rolling start.

It works like this:

1. Create your scenario, tick the rolling start check box on the scenario properties dialog (Figure 24Figure 26. Forcing simple or expert controls.).
2. Start playing the scenario. When you have done all settings (eventually bringing the consist to the desired speed), use **CTRL+F2** to save the scenario. This invokes a special save mode for rolling start.
3. Play the scenario again to verify the settings.

Note:

If you edit your scenario, it is recommended to remove the files **StartingSave.bin** and **StartingSave.bin.MD5**. These files may corrupt your scenario if you add or remove events.

The rolling start seems not save all settings. You need to check if it works as you intended.

4.3 Force cab camera

You can force the player to start inside the cab. To do this, you need to set the check mark in the scenario properties.



Figure 25. Force cab camera at scenario start.

4.4 Expert controls or simple controls

Normally the user determines in the game options if he/she wants to use simple or expert controls. For some scenarios it may be necessary or helpful to force either simple controls or expert controls:

1. Simple controls may be useful in a tutorial scenario where you assume simple controls.
2. Expert controls make sense if the player engine only supports expert controls.
3. For career scenarios the points rewarded may be based upon the type of controls used. You may want to prevent cheating.

In the scenario properties (Figure 26) you can easily set Simple controls (nr 1) or Expert controls (nr 2). You also can select both. I did not try to find out which effect this causes.



Figure 26. Forcing simple or expert controls.

5 Workorder tweaks

5.1 Empty passenger coaches

It is natural that sometimes passenger trains are empty. Since Trainsimulator 2015 you can achieve this. The driver instruction, stop at instruction and pick up passenger instructions allow you to specify the wagons must be empty. This empty status is valid till the next instruction where you can change it. You must take care to use realistic values. An empty train, performing a pick up passengers, will not be empty afterwards. An empty train just stopping at the platform should be empty afterwards.

You can simply set the check mark in the box to make a consist empty. See Figure 27 and Figure 28.

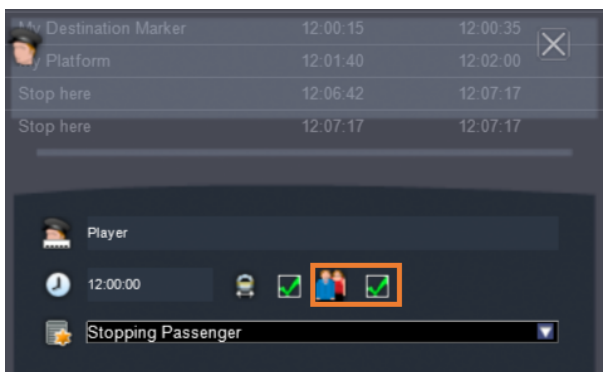


Figure 27. Set the consist as empty in the Driver Instruction.

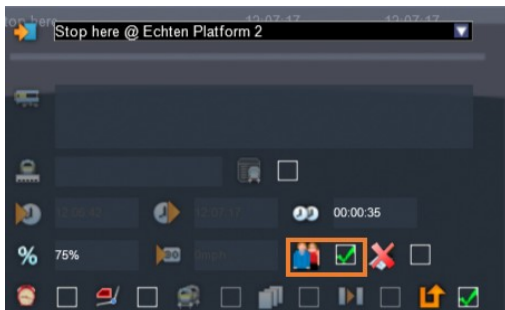


Figure 28. Set the consist as empty in the Pick-up passenger or stop at instruction.

5.2 Advanced coupling

If you have to couple a lot of rail vehicles, the list with rail vehicle numbers will be very long. There is a simple trick, you can add a descriptive text for the consist to couple, for example, you say “whole train” or “all reefers”. In the example (Figure 30 and Figure 29) have been added. The first only specifies rail vehicle numbers, the second also specifies a description. If you want both the numbers and the text, check the checkmark nr 2.

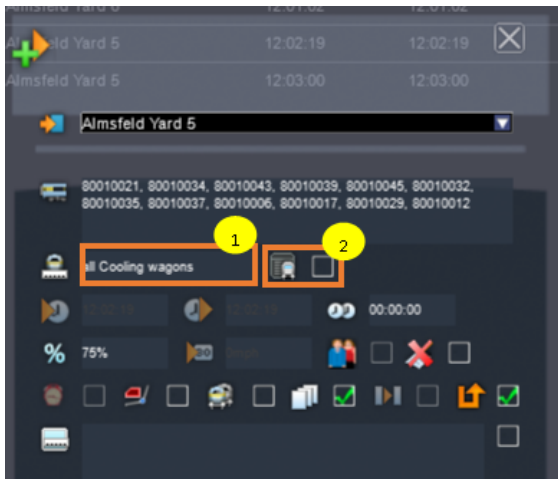


Figure 30. Define a descriptive text for a coupling instruction.

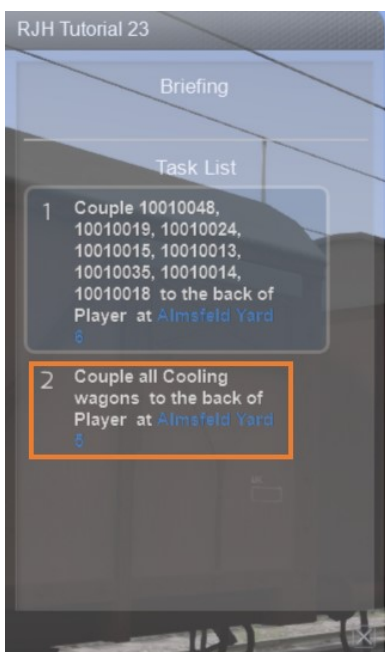


Figure 29. The resulting briefing.

5.3 Advanced marshalling

Marshalling instructions also will support the naming as described in the previous chapter.

5.4 Instruction order

There is a checkbox labelled instruction order. If you uncheck this one, you probably will be able to complete (shunting) instructions in any order. I did not test this function yet, but it may be interesting, especially if you combine several marshalling instructions in one scenario (Figure 31).

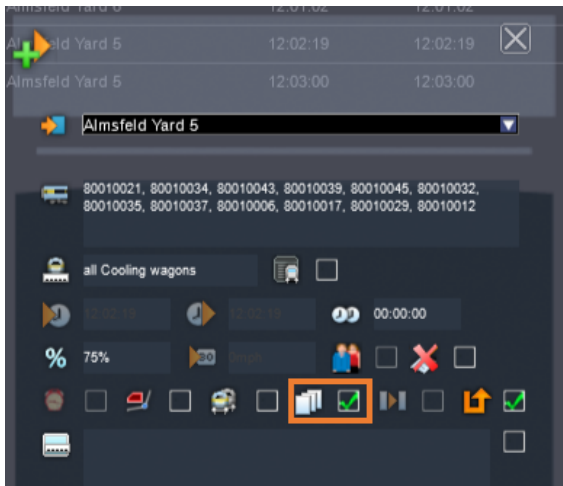


Figure 31. Uncheck this to get a free order to complete instructions.

5.5 Hidden instructions

You can hide instructions, so they are not visible in the work order. To do so, set the appropriate check mark in the stop at or pick up passenger instruction, as shown in Figure 32. Unfortunately there is a bug. Trainsimulator calculates the distance to the next station starting at the hidden instruction. This means it is less hidden than you would like and also it can be confusing. You can use a LUA instruction to unhide a hidden instruction. You cannot cancel a hidden instruction, or ignore it and declare execution as success (at least not as far as I know). This makes usability very limited. It would be nice to be able to build a “Stop at request” function as an example.

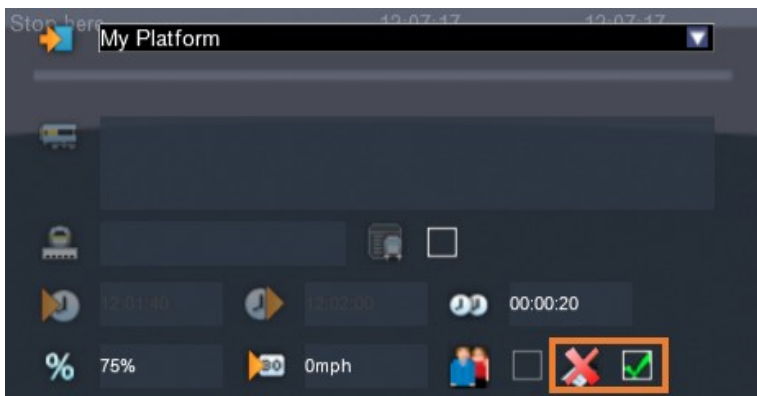


Figure 32. Check mark for hidden instruction.

6 More about scenario types

6.1 Change the scenario type

It is possible to change the scenario type. To do so, you need to edit the ScenarioProperties.xml file, which you can find in the folder of the scenario (see also chapter 14.1).

In the text editor do a search for the text

```
<ScenarioClass
```

You see a text line that looks like

```
<ScenarioClass d:type="cDeltaString">eTimetableScenarioClass</ScenarioClass>
```

You should replace the text eTimetableScenarioClass with the text that represents the desired scenario class.

The table shows the available values:

| Scenario class | Text |
|----------------|-------------------------|
| Standard | eStandardScenarioClass |
| Timetabled | eTimetableScenarioClass |
| Career | eCareerScenarioClass |

Note: You should not try to change the scenario class for freeroam and quickdrive scenarios.

6.2 Scenario marker location

I recently learned why it is important to locate the Scenario marker near the starting point of the scenario. The scenario marker causes the scenery to load near the marker before the scenario starts. If the marker is at a different location, scenery will be loaded after you start the scenario. This takes some time and is visible for the player.

Unfortunately there is no easy way to move a scenario marker over a large distance.

Note: In Part I it is explained how you can create and place Scenario markers.

6.3 Timetabled scenarios

There is no essential difference between Timetabled scenarios and standard scenarios. Some people report that you can use a timetabled scenario to create a scenario without a player engine. Then make the engine stop at the platform. You can then jump onto the train (like in freeroam) and drive it as if it is a player train. This seems NOT to work for a standard scenario.

Otherwise, you can add timetabled functions to all trains, which I will explain now in more detail.

You can add timetabled properties to the instruction type Stop at or Pick up passengers. To do so, check the checkbox (nr 1) as shown in Figure 33. Now you see you can edit arrival time and departure time, in addition to the stopping time duration (nr 2). If you change one of them, the others will change as well, to keep it consistent. If you arrive early, the stopping time in the HUD will be increased, to prevent early departure. For freight, this does not work, so you should give an instruction about the expected departure time. Eventually, you can add scripting logic to check if the player obeys these orders (you can block the throttle, or punish the player, or even end the scenario right away, see Part III of the guide for options to do this).

In standard or timetabled scenarios, the scenario fails if the player does not meet the timetable requirements. In career scenarios, the scenario will succeed, if the player ends with at least 1 point.

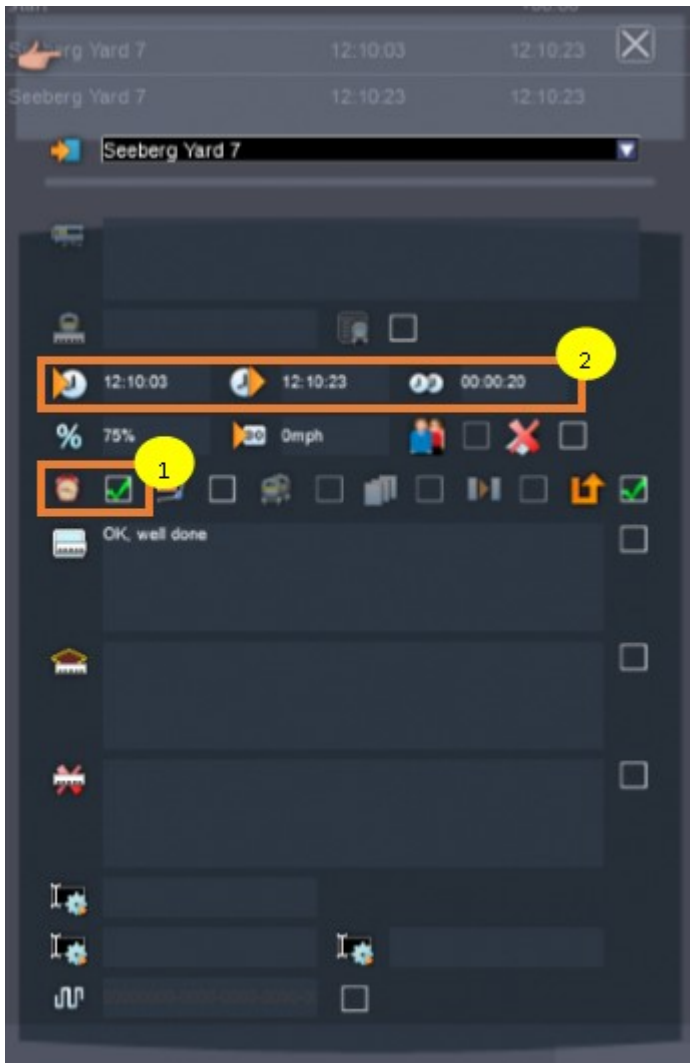


Figure 33. Editing time tabled properties

6.4 Career scenarios

You create a Career scenario, by selecting a Career scenario marker. In the Scenario Properties dialog, you will get an additional icon, a blue train symbol. Clicking at this symbol opens another tab in the Scenario Properties dialog (Figure 34). Here you can edit all rewards and penalties. The good news is that you can turn individual penalties off as well. It is up to you. You also can set the minimum requirements for the medals.

Some comments:

1. You only can award points for timetabled instructions.
2. There are a lot of opportunities to punish the player (something I don't like, I prefer rewarding good behaviour).
3. The player will not get any XP increase for scenarios not issued by DTG.
4. Career scenarios have different success criteria. They succeed always if you get at least 1 point and do not make operational errors. But if you pass a signal at danger, the scenario fails and ends. For standard scenarios you can select the behaviour in the game play settings.
5. If you turn of the career rating option, you will not get medals or XP. No points will be shown, but the success criteria still match career scenarios.

6. For user created content, career scenario will award points, but your experience will **not** be increased.
7. In addition to the default success criteria, you can use LUA instructions to make the scenario fail.
8. You need to divide the awarded points yourself to make up to 1000 points
9. You can set initial points e.g. for rounding.
10. Keep in mind that other players will have less experience when they start playing your scenario. I plea for realistic time tables, allowing to keep well below speed limits.
11. For a long scenario, create a reset point, where the player will arrive a few minutes early. It is very annoying if you cannot get any points if you are late for the first stop.



Figure 34. Edit the career rewards and penalties here.

6.5 Advanced freeroam scenarios

In part I of the this guide, simple freeroam scenarios have been discussed. Personally I have never been very interested in freeroam scenarios, until one of the readers of my guides pointed out that you can do a lot more with freeroam scenarios. It is possible to add a timetable and it also is possible to add AI. Why would you want to do this?

1. To provide an alternative to Quick drive scenarios. Quick drive is a very appealing concept, create a system in which you can place trans and drive them, change the weather, time etcetera. Unfortunately you may end up having US trains running at Dutch track. For a realistic drive you need to supply a lot of information about each consist and the route. Freeroam is less flexible but allows more control.
2. If you want to use marshalling, automatic junctions may cause problems. Once a player passes an automatic junction he or she is no longer free to choose his way back to the yard. Since freeroam makes all junctions manual in principle, you can avoid this problem.

3. If you want to drive two different trains in a scenario, or if you just want to see the train arrive and get in it, freeroam is an option.

You can achieve the last point also with a time tabled scenario.

During my experiments, I discovered a few things to consider when using freeroam as a “standard” or “timetabled” scenario:

1. You cannot set a consist in advance as a player consist.
2. Each consist initially will behave like an AI train. So, if you define instructions it will leave right away. A solution is to set the start time for the player consist 1 minute later than the start time of the scenario.
3. A **stop at instruction** at the starting location always will fail. You can use a **pick up passenger instruction** for passenger trains at a platform, though.
4. If you change driving consist during a scenario, it is unlikely the scenario will end with success. Maybe you can set success or failure using LUA scripting, but I did not try that.
5. I recommend to inform the player to click the train to drive.

6.6 Quick drive scenarios

Note: the text in this chapter is originally created by [maxtedrw](#) (Richard Maxted) and published at the [UKTS](#) community site. I am very grateful for the kind permission to use this text for this guide.

The length of this chapter reflects the complexity. The topic on creating Quick Drive Consists is barely touched, and the lack of good tooling does not really help. So, you are warned if you want to give it a try.

6.6.1 Basic principles

In Trainsimulator you can set up Quick Drive Scenarios entirely from within the program, but a number of principles apply:

1. A quick drive scenario defines three important facts. The place where the player's train is located which is called the **Starting Location**. The place where Trainsimulator will end the scenario with a "well done" message is called the **Destination**. How the player travels between the Starting Location and the Destination is defined by the **Route**.
2. Every Quick Drive Scenario is unidirectional. For every Starting Location of the **Player Consist** the scenario lists the potential Destinations from that particular Starting Location in one direction only. This is why at least two Quick Drive Scenarios are needed per route. One going up the line and a second coming down the line. AI Consists are, however, unidirectional in any direction.
3. A Quick Drive Scenario must contain at least one Player Consist Starting Location. It may contain many possible Starting Locations. You can also set up many Starting Locations within a Quick Drive Scenario which may have their own Destinations or share Destinations, provided the direction of travel remains the same from each Starting Location. Every Starting Location must have a Destination.
4. Travelling from a Starting Location to a Destination you can stop at any intermediate station provided it is on the Route. This applies to Player and AI Consists. Static Consists do not have direction of course.
5. When you build a Quick Drive Scenario you can set up many possible Destinations at which to end the scenario from a single Starting Location.
6. A Quick Drive Scenario is a **route plan** not a timetable. Like other scenarios the Quick Drive Scenario contains details of the route you take to each Destination. This means that it sets automatic points to get you or the AI Consists to the Destination or can take instructions in how manual points are set.
7. A completed Quick Drive Scenario can be edited in the Scenario Editor.

8. Quick Drive Scenario can be cloned which is absolutely brilliant for routes which have many Static Consists or for (as we shall see) doing some interesting things with Eras.
9. You are not limited to just an up and a down Quick Drive Scenario in a route. I have tested adding many more that give you additional specialised stopping and starting locations such as particular sidings / platforms / goods yards etc.

6.6.2 Planning

The key to a successful Quick Drive Scenario is planning. You need to aim to have the right amount of variation but also keep the amount of work you have to do to a minimum. Keep it simple. When you are planning your scenario keep in mind these key facts, based on the scenario properties listed above:

1. Your choice of Starting Location listed in the menu does not need to be a particular platform. It is best to think of a Starting Location as "Coniston" rather than "Coniston Platform 1". The way you define the Route to a Destination will set your Starting Platform.
2. In a similar way your choice of Destination Platform will be governed by the Route you choose. Think of your Destination as "Foxfield" not "Foxfield Platform 2"
3. You can always stop at intermediate stations on the way from your Starting Location to your Destination. How few Starting Locations and Destinations can you get away with yet keep the scenario interesting ? The temptation is to think that you may want every station on your route as a Destination. In practise are you really going to start from a halt in the middle of nowhere ? Is it ever going to be your final Destination?
4. Are you going to need to Route via particular platforms or loops on the way from your starting location to Destination. If so, you are going to have to set these up as part of the Route either by specifying them as Destinations that you go via or (if possible) by changing manual points.
5. Because you can now edit your Quick Drive Scenario from the scenario editor the kind of detailed planning I used to recommend is no longer needed. However, it is worth setting out the basics first.
6. You will need to create a PreLoad metafile for your route, I am not sure that Quick Drive Scenarios work without this information. This can be done using RWTools.
7. It is vital that (to control the AI and Static Consists) you choose a date for your scenario. The year of your scenario will help to limit the "pool" of consists who have your Route as a Valid Route to a selection of stock which match each other in time by their choice of Era Parameter.

6.6.3 Creating a Quick Drive Scenario basics

Open the scenario editor and choose "Create new quick drive scenario" for your route. Name it something sensible like "QD Up 1958". It is useful to include the year for the scenario right here in the title as well.

Note: The image shown does not correspond with the text. You may prefer to open an existing scenario, fly to the Starting Location of your Quick Drive Scenario. This is explained in Part I of the Authors Guide and avoids the problems described in chapter 6.2. Use the Scenario template marker as scenario marker.



Figure 35. Scenario Properties for a Quick Drive Scenario

Open the left hand flyouts and you will see the top flyout has the scenario choice selected. The next grey window down has a signal icon which you click. The various markers

- Marker - AI Spawn Point
- Marker - Consist Spawn Point
- Marker - Player Spawn Point

are the ones we will be using.

6.6.4 Setting up The Player Spawn

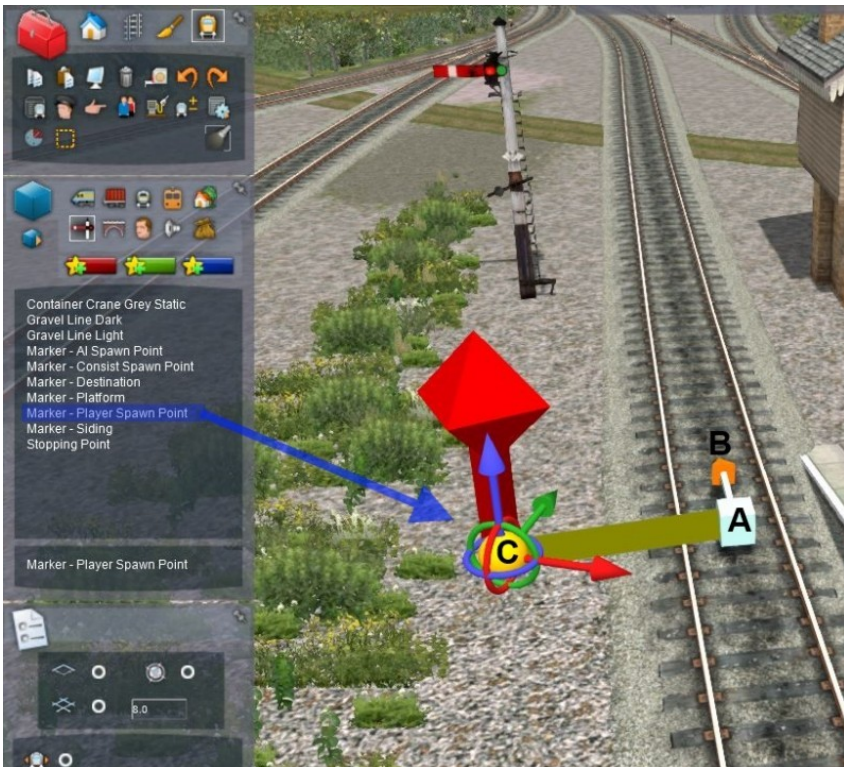


Figure 36. Player spawn marker which is the minimum requirement for a Quick Drive Scenario.

To place the **Player Spawn Point** you select and pop the marker (C) on the ground (Figure 36). As soon as you do the arrow (A) shows up and you place this at the point you want your train to be spawned from. Initially the arrow is double headed. How you place C is not important except that you are going to have select it again and so hiding it in bushes, trees, platforms isn't a good idea. Place it on some clear level ground.

The **Arrow** is the vital bit. You can fine tune the arrow by clicking on C then on the Arrow at A, it will go sort of yellow. It can then be dragged to just the right spot. The arrow (B) is key. The arrow must face in the direction you wish the consist to travel. The Player consist is placed by Trainsimulator on the box A and moves away from it in direction B.

Now click anywhere else and you'll see the Arrow disappear.

To set up where the player consist is going you need to select the **Arrow** again (hence keeping it in an easy location) by a **double click** so the flyout panel on the right shows up. This is worth pinning open using the little pin symbol in the top right hand corner (Figure 37).

Note: You may need to try several times at varying double click speed. Click on the marker, not on the Arrow.



Figure 37 Player Spawn Point properties dialog

For all spawn objects the items highlighted in the red box do not need editing.

Firstly you need to fill in Box A. This information is what appears in the right hand panel when you are choosing the Route in the Quick Drive options. If you notice you will see that I have typed "Foxfield (1952)" (without the quotes).

Why set the date ?

Putting the same date in Box A as the scenario date set earlier is an elegant way of producing Quick Drive Scenarios that are tied to a particular era. In combination with good Era specification in your Player, Static and AI consists this will limit the selections of consists made by Trainsimulator in your scenario.

However, the neat part about it is that once you have set up a Quick Drive Scenario for one era you can clone it, edit the scenario dates and those of the Player Spawns in Box A and this will give you alternative scenario start dates at each player spawn location available when you choose the route in the Quick Drive options menu. What date you choose will then control what AI and Static Consists appear.

As an example on the Coniston I have a Quick Drive Scenario I set up for 1958 that limits the AI to Blood and

Custard and Steam. I then cloned it and changed the date to 1963. I now have also got the option to drive the Route in 1963 as well with green Diesels and Ferret & Dartboard Steam locos but without having to do a whole new Quick Drive Scenario from scratch.

6.6.5 Setting the Route for your train

The first thing is to ensure your Player Consist is properly placed. Imagine that you want to place the consist so the loco is by the starting signal with carriages stretching back down the platform. You will want to place the Spawner so the Arrow in Figure 36 (B) points in the direction of travel and the Box of the Arrow (A) is where the front of the loco is. To change the direction of the Arrow click on it holding down the [Shift] key.

You need to set the flyout menu option F to **Front**. The menu option sets which part of the train is located over the Box B of the arrow.

In case of a terminus station you will want to set Box A close to the buffers with Arrow A pointing up the route and set the value of F to BACK so the back of the train is what is over Box A.

Use the **Centre option** available in selection F for loops and island platforms where you want a train to sit nicely close to the station building.

The next bit is a little fiddly

Click once in the main editor screen and then press the 2D map selection [key press 9]. You may find you lose the right hand flyout at first but with a little practise you will be able to get the route map and the right hand flyout to show at the same time.

Find your first possible destination for trains spawned from your player spawn on the map. Click the Green "+" marked B on the pic above and click on a location on the map. This will populate boxes C and D with the name of the Location. In the example you can see I clicked on Broughton Up platform. The destination (Box E) comprises the name that TS2014 uses taken from the route marker placed by the route author and a Friendly Name that is what will appear in the right hand side list of destinations when you are selecting the Quick Drive Route to use.

Don't forget that the contents of E is a list of possible Destinations from a player starting at point specified in A. You can stop at any location on the Route of the Quick Drive Scenario. In this example four possible Destinations are created. The order does not appear to be important in the Player Spawn.

6.6.6 The Via Command

The **Via Command** allows you to specify a routing. In this case any train departing from this player spawn will go via Broughton Up platform. This is a powerful way of controlling how trains go round loops and junctions. However, it is not wholly reliable. It works well for routing with manual points but can throw the odd wobble with automatic points.

6.6.7 Adding more Player Spawns to your Scenario

A good Quick Drive Scenario will give you a number of possible Starting Locations on your Route, each with their own list of Destinations. Many of these will be the same as you put more Starting Locations along the route. Remember that a Quick Drive Scenario is unidirectional. Player Spawns will either be set up working up or down the route. Hence the minimum sensible number of Quick Drive Scenario s for a route being two. Each player spawn follows exactly the same procedure outlined above.

6.6.8 User friendly naming scheme

Going back to the items in box E you will see that at Coniston I have a friendly name of Coniston. For this player spawn Coniston = Coniston Platform 1. I have also set up a Player Spawn further up the route at Torver that has Coniston Platform 2 as a destination (just to add interest). But I still use the Friendly Name of Coniston. What this means is that when I select (in the Quick Drive Menu) a starting place of Foxfield in the left-hand panel I am offered Coniston as a destination in the right-hand panel. If I select Torver as a Starting Location my list of Destinations in the left hand panel does not change but because I started at Torver the route is set to Platform 3 and not 1 at Coniston.

This is a way of

- a) adding a bit of clarity to the Quick Drive Menu
- b) adding a sort of unexpectedness to a scenario.

There are some horrible examples of Quick Drive Scenarios which have dozens of possible Destinations at any station none of which are available from more than one particular starting point.

This is a good point for a first test of the Quick Drive Scenario.

6.6.9 Static Consists

Get all the Static Consists and AI consists placed first. Clone the route afterwards and remove the Player Spawns and replace them with Spawns for the in the opposite direction.

Static Consists have their own Spawner which is placed just like that of the player with a marker and arrow.

The options are different as shown in Figure 38.

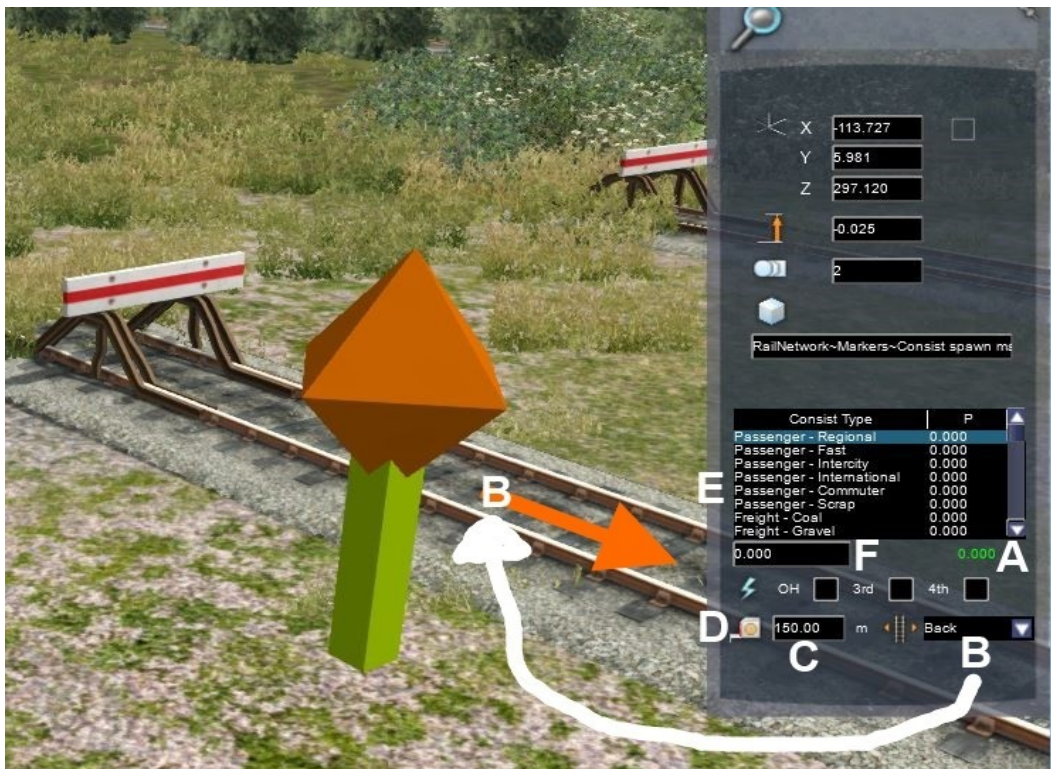


Figure 38. Static Consist Spawner Point

In the example you can see a Consist Spawner whose Arrow is pointing forward from its block near the

buffers. This means that we must be selecting **Back** in the option B so the back of the Static Consist is over the arrow's block with the consist pointing forward in the direction of the arrow shown.

Getting your head around this is quite important. Not least as CENTRE as an option in B is very useful for static consists. If you place the marker arrow inside a goods shed (leave the marker outside to make it easy to select) and then choose CENTRE in option B it ensures that no matter what length the Static Consist chosen by TS2014 the goods shed will always be occupied.

6.6.10 Using the tape measure

In the topmost left hand side flyout (Figure 39 you will see a tape measure). Find a bit of clear space and click this, click anywhere and holding the mouse drag the white line that appears out. Down in the bottom of the screen you will see a measurement in metres.



Figure 39. Using the Tape Measure.

In Box C of Figure 38 you need to set up the maximum consist length that can be spawned from your Static Spawner. The tape measure enables you to measure the siding so you don't get trains hanging over points etc. Box C can also be used to limit things. So, for instance, you can set quite a short distance here so that only a Tank Loco is spawned rather than a 9F.

6.6.11 How to Decide what is Spawned

The Contents of Box E in Figure 38 should look familiar if you have been busy building Static Consists. These are the various categories of things that can be spawned. The numbers on the right hand side refer to the probability of something being spawned. A probability of 1 is certain and of 0.5 is about as likely as unlikely.

When you click on a Consist Type in Box E you can type a probability in Box F (don't forget to hit [enter] after typing to set it into box E). The value shown at A is the total probability of a static consist being spawned at this location. It can never exceed 1. It can however be less than 1. All this means is that there is a probability of nothing being spawned. A total of 0.8 means a 20% chance of nothing being spawned.

This means that you can choose a single consist type or multiple types each with their own probability provided that the total in A does not exceed 1.

And that is about it except

The Joy of Era

The really cunning thing is that by specifying an Era for your consists and a date for your Quick Drive Scenario only stock whose Era covers the date of your Scenario will be spawned here. And even more only those stock whose ValidBuildRoute is set to this route.

This means that the same Static Consist Spawner can be used in multiple clones of a Quick Drive Scenario and it will spawn different stock correct to fit the date. If I select Foxfield - 1958 as my Starting Location this

Static Spawner will produce either 5 plank trucks or Conflats. If I select Foxfield - 1974 as my Starting Location it will select appropriate stock for this era.

6.6.12 AI Spawn Points

Unlike AI in a standard scenario the AI consists in a Quick Drive Scenario do not have a timetable as such. You can set the Route but they drive when you are around. How "around" you have to be for them to start working is something I am not sure about. I am also not sure yet what the "You" is. I think I have managed to get an AI Consist to start working by looking at it in Free Cam Mode (key press 8) but I am not sure yet.

Certainly your Player Consist triggers their activity.

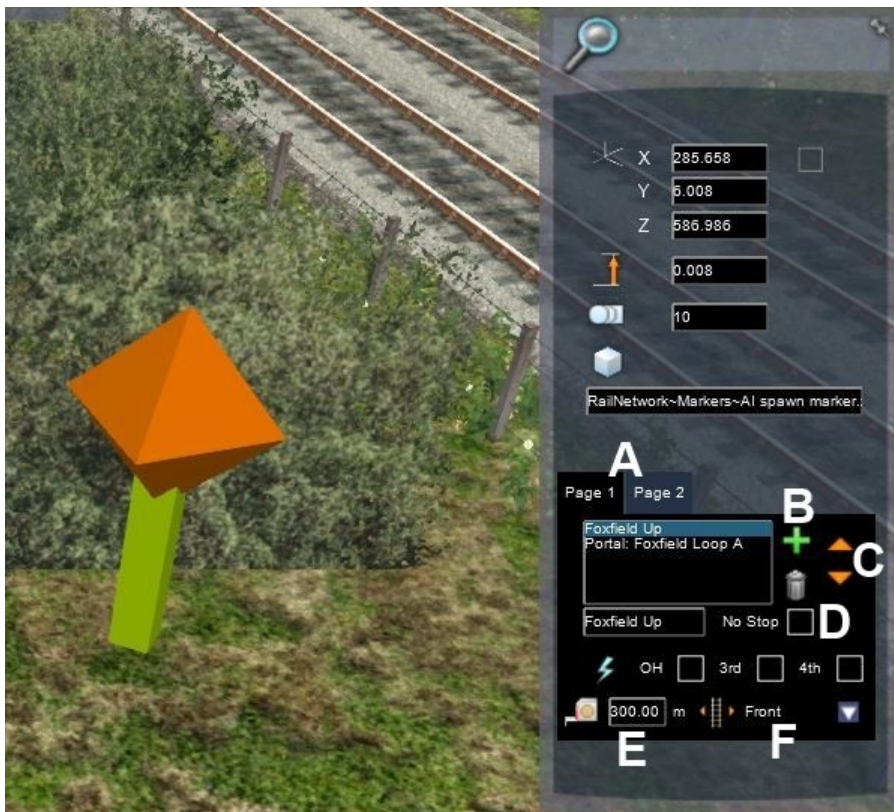


Figure 40. AI Spawn Point dialog showing the first tab.

The other thing to be aware of is that consists spawned by the AI Spawner will give you priority. To have passing traffic in loops on single line routes takes a little "jiggling" but is possible.

In many respects AI Spawners are a sort of cross between a Static Spawner and Player Spawner. AI Consists have their own Spawner which is placed just like that of the player with a marker and arrow. However, the options are different.

In Figure 40 you can see an **AI Spawn Point** and the first of two pages of information you need to give it. This page looks just like the Player Spawn Point. Information in field F determines the part of the consist that is located over the placement arrow block described earlier. Just as before this allows you to place the front of the train exactly or the back or the centre.

One of the things to remember is that an AI spawner starts working when you are near it (about half a mile away I think). So you do not need to place it at a signal or at any particular location. A bit of trial and error is sometimes needed so that it can be placed far enough away that the AI consist is moving by the time you see

it but near enough to be triggered and for the AI consist to do the thing you want when you want it i.e. pull into the station, pull away from the signal etc...

6.6.13 Routing your AI

You set the locations in Box C in just the same way as the Player Spawner by clicking on the map. The exception is that you only need a Destination that is out of sight. More places in this list set the route if it is not a simple A to B thing or if you want the train to stop. In this example I have added Foxfield Up and left Box D unticked. This forces the AI to stop at the through platform. For an express train I would tick this box, or if it was just a routing through a loop.

The routing is linear and follows the instruction list in order. TS2014 sets points etc. to achieve this. You can have some fun with this. I have had a light loco start in a head shunt, go forward to a loop, stop and then reverse back into an engine shed.

6.6.14 Selecting the AI to be used

The AI available to be used is governed by the ValidRoute in your Consists and by the Era of your Quick Drive Scenario falling within the Era of the Consist. As before this gives you the ability to use the same AI Spawners in Quick Drive Scenarios of different eras generating AI traffic appropriate to that period. Another selecting factor will be length of the consist set in Box E. If you are not bothered then make this



Figure 41. Selecting AI train types for Quick Drive Consists

large - but be careful if you are spawning from a siding or a loop platform if you are not to block the player route. The measuring tape is your friend.

The second page (Figure 41) of the AI Spawner is used exactly like the Static Consist. It determines the probability that your AI Spawner will fire off a particular kind of consist.

A Few Wrinkles

If you are careful it is possible to have AI traffic fight it amongst themselves. A goods train can leave the goods yard followed by a commuter train arriving before an express routes around it. However, I have also managed to generate a fatal error doing this.

6.6.15 Goods Yards

All of this also applies to Goods Yards and the like. It is possible to set up separate Goods Quick Drive Scenario to create a Route between goods and other yards.

This can be prone to issues if you are not mindful that a Via command applies to all traffic which is generated by a Player Spawner. For routes with automatic points this can mean you being sent to yards you don't want to go to. But for manual points this is brilliant as all you need to do is pick a particular obscure destination and then you can shunt in the yard all day long without the Quick Drive ever terminating and with a merry flow of AI traffic behind you.

6.6.16 Creating Quick Drive consists

QuickDrive scenarios use Quickdrive Consists to enable you to drive along routes populated with static and AI stock. You are driving the player consist. Stock which never moves are **Static Consists**. Trains which can move under their own power are **AI Consists**.

Quick Drive Consists are set up in the **Preload Directory** of most Assets. Every Preload directory is "Live" all the time for every route. That is (unlike other Assets) you do not need to actively select Preload Directories to be considered by Trainsimulator when running a Quick Drive Scenario. A poorly crafted Quick Drive Consist can infect many routes, e.g. by spawning a narrow gauge loco in the middle of the West Coast 2010 electrified main line. Technically any Quick Drive Consist can be used on any route.

There are several ways to set up Quick Drive Consist. RwTools is one and the other is manually.

Note: Using the built in consist editor Trainsimulator is not recommended, because it has very limited options for settings to avoid problems as mentioned before.

The way in which consists are used in a Quick Drive Scenario is entirely determined by interplay between the contents of the Quick Drive Consist and Quick Drive Scenario.

Fragments

There are many fields in a Quick Drive Consist definition, all of which now play a really important role in controlling how you can make realistic Quick Drive Scenario s.

The first decision is over **Fragments**. A Quick Drive Consist whose filename ends with "...fragment.bin" is not used directly in a Quick Drive Scenario. A Fragment is used as a building block for complex Quick Drive Consists. For instance I may want to have a choice of several different locos but I am happy for any of them to pull along five Mark1 coaches. If I set up a fragment of five Mark 1 coaches I can then create a whole bunch of Quick Drive Consist s which specify the loco and the MK1 fragment rather than retype all the coaches each time. This also is very useful for goods trains where I can set up "5xBoxVans Fragment" (note the vital space before Fragment), 5xCoalTrucks, 5CoalTrucks weathered etc. Having got the building blocks in place it is a lot easier to make many and varied freight trains.

The LocoName and DisplayName parameters

These fields only apply to Player Consists. They can be left blank for Static Consists or for consists which are only going to be AI Consists. In fact they must be left blank if you are writing AI Consists that you do not want to show up as player enabled at all. (More of this subtle effect later)

The loco name is an excellent field for grouping things together. For instance I use a **LocoName** of Class 37 and the **Displayname** to describe what kind of Class 37 (BR Green) and what train (10 mixed goods). The LocoName is what controls what is displayed in the select loco panel on the Quick Drive Menu. The DisplayName is the list that is subsequently displayed.

You could actually completely invert the system for consists - LocoName=BR Maroon Mark 1 and then Displayname would be used to show all the different sorts of locos used to drive BR Maroon Mark1 Stock. This works but it is a slightly odd way of doing things.

EraStart and EraEnd Years parameters

Much undervalued but probably the most subtle parameter in the whole toolbox. There was some sceptics who felt that this did not work but it does and can have a really impressive effect on a Quick Drive Scenario. Do not be tempted to enter 1928 to 2050 as I have seen done. It is worth getting this as near correct as possible as it really can make or break a scenario. The Era parameter works at the Scenario and not the route level.

ValidBuildAndDrive parameter

This preselects the consists that Trainsimulator makes available to any route. It works at the route and not scenario level. However, In combination with the Era it can really tie particular consists to particular Quick Drive Scenario. This additional parameter enables the Quick Drive Consists to be used selectively.

DrivableConsist parameter

The value **eTrue** means you can drive this consist and so can the AI engine. Only eTrue here produces drivable consists. This, however, contains a hidden danger. Trainsimulator treats all Player consists as potential AI consists and will use them as this given half a chance. This is how you end up with poorly designed consists appearing in scenarios when they are the wrong motive power, period or even gauge. This is why setting the route and era is so important.

eFalse means it is a static consist. It is worth making the naming of your static consists obvious. I put "Static_" at the front. eFalse consists do not show up in the QD selection menus. Thus it is possible to make Static Consists of just trucks or coaches by using eFalse in this parameter.

ConsistType parameter

The **ConsistType** parameter controls what Static & AI consists will be chosen by Trainsimulator to be spawned on your route at a specific location. Player Consists do not need this parameter. But all Player Consists can also be AI consists, which need this parameter to be set. There are lots of types here some of which (PassengerType and FreightType) are pre-set as well as Custom1 to 8. The temptation is to use Custom1 to 8 for your route to stop accidental spawning of generic consists. Except that this doesn't work. In fact it is the worst possible solution as Trainsimulator makes all Consists available to all routes and controls selection of them by the ValidRoute and Era parameters.

If the era and valid route are set correctly you can actually use the generic ConsistType to broaden out what you see on your Quick Drive Scenario. The ConsistType parameter is best thought of as location specific control on a set route in a set era. It gives huge ability to target the final "look and feel" of a Quick Drive Scenario to, for instance, the sheep pens at Arkengarth Station in 1958. It ensures that only livestock trains of the correct vintage will be spawned here.

There is a final very powerful (and misused) ConsistType. "**DoNotSpawn**" is best used for consists that you

want to be only Player Consists. The most obvious examples are some of the more complex locos such as the 3F Steam Loco in advanced mode which fails if used as an AI loco. Here the player advanced loco needs to be in a "DoNotSpawn" ConsistType.

Note: AI loco only can be set using (i.e. do not make it available as a player choice) by setting a ConsistType, DrivableConsist is eFalse, and the DisplayName and DrivableConsist are left blank.

In summary

StaticConsists - stock that goes nowhere displaying at a fixed location. Which stock is used on the route is controlled by ValidRoute. Era and ConsistType determines which stock, that is valid for the route, is enabled in a QuickDrive scenario. You do not LocoName or DisplayName and DrivableConsist should be unticked/eFalse. A static consist can use fragments. A static consist does not require a locomotive.

PlayerConsists - stock you can drive. Any stock can be used on any route in any era. DrivableConsist must be eTrue which means all these stock will be used as AIConsists unless the ConsistType is "DoNotSpawn". To be able to select consists LocoName & DisplayName are required.

AIConsists - stock whose movements are controlled by the QDScenario. Which stock is used on the route is controlled by ValidRoute. Era and ConsistType determines which stock, that is valid for the route, is enabled in a QuickDrive scenario. You do not LocoName or DisplayName. DrivableConsist is eTrue.

Scenario chaining

6.7 Relay scenarios and scenario chaining

In Trainsimulator 2013 **Relay Scenarios** were introduced. The idea is that you play part of the scenario and then pass it to another player.

There is a Relay Scenario marker, but:

1. Nowhere is documented how to create Relay scenarios
2. Relay scenarios do not appear in the scenario lists, though you still can access them in the editor.
3. The Trainsimulator manual does not mention them anymore.

DTG confirmed to me that Relay scenarios are no longer supported, so you should not try to create them.

The same holds for **scenario chaining**. This construct allowed you to start a scenario upon successful completion of another scenario. You can still create them, but they are not working anymore.

6.8 Railfan scenarios

In Trainsimulator 2016 DTG introduced Railfanning scenarios. A Railfan Scenario is essentially an standard or career type scenario. To create one, you need to perform following steps:

1. Add a short piece of invisible track, or use an existing siding.
2. Create a Destination Marker if needed at a location that does not block the traffic you want to show.
3. Put an (invisible) player engine on the track.
4. Give the player engine a driver and the final destination defined in step 2.
5. Give the player engine a trigger event for the LUA script.
6. Add a LUA script that terminates the scenario after X minutes. You need to compile this script.
7. Add all AI trains to the scenario.

NB The **LuaCreator** (see 10.2) program can help you with a complete skeleton for a Poor Man's Railfanning script.

6.8.1 Add invisible track or use a siding

In the Railfanning scenarios DTG created, invisible track is used. You can find this track in the folder **Assets\...\RailNetwork\Track\Inv_road.bin**. Unfortunately you cannot use it in the scenario editor, which means you have to change the route to lay this track. For the tech guys and girls, the Blueprint for track sections does not support the field "ValidInScenarios".

You only can use existing track. Best is to choose a siding, where you have room to add a destination marker.

6.8.2 Create a destination marker

Though you do not intend to actually drive the player train, it needs a final destination. You can eventually add a Destination Marker (see 2.2). Make sure to avoid blocking other traffic, so keep it at the siding, or you must give the player train the lowest possible priority to keep the AI traffic running.

6.8.3 Put a player engine at the track

The next step is to place a player engine on track. If you bought TS2016, with each of the routes that support Railfanning, you will find this loco: **Rainfan Invisible**. In the example I activated the Riviera 50s route (called ExeterKingswear) and then got the engine. See Figure 42.



Figure 42 Selecting the Railfan Engine

In the editor it will look like a camera.

6.8.4 Instructions setup for the player engine

Now you can set up the player instructions. It needs a driver, final destination and a trigger instruction to start the script.



Figure 43 Railfan “engine” with Driver instruction

Make sure to set the trigger event as shown in Figure 44.

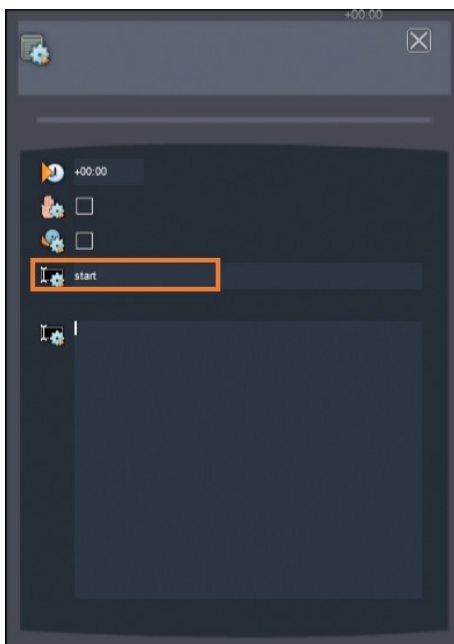
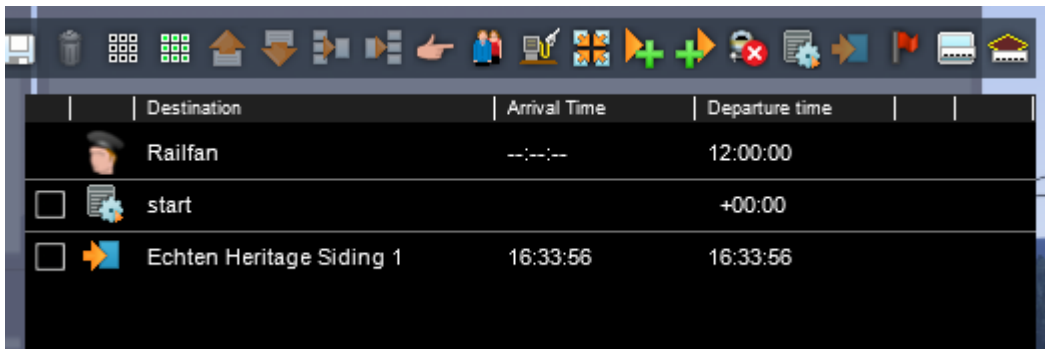


Figure 44. Make sure to enter “start” as the trigger event value.

The resulting instructions should look like Figure 45.






| | Destination | Arrival Time | Departure time |
|--|--------------------------|--------------|----------------|
|  | Railfan | --:--:-- | 12:00:00 |
| <input type="checkbox"/>  | start | | +00:00 |
| <input type="checkbox"/>  | Echten Heritage Siding 1 | 16:33:56 | 16:33:56 |

Figure 45. Instructions for Railfan scenario

6.8.5 Insert the LUA script

Now you need to insert the LUA script. Maybe the easiest way to do so, is to use the script snippet that comes with the LuaCreator program and use LuaCreator to install and compile the script.

You also can do this manually, which is described in part III of the Scenario Authors Guide.

The script may look like it is shown below:

```
--[[
Common script snippet

You always must include this App because it contains common functions that are
used by many Apps

(C) 2015 Rudolf Heijink
]]--

-- Identification
SnippetVersion={} -- create an array to hold version numbers for each module

SnippetVersion["Common"]="1.0"

-- Constants

-- debugging support using -LogMate option and Print() statements, default off
DEBUG=false

-- Booleans
FALSE = 0
TRUE = 1

-- Vertical positions for message boxes
MSG_TOP = 1
MSG_VCENTRE = 2
MSG_BOTTOM = 4

-- Horizontal positions for message boxes
MSG_LEFT = 8
MSG_CENTRE = 16
MSG_RIGHT = 32

-- Combinations
```

```
MSG_TOP_RIGHT = 33
MSG_TOP_LEFT= 40
MSG_CENTRE= 18

-- Sizes for message boxes
MSG_SMALL = 0
MSG_REG = 1
MSG_LRG = 2

-- Simple message types
MSG_INFO = 0
MSG_ALERT = 1

-- Pause no pause option for messages
PAUSE_GAME = 1
DONT_PAUSE_GAME = 0

--Test condition return values
CONDITION_NOT_YET_MET = 0
CONDITION_SUCCEEDED = 1
CONDITION_FAILED = 2

-- Speed units definitions
KMPH=1
MPH=2
KMPHText="km/h"
MPHText="Mph"

-- Calculate km/h or Mph for meters per second
MPStoKMPH= 3.6
MPStoMPH= 2.23693629

--Video parameters
FULLSCREEN = 0
FRONT_AND_CENTERED = 1
VIDEO_CALL = 2

PLAY = 1
PAUSE = 2
STOP = 4
SEEK = 8

-- Locking controls

LOCK=1
UNLOCK=0

-- Syscall strings
PlayerEngine="PlayerEngine"
InfoMessage="ScenarioManager:ShowMessage"
ShowInfoMessageExt= "ScenarioManager:ShowInfoMessageExt"
BeginConditionCheck= "ScenarioManager:BeginConditionCheck"
PlayDialogueSound= "ScenarioManager:PlayDialogueSound"
EndConditionCheck= "ScenarioManager:EndConditionCheck"
LockControls="ScenarioManager:LockControls"
UnlockControls="ScenarioManager:UnlockControls"
RegisterRecordedMessage="RegisterRecordedMessage"

-- Logging definitions
NoLogging=0
```

```

LogError=1
LogCheck=2
LogDebug=3

-- Controls

-- Variables
RouteGUID= "6b8cb9e5-aabf-4d9c-b640-d3d24d7cfffac"
ScenarioGUID= "1f6be5ad-a65f-4a06-93f0-9073407fd7b4"
RouteName= "[WS] Seebergbahn"
ScenarioTitle= "RJH Poor mans railfanning"

logginglevel =NoLogging

-- Functions

-- the table representing the class, which will double as the metatable for the
instances
LoggingObject = {}

-- failed table lookups on the instances should fallback to the class table, to
get methods
LoggingObject.__index = LoggingObject

function LoggingObject.new(logfile,mode,scenariotitle)
    local self = setmetatable({}, LoggingObject)
    self.logfile= logfile or "logfile.txt"
    self.mode= mode or "w+" -- open mode
    self.scenariotitle=scenariotitle or ""
    self.filepath="Content\\Routes\\" .. RouteGUID .. "\\Scenarios\\" ..
ScenarioGUID .. "\\.. self.logfile
-- You can find this file in the scenario directory
    self.debugfile= io.open(self.filepath,self.mode) --file handler for debug file
    self.WriteLog(self,"Logging started for scenario
"..self.scenariotitle,LogDebug)
    self.WriteLog(self,"Filepath: "..self.filepath,LogDebug)
    return self
end

-- write to logfile
-- message is a string containing a text message

function LoggingObject:WriteLog(message, level)
    level= level or NoLogging
    if level<= logginglevel then
        if self==nil then
            self= LoggingObject.new()
        end
        local dt= os.date("%d-%m-%Y/%X ") or ""
        self.debugfile:write(dt .. message ..'\n')
        self.debugfile:flush()
    end
end

-- Support for associative arrays
-- function to iterate over a table and retrieve its index (i) and value (v)
-- Source: Programming Lua v5.2

local function iter (a, i)
    i = i + 1

```

```

        local v = a[i]
        if v then
            return i, v
        end
    end

-- function that returns the next index, value pair of an associative array
-- Source: Programming Lua v5.2

function ipairs (a)
    return iter, a, 0
end

-- function to fill associative array with index values.
-- returns number of destinations

function createIndex(a)
    for i, v in ipairs(a) do
        a[v]=i+indexStart-1
        print(v, " ", a[v])
    end
    return i
end

-- round a number to idp decimals

function round(num, idp)
    local mult = 10^(idp or 0)
    return math.floor(num * mult + 0.5) / mult
end

--[[
Poor man's rail fanning
A lua script that facilitates railfanning, even if you did NOT buy trainsimulator
2016 steam edition
(C) 2015 Rudolf Heijink
]]--

-- Identification

SnippetVersion["RailFanning"]="1.0"

-- Constants

-- Parameters

Duration= 10 -- scenario duration in minutes

RailFanning = {} -- the table representing the class, which will double as the
metatable for the instances
RailFanning.__index = RailFanning -- failed table lookups on the instances should
fallback to the class table, to get methods

function RailFanning.new(Duration)
    local self = setmetatable({}, RailFanning)
    self.Duration=Duration or 10
    self.DurationSeconds=Duration*60
    self.ConditionName= "RailFanning"
    SysCall ( "SceneManager:ShowInfoMessageExt", "Rail fanning", "This is a rail
fanning scenario. So relax, get out of the engine, and use the free camera (key
8) to look around.", 15, MSG_TOP_RIGHT, MSG_REG, TRUE);

```

```
-- Lock the reverser, so the player engine cannot drive
SysCall("PlayerEngine:LockControl", "Reverser", 0, LOCK)
-- Set camera to free camera
SysCall ( "CameraManager:ActivateCamera", "FreeCamera", 0 );
-- If you like, try to ove the camera to a specific position, using the next
statement
-- SysCall("CameraManager:JumpTo", longitude, latitude, height)
self.StartTime = SysCall("ScenarioManager:GetScenarioTime", 0)
logwriter:WriteLog("RailFanning created",LogDebug)
return self
end

function RailFanning.Begin(self, ConditionName)
    self.ConditionName= ConditionName or "RailFanning"
    SysCall ("ScenarioManager:BeginConditionCheck", self.ConditionName );
    logwriter:WriteLog("RailFanning begin checking",LogDebug)
    return TRUE;
end

function RailFanning.Finish(self)
    self.ConditionName= ConditionName or "RailFanning"
    SysCall ( "ScenarioManager:EndConditionCheck", self.ConditionName);
end

function RailFanning.Check(self)
    local CurrentTime = SysCall("ScenarioManager:GetScenarioTime", 0) or 0
    if(CurrentTime-self.StartTime >=self.DurationSeconds) then
        -- end the ScenarioManager
        SysCall( "ScenarioManager:TriggerScenarioComplete", "You are a real
Rail Fan!!!");
    end
end

-- Commands to start/stop railfanning, you need to create a trigger event to make
this work
function Initialise()
    logwriter = LoggingObject.new(filename, mode,ScenarioTitle)
end -- Initialise

function OnEvent(event)
    if event== "start" then
        railFanning=RailFanning.new(Duration)
        railFanning:Begin()
        return TRUE
    end
    return FALSE
end

function TestCondition ( condition )
    if condition == railFanning.ConditionName then
        railFanning:Check()
        return CONDITION_NOT_YET_MET
    end
    return CONDITION_NOT_YET_MET
end
```

You may need to edit the red coloured texts. Do not forget to compile the script!

6.8.6 Set up the AI traffic

The last step is to create an interesting amount of AI traffic. In Part I you learned how to do this. Some tips:

1. You may set the player performance a bit lower to slow down the trains.
2. At platforms, use the pick-up passenger instruction if you make the trains stop.
3. Add some animated scenery to make the whole scene a bit more interesting.

7 Add scenery to scenarios

7.1 Add scenery

In the scenario editor it is possible to add scenery items to the scenario. These scenery items will be visible in the scenario only. To give you some ideas:

1. Populate a cattle loading platform.
2. Add a fire or accident to the scenario.
3. Cause a track blockage.
4. Add lots of people watching a special event.
5. Add people with e.g. task to set switches.
6. Add a fair
7. Populate a railway museum
8. Add a beach party
9. Add track works

You can do a lot of interesting things, let your imagination do the work.

As with all stuff, sometimes it sounds easier than it works out in practice. Let's start with the basics and try to create a fair at the Seebergbahn. For this fair I use a number of objects created by Jeff Layfield (http://members.uktrainsim.com//filelib-info.php?form_fileid=28530). It may be interesting to use this pack as well: http://www.uktrainsim.com/freewarepacks/index.php?p=ThePacks&pack_id=Clutter1#vh which contains some market booths. It may take some time to locate interesting objects, but that is part of the game.



Figure 46. Use the triangle cube to register the provider and product of the stuff to use.

The first step is to make the items you want to use available. This is explained in detail in part I, so here only a brief reminder:

1. Select the blue cube with the orange triangle (see Figure 46).
2. Then, in the right hand form, select JeffLayf and the tick the two checkboxes (Figure 47).



Figure 47. Tick the train and eye checkbox to make Jeff Layfields items visible.

Now you can pick up an item and place it in the game. This is a bit different from rolling stock. Rolling stock always will snap to a track, scenery items do not snap anywhere. You can move items using the keyboard, especially snapping to terrain is very useful.

| Key | Action |
|----------|--|
| V | Moves the object up. |
| B | Moves the object up down. |
| G | Rotates object clockwise. |
| F | Rotates the object anticlockwise. |
| N | Moves the object along the terrain towards the camera. |
| C | Moves the object along the terrain away from the camera. |
| J | Snaps the object from its original height to the terrain height. |

This gives you a fine grained control over the object placement, but some movements are not supported in this way. If this case you also can use the toolbox (see Figure 48).



Figure 48. Tools for manipulation objects.

1. Selects a shifting tool, you can move the object in three directions.
2. Select rotation tools in three directions.
3. Select a tools that allows to change size of an object.

You find this toolbox at the bottom of the editor screen.

7.2 Make scenery objects available in the scenario editor

If you open the scenario editor, you may notice that generally not all objects are available. This has a good reason, it is unlikely that you want to place additional houses in a scenario. Therefore, the blueprint files have an option that must be ticked to make the object visible in the scenario editor.

It happens frequently that object creators forget to turn this feature on. You can solve this in many cases, using RWTools (see 10.5). It does not always work, e.g. loft items (e.g. platforms, roads, bushes, fences) and track do not have the option in the blueprint, so you cannot do the trick here.

Since a few years many asset packs are zipped in a .ap file (which is a .zip file). RWTools cannot handle .ap files directly, but requires them to be unpacked first. You can do this and it works, but the next time you perform a cache verify, the unpacked files will be deleted automatically.

Note: before attempting this procedure make sure you have a backup of all your assets.

Respecting these restrictions, you may proceed as shown here:

1. Open menu item **Edit Assets**.
2. This item has one sub menu **Asset editor**. Select this one.
3. Now select the asset folder you want to update, using the folder selector (nr 1)
4. Click **Add to List** (nr 2).

Now you see all assets in your selection.

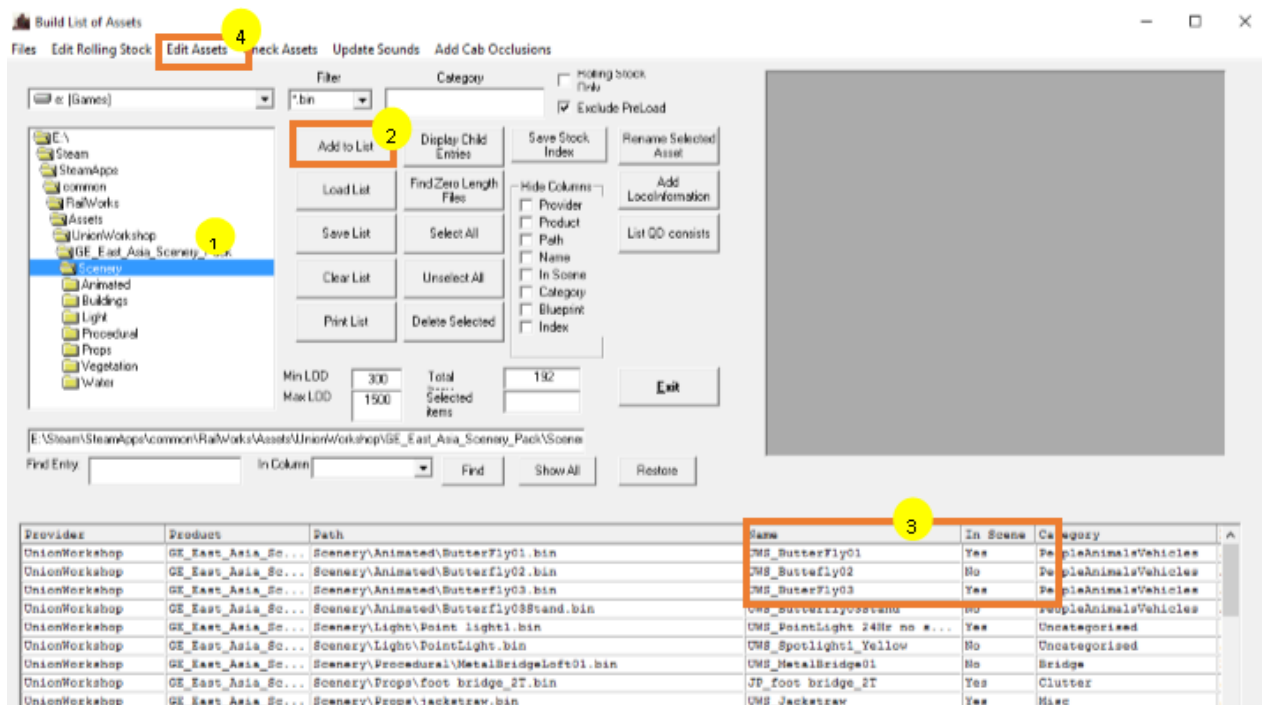


Figure 49. Use the asset editor in RWTools to make an item available in scenarios.

The column In Scenario shows if the object will be shown in the scenario editor. In this case you see butterfly02 (nr 3) is not available, clearly due to an error of its creator. Now select the objects you want to make available. Choose the **Edit Assets** menu item (nr 4) at the top of the screen. In this menu you find a sub item **Use in Scenarios**. Click on this item and you are done.

8 Changing the weather

8.1 Selecting weather

Trainsimulator comes with a variety of standard weather patterns you can use. You find them in the scenario editor in the scenario properties. In most cases you can use the standard weather patterns provided with the route. As with all assets, you can select weather from other providers in addition to the standard weather. The procedure is the same as described in 7.1. To demonstrate this, I downloaded this thick British fog file from UKTS (http://members.uktrainsim.com/filelib-info.php?form_fileid=31681). You need to select newbouy as provider (he also created some wonderful rolling stock, post vans with rolling parcel trolleys, salt wagons etc.).



Figure 50. For the example, Newbouy weather patterns have been enabled. Then the weather pattern can be set as a scenario property.

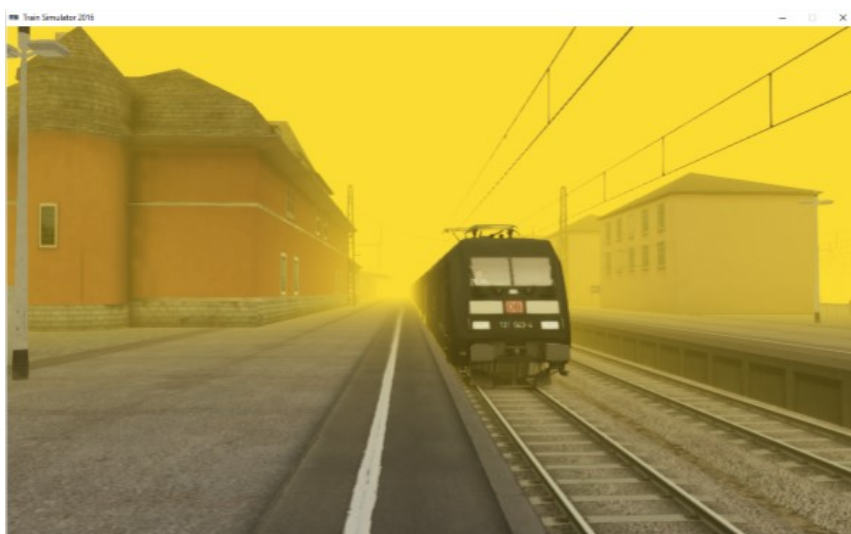


Figure 51. A very dirty foggy atmosphere is the result.

8.2 Create your own weather

In the example in the previous section a heavily customized weather pattern is created. Matt PoddLesdens created an excellent tutorial on this topic.

http://trainsimlive.blogspot.nl/2015/02/ts2015-scenario-scripting-in-lua-part_15.html

I used this tutorial and the very old Railsimulator wiki to provide a more detailed guide on extended weather patterns, that were introduced for TS2013. These allow thunder and sound effects as well.

In the developer docs you can find additional descriptions of the functions. (in the Trainsimulator installation folder look for the “dev” folder and then for “docs”).

8.2.1 Structure of the weather pattern

In order to create weather patterns, you need to understand how the patterns are built before you can start. I will use the new extended weather pattern blueprint, which is much better structured than the old blueprint. The blueprint is structured as described in the table:

| Part | Description |
|-----------------------------------|---|
| Display name | The name of the pattern, as shown in the scenario properties form. |
| CloudDescription | Specification of clouds to be used. |
| PrecipitationDescription | Specification of rain, snow etcetera. |
| Weather | Glue some stuff together in a weather description |
| TriggeredWeatherEventChain | This makes it possible to control the weather sequence from LUA scripting |
| AudioControl | Add audio effects |

As a first step, you need to define how weather for your scenario will develop. I use this form to specify the weather:

| Event | Duration | Weather type | Clouds | Precipitation | Wind | Visibility | Lightning |
|--------------|----------|--------------|------------|---------------|-----------|-------------|-----------|
| Start | 5 | Sunny | None | None | A bit | Clear | No |
| Windy | 10 | Cloudy | White, few | None | Increased | Clear | No |
| Rain | 30 | Rainy | Grey | Rain (little) | Moderate | Bit reduced | No |

It depends bit on your needs what you want to specify, but this shows that you need to prepare 3 cloud types, one precipitation type and change wind and visibility. In gluing it together, you need three types of weather.

8.2.2 Getting started

Before you can start, you need to set up a source folder structure under the railworks folder, e.g. like shown in Figure 52. It most contain a subfolder for the provider (you) and a product folder (In this case the Weather folder). If you create other assets as well, I recommend to create a meaningful product folder and put the weather folder below that.

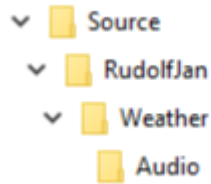


Figure 52. Source folder structure.

Now you have done so, open the program **Blueprinteditor2.exe** in the railworks folder. This will look like shown in Figure 53.

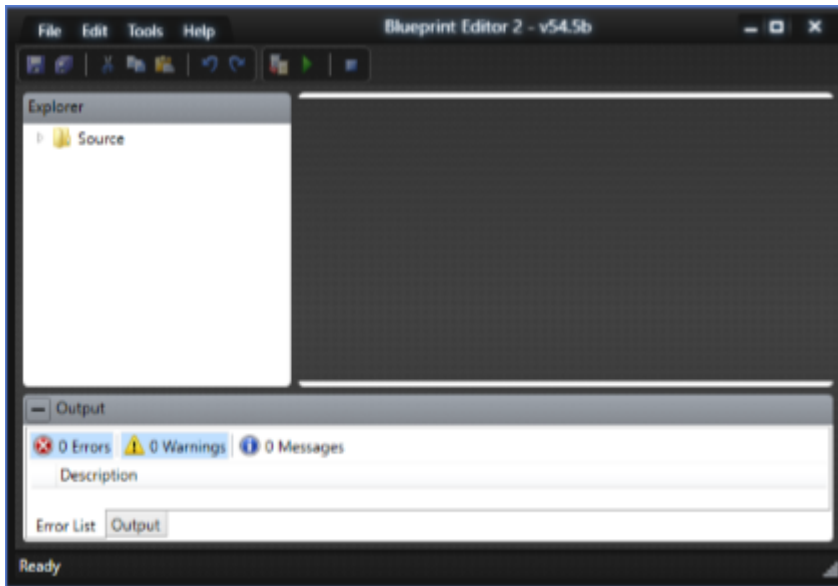


Figure 53. Opened blueprint editor, where source is set up properly.

Now, you navigate to your weather directory, **right click** on the directory and choose a new blueprint of the type “Weather pattern extension blueprint”. Click on OK to select (Figure 54).

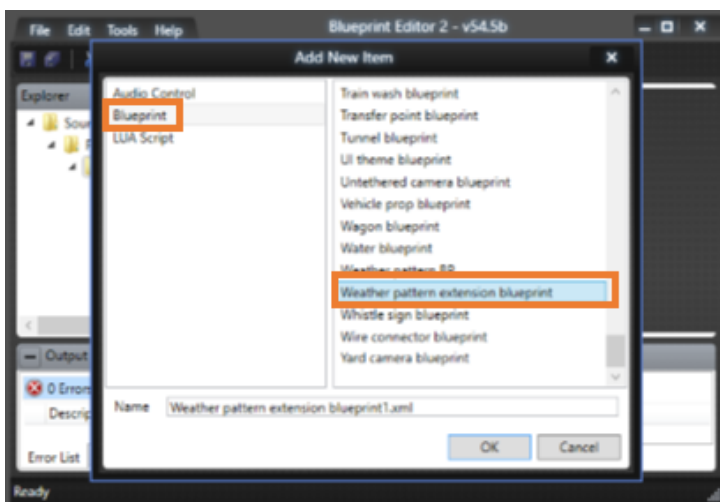


Figure 54. Select a new weather pattern extension blueprint.

I recommend to edit the file name right now, to create a meaningful XML file.

Now we have finished this we can start developing the content of the weather pattern. We did get an empty weather pattern blueprint file now.

8.2.3 Set the display name

The first step is to set the display name. This name is not localized (which is inconsistent with other places where you can use a display name).

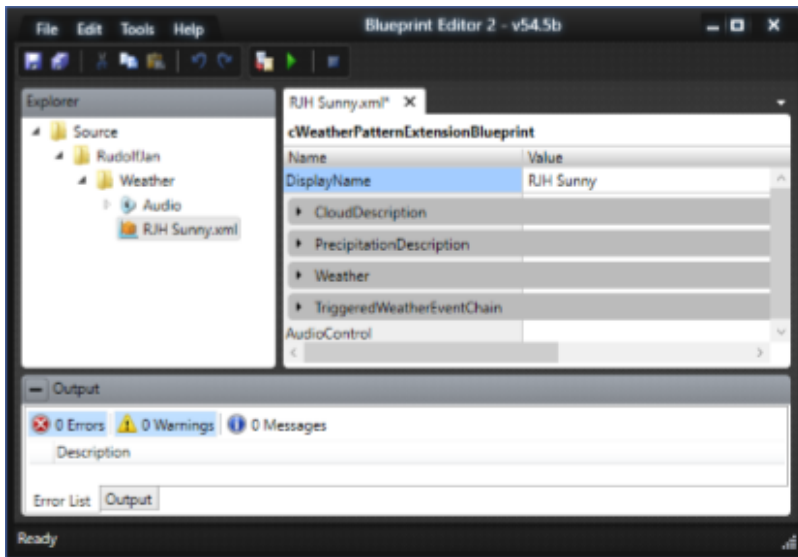


Figure 55. Add a display name to the weather pattern

This step is straightforward. The result is shown in Figure 55.

8.2.4 Set the clouds

In this step, the clouds will be set. As we designed to set up three different sets of clouds, we need to repeat this step three times. If you open **CloudDescription**, you see the “<AddElement>” to add a CloudDescription.

Now it is a bit tricky, the CloudDescription element contains a number of other elements. So, you need to expand the CloudDescription element again, as shown in Figure 56.

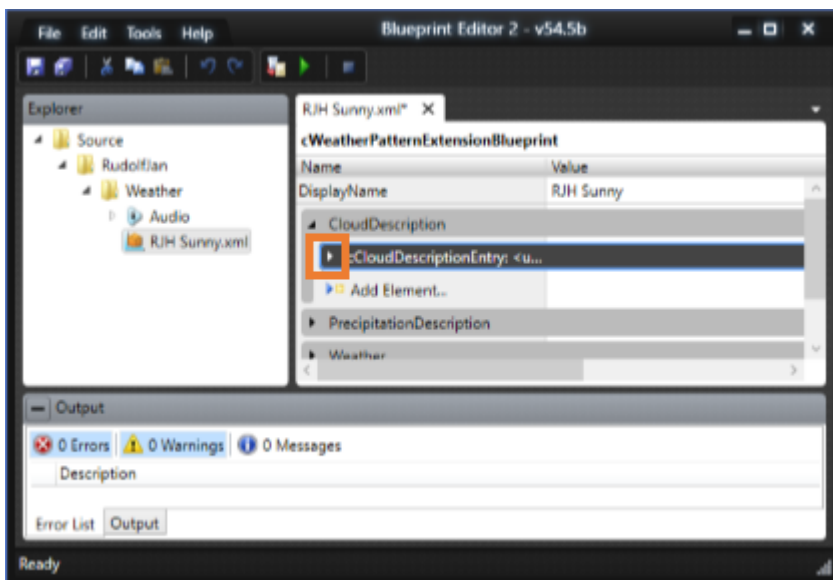


Figure 56. Expand de CloudDescription element.

Once you have done this, you see two elements, **Name** and **CloudDescription**. We fill the value for **Name** and the expand **CloudDescription**. You may find it confusing to see CloudDescription again, but this CloudDescription only shares the name with CloudDescription as we did see before. This CloudDescription

contains the details of what the clouds look like. You can set four parameters. I guess the normal values will be 0 or higher, but they may exceed value 1.0, Matt indicates that ColourDarkeningIntensity may have values of 1.5 or 2.0 as well.

| Parameter | Interpretation |
|--------------------------|------------------------------------|
| WispyLayerAlpha | Thin high wispy clouds |
| ThickCloudLayerAlpha | Thick clouds |
| StormLayerAlpha | Clouds fitting with stormy weather |
| ColourDarkeningIntensity | Darkness of the clouds |

You will need to experiment a bit with the values to find out what works for you. To give you some guidance:

| Parameter | Almost clear air | Bit rainy | Storm |
|--------------------------|------------------|-----------|-------|
| WispyLayerAlpha | 0.5 | 0 | 0 |
| ThickCloudLayerAlpha | 0 | 1 | 0 |
| StormLayerAlpha | 0 | 0 | 1 |
| ColourDarkeningIntensity | 0 | 0.5 | 2 |

Note: you need to treat the values as decimal values, using the dot as separator for the decimals, so do not use the comma, as is normal on the European continent.

The result for the example looks like Figure 57, with all details close.

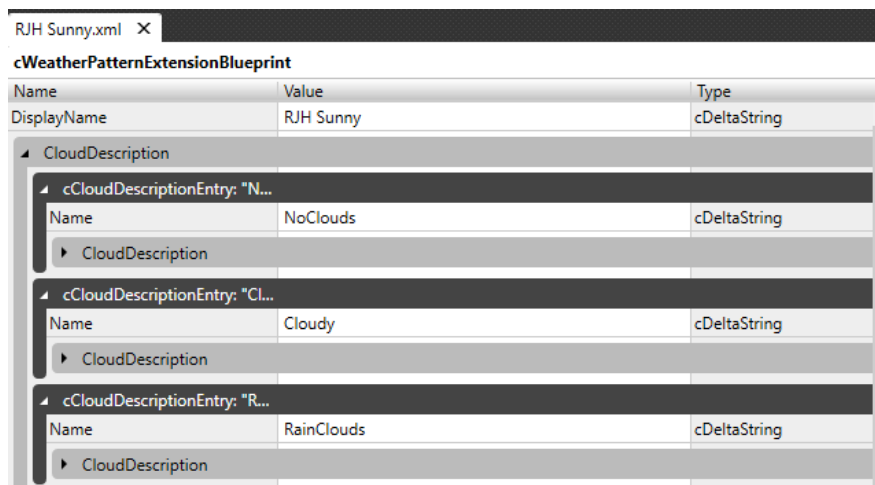


Figure 57. Clouds defined in weather pattern.

8.2.5 Adding precipitation

To make it rain or snow, you need images that represent what is falling from the air.

Precipitation will let you set up a number of different things including the textures for the rain drops, snow flakes, or whatever else you want to fall from the sky. You will need to create a small texture for the rain drop using a paint application (e.g. Paint.NET), 64x64 pixels in size. Use a tool like [RW Ace Tool on UKTS](#) to convert it to the .ACE format. It must be placed in the Weather folder.

Note: in my Google drive I placed a few ready to use .ace files for this purpose.

You need to set the name and then expand, as with the clouds, but the number of options is large. These are described in a very old wiki:

<http://railworkswiki.com/tiki-index.php?page=Weather%20Pattern%20Blueprints&structure=Reference%20Manual>

I will repeat the descriptions here, but in the order they appear in the blueprint:

Bottom Left View Space Offset / Top Right View Space Offset

For these values you are picking an area where the texture will be resized into based on the location of the bottom left and top right values. If the bottom left is $X=0$ $Y=0$ and the top right is $X=1$ $Y=1$ This will show the whole image with equal size sides and there will no resizing. If you then tweak these numbers so that the top right is half the height ($X=1$, $Y=0.5$) then you will have a displayed image squished so it is normal width but half height.

Making these values tiny will make the texture display in a little raindrop size texture. So Bottom left is still $X=0$ $Y=0$ but the top right can be set to $X=0.02$ $Y=0.04$ so it is very small, but it is twice as tall as it is long.

One way to look at these values is to think of them as referencing the raindrop texture that is like a foam ball and it is squishing it down in size based on the numbers. The ball itself isn't changing, just its height/width/size.

Bottom Left UV / Top Right UV

Whereas the values of the Viewspace offset could be thought of as squishing a foam ball, these values can be thought of how you slice up that ball.

Using the same XY 01 grid above, imagine the texture is laid out on that grid and the bottom left is again $X=0$ $Y=0$ and the top right is $X=1$ $Y=1$. This is the whole texture. When you change the values from this you start to slice into what part of the texture is displayed; so if the top right of the image is specified as only halfway up the image ($x=1$, $y=0.5$) then only the bottom half of the texture has been specified to be drawn, so will look like the bottom half of a circle.

I am finding it is better to leave these values so that the UV show the entire texture (bottom left is again $X=0$ $Y=0$ and the top right is $X=1$ $Y=1$) and use the space offsets to change the size. The best size for the offsets I have found so far for light rain is bottom left $X=0$ $Y=0$ top right $X=0.02$ $Y=0.04$. Then set alpha top to 255 and bottom to 0. and precipitation speed to 40. If you want lashing rain, make the offset sizes a little bigger (like loosening the grip on the sponge ball I have forced into a tiny size for the light rain) and maybe up the streak modifier to 2 or 3.

Alpha Top

Adjusting this value sets the alpha, or opaqueness at the top of the texture.

Alpha Bottom

Adjusting this value sets the alpha, or opaqueness at the bottom of the texture. If the values are the same as the top then the texture will have the same opaqueness throughout.

Particle Streak Modifier

This value determines the factor by which the texture used in the precipitation type becomes elongated. By

default this value is 1 which means it is not elongated. Increasing this value will turn a roughly spherical snowdrop into a streak. A raindrop which has been given a value of 10 turns the rain into a series of streaks.

Texture ID

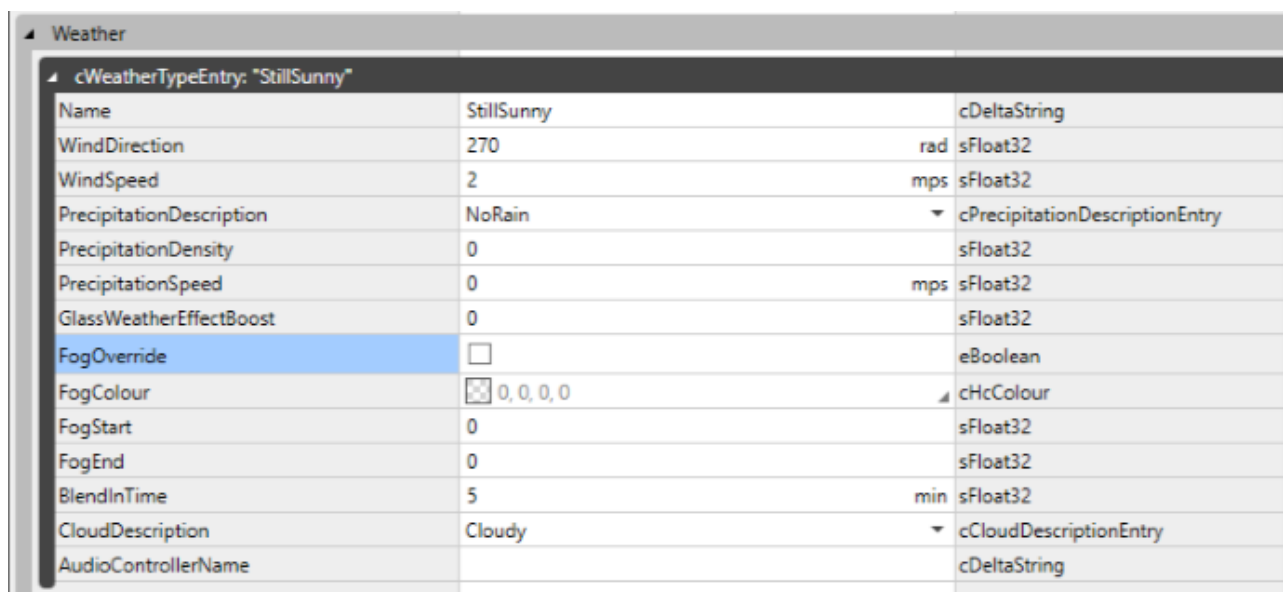
This field points to the location of the texture used for the precipitation type.

For beginners, make particle streak bigger (e.g. 3) and alpha bottom bigger (e.g. 40) to create heavier rain. Make set alpha top to 0 .

You normally must set Top right UV values both 1, otherwise the rain will not be visible.

8.2.6 Add a weather section

We now have the components, clouds and precipitation ready. The next step is to pack the components together, while adding more detail like wind direction, wind speed and fog.



| Weather | | |
|---------------------------------|--------------------------|--------------------------------|
| cWeatherTypeEntry: "StillSunny" | | |
| Name | StillSunny | cDeltaString |
| WindDirection | 270 | rad sFloat32 |
| WindSpeed | 2 | mps sFloat32 |
| PrecipitationDescription | NoRain | cPrecipitationDescriptionEntry |
| PrecipitationDensity | 0 | sFloat32 |
| PrecipitationSpeed | 0 | mps sFloat32 |
| GlassWeatherEffectBoost | 0 | sFloat32 |
| FogOverride | <input type="checkbox"/> | eBoolean |
| FogColour | 0, 0, 0, 0 | cHcColour |
| FogStart | 0 | sFloat32 |
| FogEnd | 0 | sFloat32 |
| BlendInTime | 5 | min sFloat32 |
| CloudDescription | Cloudy | cCloudDescriptionEntry |
| AudioControllerName | | cDeltaString |

Figure 58. Parameters used for the weather section in the blueprint.

Wind Direction

This is a value between 0 and 359 (degrees) which corresponds to the degrees on the compass. Setting the wind direction has a twofold effect:

1. It affects the direction in which the clouds move and secondly
2. It affects the direction in which precipitation falls.

For example, if the wind direction value is set to 90 then the clouds move east and the rain falls diagonally to the east.

For your convenience, you may use this table to find out which value to use:

| Wind direction | Degrees |
|----------------|---------|
| North | 0 |
| North-East | 45 |
| East | 90 |
| South-East | 135 |
| South | 180 |

| | |
|-------------------|-----|
| South-West | 215 |
| West | 270 |
| North-West | 315 |

Note: in the new blueprint editor wind direction units is specified as rad, which is not consistent with the values. Rad would range from 0 to 2π . This probably is a mistake due to lack of math knowledge.

Wind Speed

Setting the speed will affect the rate at which clouds travel across the sky. The higher the value the faster the clouds will move and the shallower the angle at which precipitation falls. Note that the fall angle of precipitation is also controlled by the speed, described later. Wind speed is set in meters/second. In the table below you can derive sensible values.

| Wind speed (Beaufort scale) | Description | Meters per second |
|--------------------------------|-----------------|----------------------|
| 0 | Calm | 0.0 - 0.2 |
| 1 | Light air | 0.3 - 1.5 |
| 2 | Light breeze | 1.6 - 3.3 |
| 3 | Gentle breeze | 3.4 - 5.4 |
| 4 | Moderate breeze | 5.5 - 7.9 |
| 5 | Fresh breeze | 8.0 - 10.7 |
| 6 | Strong breeze | 10.8 - 13.8 |
| 7 | High wind | 13.9 - 17.1 |
| 8 | Gale | 17.2 - 20.7 |
| 9 | Strong Gale | 20.8 - 24.4 |
| 10 | Storm | 24.5 - 28.4 |
| 11 | Violent storm | 28.5 - 32.6 |
| 12 | Hurricane | > 32.6 |

Precipitation Description

This allows you to choose one of the precipitation types you defined before. Just select one from the drop down.

Precipitation Density

This is a value between 0 and 1 where zero is no precipitation fall and 1 is the heaviest. Setting a value greater than 0 but below 1 will provide a varying heaviness of precipitation.

Precipitation Speed

This value works in conjunction with the wind speed to determine the speed and angle of precipitation fall. This value could be thought of as the vertical momentum of the precipitation whereas the speed is the horizontal momentum. For snow choose a low value, for rain use high values. Matt Peddlesden uses 9.8m/s. This clearly has been inspired by the acceleration of the gravitation, which is 9.8 m/s^2 in vacuum, but has nothing to do with the value we are looking for. I checked, a range of 1-10 m/s should be realistic values for rain. For snow use a range of 0.1 to 0.3 m/s.

Note: I did not test these values, let me know if they behave realistically.

GlassWeathereffectBoost

Determines how much water will be on the windows. Use 0 for no drops and 1 for the maximum amount (you wipers are not able to handle that much).

The next four parameters define fog. Leave fog override off if you do not want to set fog.

Fog Override

This is a true/false option which simply determines whether the default fogging is used, as set up in the time of day Blueprint, or whether the settings used are those set in this blueprint, overwriting these defaults.

Fog Colour

The fog color can either be set by sliding the separate red, green and blue slider bars or by clicking the color bar itself where a new window will appear.

You need to set the color using combined red, blue and green. If you keep the values of all three about equal, the result will be grey shared, where 0 for all three is white and 1 for all three results in black. Unfortunately DTG did not choose to use hexadecimal values, but fractional values. So the best way is try and see, until you get the desired result. The fourth channel (labeled A) is the alpha channel, which makes the fog more or less transparent. Not sure if it works properly in practice.

Tip: You may want to tailor weather patterns to a scenario. A weather pattern with light color fog may look great in daytime, but will make a scene look illuminated at night. For darker times of day a darker fog color is recommended.

Fog Start

This value is in meters and determines how close to the camera the fogging effect begins.

Fog End

This value is also in meters and determines the point at which all objects become fully fogged out. Setting this to a relatively small value, such as 150, will mean that any objects further than 150 meters from the camera will be hidden by the fog. 50 meters is fairly thick fog, 1000 meters is just a bit hazy. Clear visibility may use ranges in the order of 20000 meters. Not sure if it is wise to do this.

BlendInTime

The amount of time the weather pattern uses to evolve from the previous pattern. Blending is a bit tricky, because you also can make weather chains that also do some blending. For the first weather in the blueprint, set this to 0, for others use at least 1 minute, but 5-10 minutes is more realistic.

CloudDescription

This refers to the clouds you are using. Choose one of the cloud types you defined before from the drop down list.

AudiocoltrollerName

This refers to the audio blueprint used by the weather. I do not yet know how this works exactly.

9 The cinematic camera

Coming in 2016!

10 Tools

10.1 Utilities

You can use the Trainsimulator Utilities tool for several purposes:

1. Package your scenarios into a distributable .rwp file
2. Unpack assets that are stored in .rwp files.
3. Localise your scenario for other languages.

You can find the Utilities.exe file in the Railworks folder.

10.2 LuaCreator

LuaCreator is a new powerful tool. It still is in early development stage. As the name says, it is targeted towards LUA scenario script creation, but serves other purposes as well for scenario authors:

1. List all routes and scenarios, including the ones packed in .ap files.
2. Show the most important scenario properties.
3. Allows easy access to route- and scenario folders.
4. Capabilities to create LUA scripts from scratch or use code snippets (several code snippets are provided).
5. Compile LUA scripts.
6. Get an overview of all used rolling stock and missing rolling stock.
7. Launch trainsimulator with various debugging settings.

More features will come. You can download LuaCreator here:

<https://drive.google.com/open?id=0B5xmS3GoYKV1UjBJeHhIOFIOSzQ>

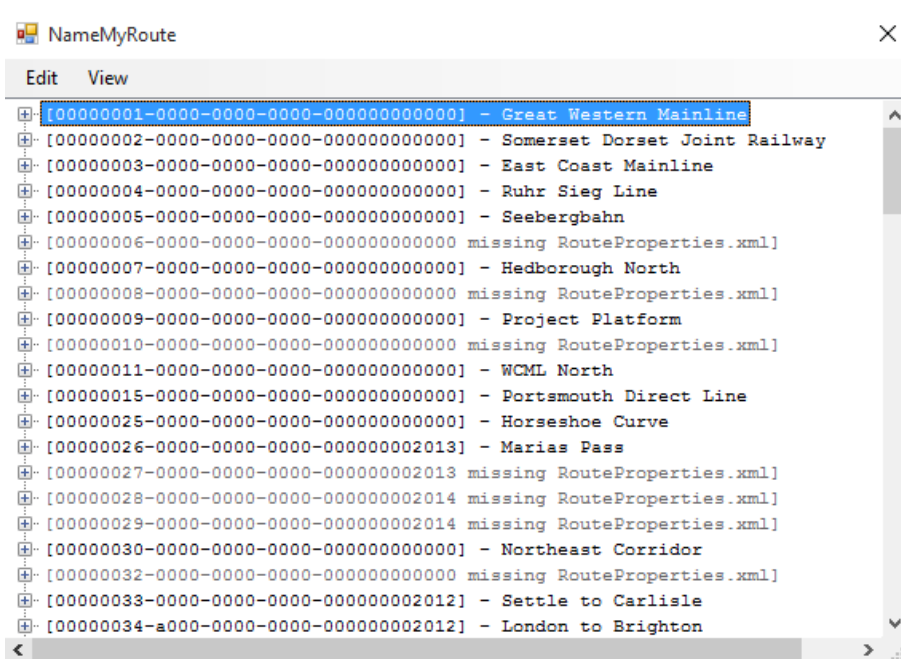


Figure 59. Name My Route utility.

10.3 NameMyRoute

NameMyRoute is a small tool provided in the Railworks folder. It will translate route GUIDs into route names, but routes packed in .ap files are not supported. It can open the folder of the route or scenario as well, which makes it very useful.

If you click at the + signs, it will show the scenarios as well.

Note: LuaCreator will also show packed routes and allows you to open the folder from the tool

10.4 Serz

Trainsimulator uses a lot of XML files. To save disk space, many of these files are compressed, using a tool called **Serz**. You will find Serz in the Railworks folder.

Serz works very simple. Drag a .bin file over Serz or open it with Serz and it will create the (uncompressed) xml file. Xml is readable, though you probably will not like it.

Vice versa works the same.

Of course, the use of .ap files creates a lot of trouble here as well.

10.5 RWTools

RWTools is the most important tool for any user of Trainsimulator. It will help you to check scenarios for missing rolling stock and allow you to replace it. It also has a number of tools for editing Trainsimulator files, including unpacking all .ap files.

RWTools is donation ware, but the author of RW Tools fully deserves your donation!

You can obtain RWTools here: <http://www.rstools.info/downloads.html>

10.6 RW TextEdit

RW TextEdit is a tool I recently discovered. It combines the Serz function with a text editor. This allows you to open a .bin file, change it and save the result into the .bin file. You can get it here for free:

<https://www.ivimey.org/content/rwtextedit>

10.7 XML Notepad

Originally developed by Microsoft people, the development is now continued externally. XML Notepad is a simple editor for XML files. The big advantage is that it helps you a lot to have a fast overview of the contents of the file. You can download the latest version here:

<https://xmlnotepad.codeplex.com/>

11 Create a scenario for other people

The scenarios you learned to create in Part I are meant for your own use. But, if you spend a lot of time in creating a fancy scenario, it is very nice to share it with other Trainsimulator community members. In this chapter , I will discuss some aspects you need to consider when creating scenarios for other players.

11.1 Publication medium

The first thing to think of is, where you want to publish your scenario. In chapter 12 I will provide information in more detail. Some publication sites impose restrictions on what you are allowed to include in your scenario or what requirements exist for e.g. documentation. Especially if you want to publish in Steam Workshop, you only can use assets that are commercially available through Steam. It saves a lot of trouble to think about this before you start working.

11.2 Use of rolling stock

Suppose you collected tons of assets and fine liveries and you create a very accurate realistic scenario. That's fine, but with each asset pack that is needed, the number of people that can play your scenario decreases.

What you can do is to have a look in Steam Workshop . Pick the route for which you want to create a scenario and make a list of scenarios previously published (Excel is a good tool for this purpose).

I did this exercise for Weardale and Teesdale Network. At the moment 116 scenarios are available. In ... you can see the European Assets are widely used. This should not be a problem. You also can see a number of authors use steam routes, e.g. Western Lines of Scotland and Falmouth Branch. Surprisingly, the Rivieraline is not widely used, while you may expect a lot of people have this line. Maybe this is because the WTN route was issued a few months before Trainsimulator 2016 was issued.

| | subscribers | Class 105 | Eur. Assets | WLOS | Falmouth |
|--|-------------|-----------|-------------|------|----------|
| Have with yow to Walsingame | 98 | | 1 | | |
| Single line coal | 129 | | | | |
| Morning rush | 79 | 1 | | | |
| Durham-Darlington stopper | 120 | | 1 | | |
| Hill climb | 60 | | | | |
| Middleton shunt v2 {part 1 of 2}. | 37 | | | | 1 |
| Rescue Mission | 45 | 1 | | | |
| Wear Valley 3a: An Early Sunday Morning | 39 | | | 1 | |
| Jack Frosterley (Class 25) | 193 | | 1 | | |
| St. Blazey Shunt | 38 | | | 1 | 1 |
| Darlington-Durham stopper (Class 101) | 178 | | 1 | | |
| Running Shunting with the Bishop [Hard] | 126 | | | 1 | |
| Barnard Castle to Darlington (Class 101) | 472 | | 1 | | |

To minimize the number of needed DLC, it may be a good strategy to use one or two additional and popular routes to create more variety in rolling stock. In the table you can see the number of subscribers drops as low as 30-40 if you require more than 3-4 DLCs for your scenario.

Of course, your customers can exchange rolling stock using RW Tools, but this is a somewhat cumbersome process.

11.3 Timing and testing

During development of a scenario you must do a lot of testing. This makes you very experienced in running the scenario. You know in advance exactly how the scenario behaves and how it works. Other people lack this experience, so it is likely they will need more time to complete the scenario than you do. Allow some time at the start, and relax time tables a bit. During testing, drive as if it is the first time you drive the scenario. If you see a yellow signal, brake as if the next signal is red, even if you know it will be green long before you arrive. Another player will NOT know this.

Similar remarks hold for manually setting junctions. In a large yard it may take some time to find out which junctions must be set. Allow for this time, and eventually include a stop for this purpose.

You also must take care to check for possible “signal passed at danger” problems. These should never occur in a published scenario. If the player needs to ask permission, you should in general inform her about this, especially if the scenario is timetabled. Unfortunately arriving late at a timetabled stop in a standard scenario is interpreted that you did not complete the scenario. A very annoying decision, but something to consider when you design the timing for a scenario.

Another signalling topic is that sometimes the distance between signals is too short. In this case you should inform the player to reduce speed ahead. For normal gameplay the player should be able to stop in time without excessive braking from track speed.

You should make clear which driving style you expect. Should the player seek the extreme limits or do you want to promote decent driving? You also may recommend a driving speed lower than track speed. For most scenarios nothing is said about this aspect, leaving it to the player to discover how it works.

11.4 Make sure AI trains are seen

Another aspect of timing relates to AI trains. You may intend the player to have to wait at certain points or to see AI trains. You never can be sure the player will arrive at the right time at the right spot, so take care to include some room for deviation from your schedule. A sensible approach may be to cut a long scenario in more parts or to introduce timetabled stops. This works best for passenger trains, for freight trains you need to include an instruction to tell the player when to depart. Players that are very fast, will have to wait for some time at this point, so you can sync with AI traffic again. Late players still can catch up then.

11.5 Documentation

Normally I tend not to document scenarios, except the in game instructions and a list of required assets. Most websites require a description, rolling stock list and an image. I always provide the rolling stock list directly at the download site, so you can decide if you can play the scenario without problems.

11.6 Localization

Localization means you create a scenario with instructions in a specific language. It is surprising how many people don't understand even a small bit of English. In theory, Trainsimulator supports localization. In practice, it doesn't work properly. A good practice is to select English as the Trainsimulator language, develop your scenario in English and then use the localization tools to create a translation.

If you don't do this, but instead create a scenario using e.g. German language with German set as localized language, your texts will be invisible for people that have set English as language. So people will entirely miss the instructions you provide. So always make sure to create texts for English as game language. You can use German texts in this case as well if this is convenient for you.

11.7 Translating your scenario to other languages

Coming in 2016!

12 Publishing scenarios

12.1 Where to publish

If you want to publish your scenario, you first will need to consider where to publish it. Publishing at steam workshop is easy, but imposes a number of limitations.

There are a number of good community websites where you can publish a scenario. Of course you are free to publish at multiple sites, but you in case of updates you need to change the scenario at all sites.

You should consider an number of aspects to choose a site:

1. The country/region for which you developed the scenario.
2. How much control do you have on what will look like?
3. Can you publish updates?
4. Can you retreat your contribution later?
5. Do you keep the copyright?
6. Has the site open access for download?

Let me give you some fairly good sites to consider:

| Site | Country | Edit | Update | Retreat | Copyright | Access |
|----------------------------|-----------------|------|--------|---------|-----------|-------------|
| UKTS | International | Yes | Yes | Yes | Yes | Slow or pay |
| RWA | USA | No | No | mail | Yes | Free |
| Treinpunt | Netherlands | Yes | Yes | ? | Yes | Registered |
| Railsim-de | Germany/Austria | Yes | Yes | ? | Yes | Free |
| RW Austria | Austria/Germany | No? | No | mail | Yes | Free |

12.2 Create an rwp file

Except if you intend to publish in steam workshop, you need to compress the scenario files into an installation package. For Trainsimulator the **.rwp** file format is used for this purpose.

In principle, creating a .rwp file is easy. You do this using the **Utilities** program, which is part of Trainsimulator (see 10.1 for more information).

Open the Utilities program and select the tab “Packager” Now, press the button “Refresh” and have a lot of patience (Figure 60). The utilities program will now do create a database of all game files, which takes a lot of time. When it is finished it will show the route names, scenario names and asset folders. This allows you to select one or more scenarios to include in the .rwp file as shown in Figure 61. You must take care here, not to select files that should not be published, e.e. CurrentSave.bin contains the saved game and should never be published. You also never must publish backup files and you may choose not to publish LUA files that are not compiled.

So the steps are (referring to Figure 61):

1. Select the files to publish. Note that you also can select manual files and assets, but always take care to have permission from the author if you want to do so.
2. Set the author name and licence. I am not sure what protected exactly does, so I never use it.
3. Use toe copy button to build the complete selection.

4. Use the Create Package button to create the rwp file. This opens the file dialog that allows you to set the file name.

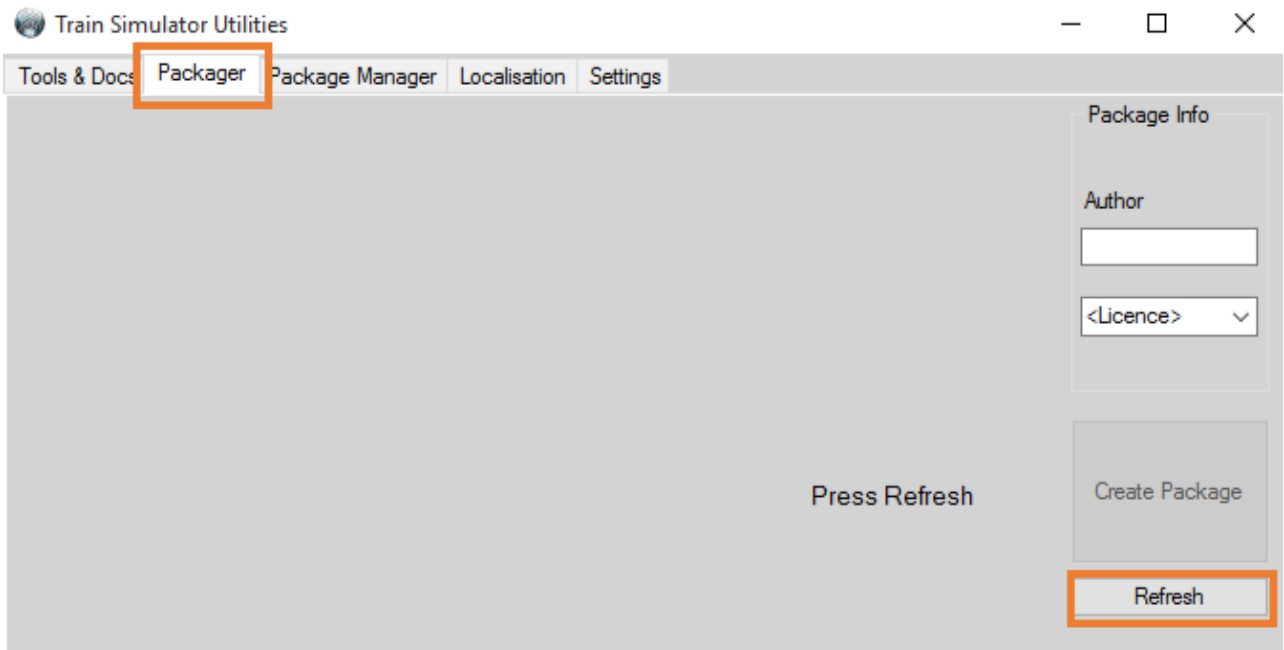


Figure 60. Utilities packager screen

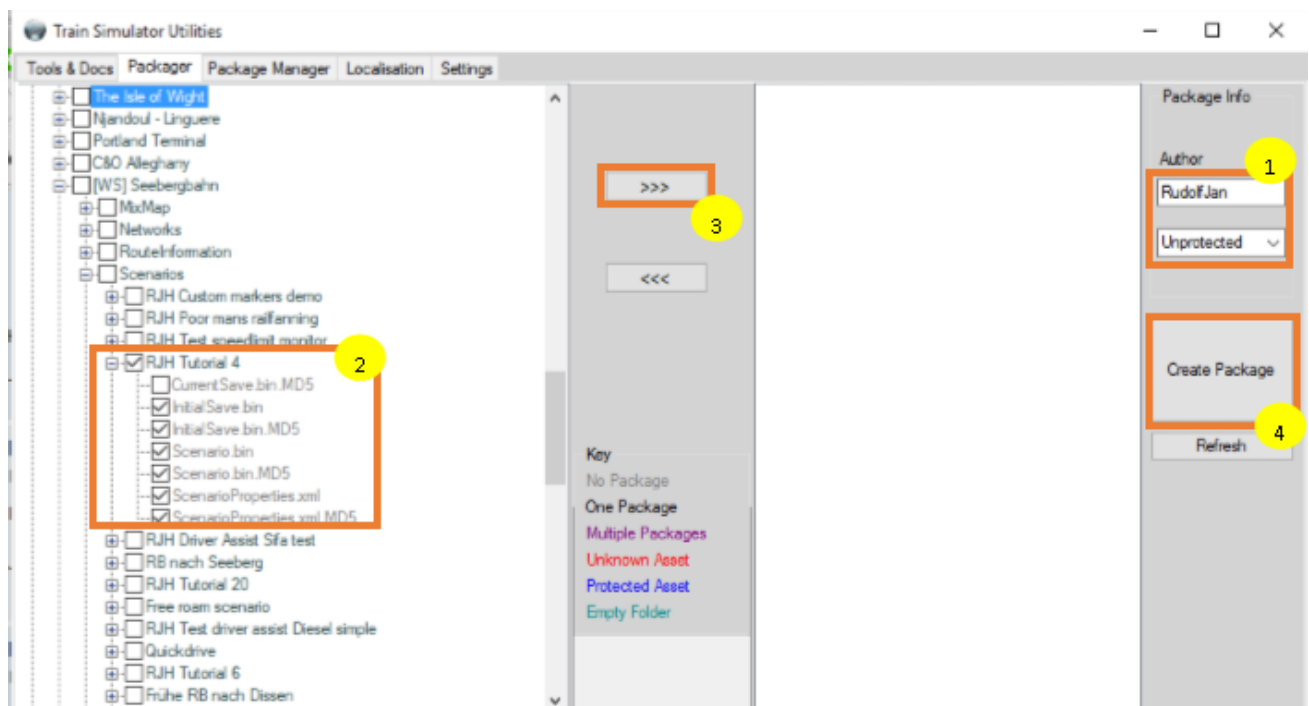


Figure 61. Selecting files in the packager

The .rwp file is simply a .zip file, which can be opened with for instance 7Zip.

Note: you will soon discover that the packager will only show the GUID and not the route name if the route is packed as a .ap file, which is quite common now. In this case you can use RWTools (see 10.5) to unpack the route properties first. If you refresh Utilities afterwards, the route name will be shown.

12.3 Compressing files

Coming in 2016!

12.4 Publish in steam workshop

If you want to publish on Steam Workshop, always make sure to have at least one screenshot available.

Screenshots for use with workshop uploads have to be taken by the Steam Screenshot system. To take a screenshot using the Steam system, press **F12** when in game. When you restart steam, select your user name (1) in the top menu, then the sub menu “Content” and the “Screenshots” (2).

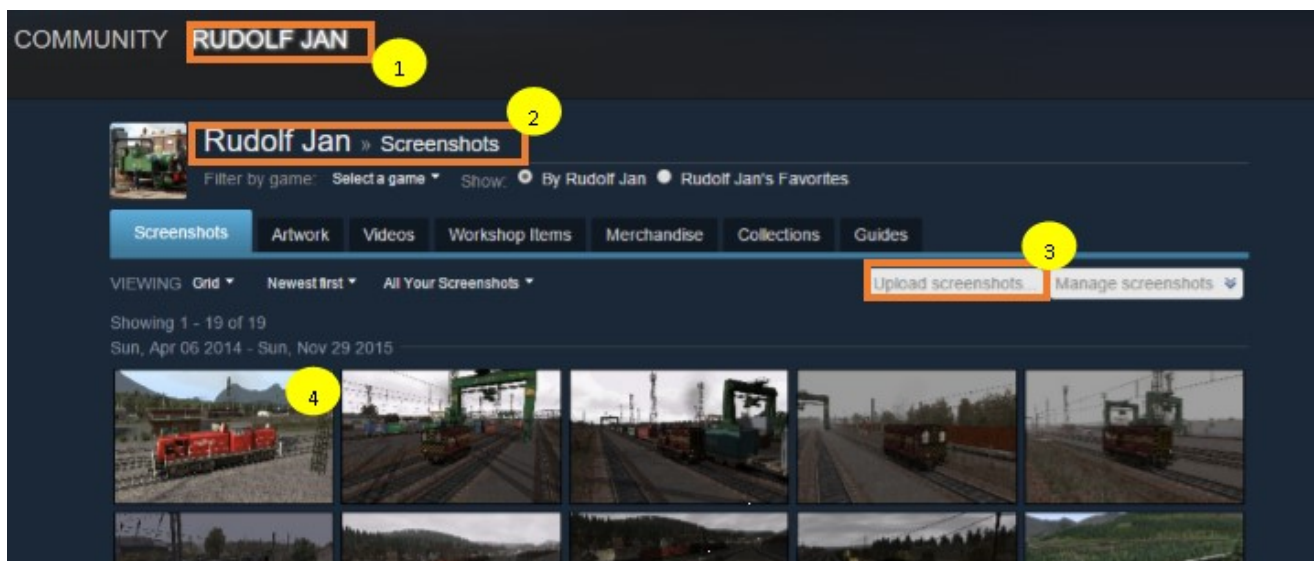


Figure 62. Uploading screenshots to steam.

Now you can upload the screenshots by pressing the “Upload screenshots” button (3). See Figure 62. Then you can start preparing your workshop scenario from Trainsimulator. It all looks a bit clumsy and confusing, but this seems to work.

Now you restart Trainsimulator, and select the “Build” button.

Now complete the following steps (numbers refer to Figure 63):

1. Select the publish tab in the Build menu.
2. Select the Scenario button
3. Locate the scenario you want to publish. Note that quite a lot of scenarios are shown here and the route names are not mentioned. So I recommend to use a naming scheme that makes it easy to locate your scenarios (I always use a prefix RJH).
4. Select the main screenshot to display.
5. Complete the description. I suggest you mention at least which DLCs are required.
6. Complete the other information, including your agreement to the terms and conditions.
7. Push the publish button

After step 3 Steam validates if you are allowed to publish this scenario. The main requirement is that you only can use DLCs that have been published through steam. Unfortunately this is not always working properly. Sometimes DLCs are not registered correctly, causing a denial.

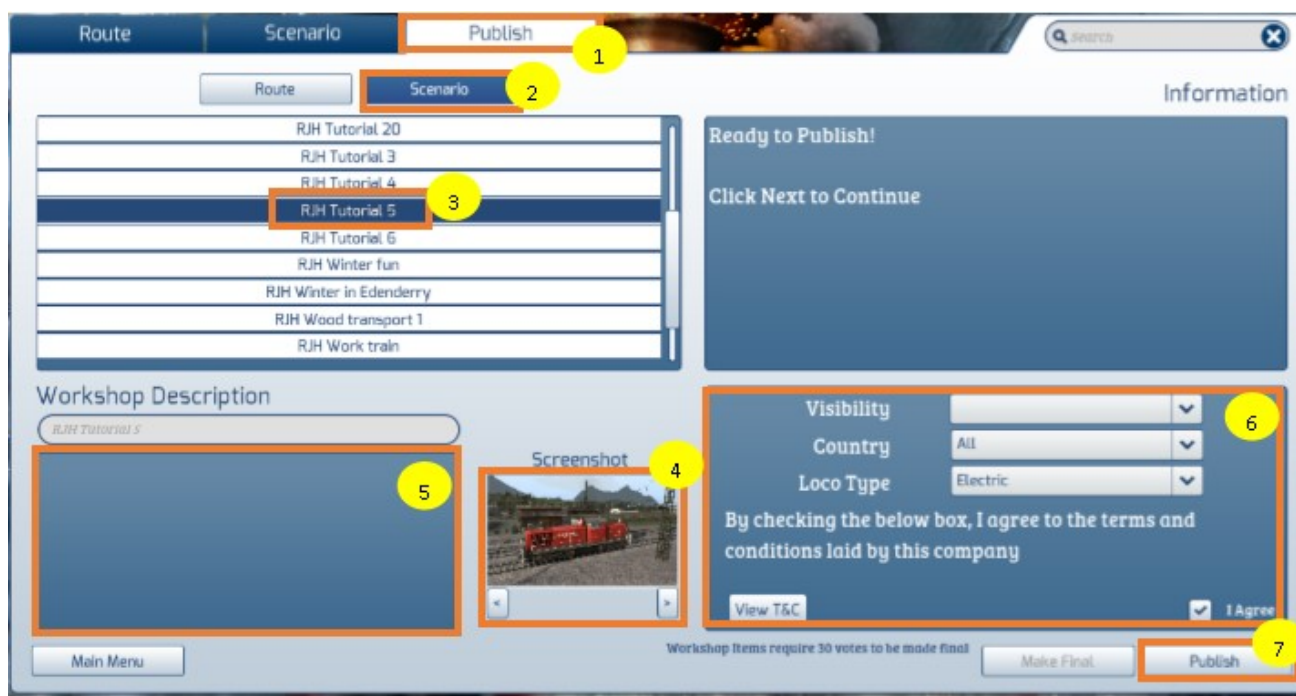


Figure 63. Publishing a scenario at steam workshop

13 Scripting

Coming in 2016!

14 Trouble shooting

14.1 Editing the scenario properties

Coming in 2016!

14.2 Editing scenario.bin file

Coming in 2016!

14.3 Follow AI trains

(With many thanks to [styckx](#) and UKTS).

During scenario play you can easily switch from the player train and through all running AI trains. This may be useful if an AI train is causing trouble, or just for fun.

To enable this function, you need to add the option **-followaitrain** to the Trainsimulator launch options.

Note: a good way to do so is to use LuaCreator, which has a launcher that supports this option.

Now you can select a train and press **CTRL+ the left mouse button**. This moves you to the external camera view for the first AI train.

Using the **PageUp** or **PageDown** button you can cycle through all AI trains and the player train.

You cannot use any controls in this mode, just watch. It also is not possible to change the camera view.

You can select the 2D view. It seems it was intended that you can select any train from a menu there, but this does not work. The list with trains remains empty.

By a **left click** (without CTRL key) you return to normal mode, but you will need to release brakes as the player train will be automatically be set to stop.

14.4 Speed up scenario execution

Coming in 2016!

14.5 Recalculate timing and show driver list

Sometimes it can be useful to recalculate all timings and relations between elements used in a scenario. You can do so in the scenario editor, by pressing the **F5** button.

If you set the launch option -ShowDriverlist and inside the scenario editor and you select the driver icon of one of the trains, you see an additional button, as shown in

. If you click on this button a window appears, showing all movable consists . Pressing apply will cause a recalculation, but you can disable this recalculation, making the game think the consist do not exist. This can be helpful to track errors and conflicts in your scenario. Start disabling all consists, but one, then and another consist and see when the errors appear.

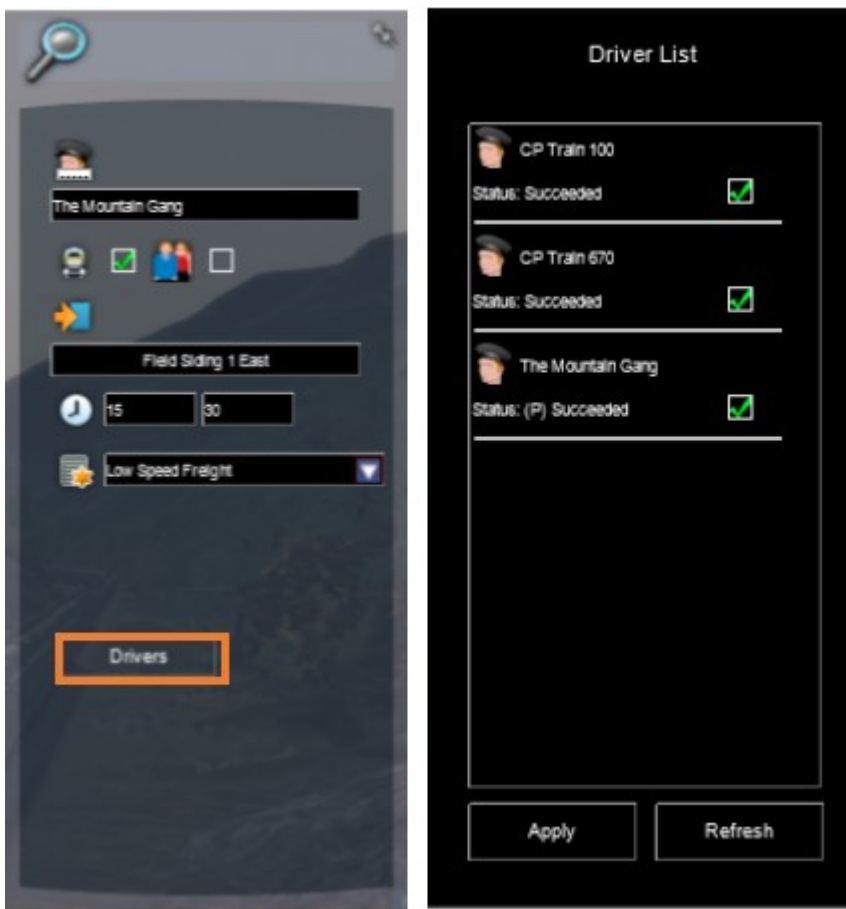


Figure 64. Driver instructions list

14.6 Miscellaneous troubleshooting tips

14.6.1 Avoid service class Special

The service class **Special** has some undocumented properties, that make it behave weird. The general advice is to avoid using it. If you use it and unexpected things happen, stop using it. If you have difficulties with a scenario you cannot solve in any other way, try **Special**.

14.7 Avoid minimum speeds for stop at instructions

In a **Stop At** instruction you can set a minimum speed. Setting this minimum speed to 1 Mph is OK, it make the instruction a Go Via instruction. Setting higher minimum speeds may cause trouble, if the dispatcher thinks it is not possible to reach that speed. It will then try to avoid the path and do very strange things.

14.7.1 Avoid testing a scenario directly from the editor

The orange triangle which you need to use to exit the editor, starts the scenario right away. Unfortunately this causes a lot of problems. The main problem is that signals will be set to a default state that is incorrect

for game play. Therefore you always should terminate the scenario execution after editing. It is OK to use the Pause button and then select restart scenario (Figure 65). This will work properly.

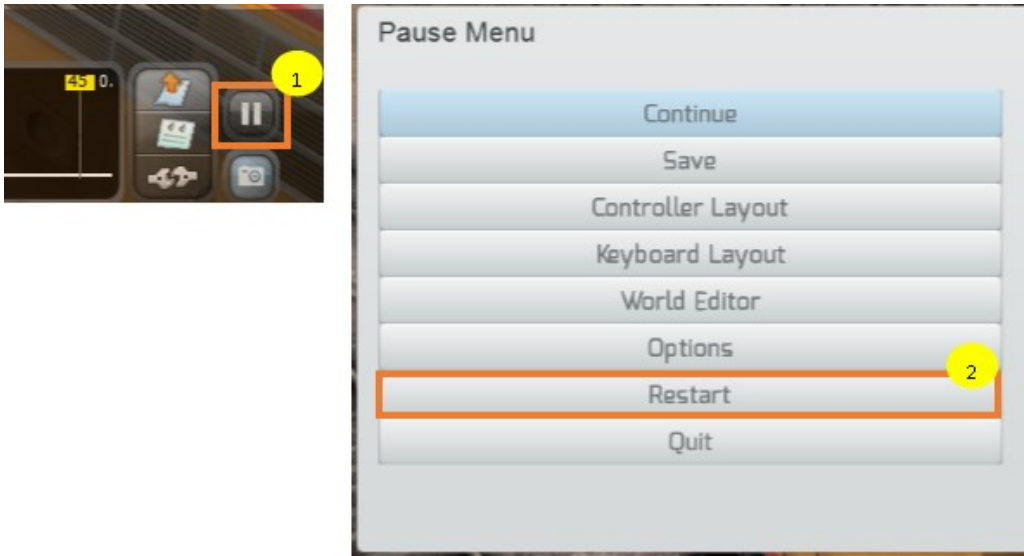


Figure 65. Always restart the scenario after editing

14.7.2 Check your path

In time table view you always can show the intended path of all trains. Sometimes the dispatcher calculates an unexpected path, maybe due to errors you made or due to pathing designed for the route. You always must check the path for each consist. Eventually you may need to add additional **Waypoints** or **Go Via** instructions to correct pathing issues. In some occasions, changing the **service class** may work as well.

14.8 Solving problems with tracks and signals

Pressing the spacebar in the scenario editor, makes it display track properties. First time shows line type, second shows directionality, third shows electrification type. You can continue cycling, it also will show speed limits and some other things I do not understand. The last click will give you additional information about signaling.

The complete list of what it shows:

| | |
|--------------|----------------------------|
| | |
| No Space bar | Normal rendered track. |
| 1 | Line Type (Yard, main etc) |
| 2 | Directionality |
| 3 | Speed Limits |
| 4 | Sounds |
| 5 | Electrification |

6 = Super-elevation

7 = Wobble

8 = Track linked objects

9 = Return to normal rendered track

| Example | Color | Interpretation |
|---|--------------------|---|
|  | Yellow | Main line |
|  | Red | Freight |
|  | Blue | Passenger |
|  | Brown | Yard |
|  | Both directions | This track is meant to be used in both directions |
|  | Up/down direction | Here the tracks are unidirectional. |
|  | No electrification | This track has no electrification. You cannot drive here with electric engines. |
|  | Overhead wires | This track has overhead wires. |
|  | Third rail | This track as a third rail. |
|  | Speed limits | This example shows a location where speed limits change. |

You can use this view to solve a number of issues with tracks in your scenario. In a scenario it is possible to override most track settings

A. Useful links

| Site | URL's |
|--|---|
| Trainsim live (Matt Peddleston) | https://www.youtube.com/playlist?list=PL9DHSvtA6yvSfjkhQesr-eSjTcrpgSSFU |
| RWTools | http://www.rstools.info/downloads.html |
| RWTextEdit | https://www.ivimey.org/content/rwtextedit |
| XML Notepad | https://xmlnotepad.codeplex.com/ |
| LuaCreator | https://drive.google.com/open?id=0B5xmS3GoYKV1UjBJeHhIOFIOSzQ |

B. Availability of this guide

The scenario authors guide is available at these community websites (URLs refer to part III):

| Site name | URL's |
|---|---|
| Treinpunt | http://forum.treinpunt.nl/index.php?action=downloads;sa=view;down=5867 |
| UKTS | http://members.uktrainsim.com/filelib-info.php?form_fileid=33231 |
| Railworks America | http://railworksamerica.com/index.php/download-library/accessories?view=document&id=1845:scenario-author-s-guide-part-iii-lua-scripting&catid=24:accessories |
| Railsim-de | http://rail-sim.de/forum/wsif/index.php/Entry/1534-Scenario-Authors-Guide-Part-III-Scripting/ |
| Google Drive (all parts, beta's and other goodies as well) | https://drive.google.com/open?id=0B5xmS3GoYKV1UjBJeHhIOFIOSzQ |

Index

| | | | |
|--|--------|-------------------------------|----|
| AI spawn point..... | 10, 42 | Destination Platform..... | 35 |
| AI Spawn Point..... | 36, 42 | Displayname | 45 |
| AI traffic | | distance between signals..... | 61 |
| Railfan..... | 54 | documentation | |
| AI train | | scenario..... | 61 |
| consist operations | 19 | DoNotSpawn | 46 |
| AI trains | | down the route | 39 |
| scenarios..... | 61 | DrivableConsist | 45 |
| automatic switches..... | 24 | driving speed..... | 61 |
| Back | 41 | eCareerScenarioClass..... | 31 |
| backup files..... | 64 | edit scenario | |
| bidirectional track..... | 16 | rolling start..... | 25 |
| bug | | edit the marker | 16 |
| localization..... | 61 | empty passenger coaches..... | 28 |
| cab camera | 26 | Era Parameter | 35 |
| career scenario | | EraEnd | 45 |
| rules | 32 | EraStart | 45 |
| success..... | 32 | eStandardScenarioClass..... | 31 |
| career scenarios..... | 26 | eTimetableScenarioClass | 31 |
| Centre option | 39 | exchange rolling stock | 60 |
| change the marker to a stopping point..... | 14 | exit the editor | 71 |
| change the scenario type | 31 | expert controls..... | 26 |
| community websites | 63, 76 | final destination | 47 |
| consist | | -followaitrain | 69 |
| save..... | 19 | force cab camera | 26 |
| consist operations | | Fragment | 44 |
| AI train | 19 | goods yard | 44 |
| consist properties form | 19 | grey cubes..... | 19 |
| Consist spawn point | 10 | GUID..... | 65 |
| Consist Spawn Point | 36 | hidden instructions..... | 30 |
| Consist Type..... | 41 | InitialSave.bin | 24 |
| ConsistType | 45 | instruction order | 29 |
| couple wagons | | instructions | 14 |
| AI train | 21 | Inv_road.bin | 47 |
| CurrentSave.bin | 64 | invisible track | 47 |
| custom track properties | 16 | launcher | 69 |
| decent driving..... | 61 | length of the consist | 43 |
| descriptive name | | line type | 73 |
| saved consist | 20 | Loading point | 10 |
| descriptive text | | localization..... | 61 |
| coupling instruction..... | 28 | LocoName | 45 |
| Destination | 34 | LUA script | |
| destination marker | 11 | Railfan | 49 |
| Destination marker..... | 10 | LuaCreator | 69 |
| Destination Marker | 47 | manual switches | 24 |

| | | | |
|--|--------|------------------------------------|--------|
| save..... | 24 | rwp file format..... | 63 |
| manually setting junctions | 61 | RWTools | 59 |
| marker visibility | 16 | save a consist | 19 |
| maximum consist length | 41 | saved consist | |
| medal | 33 | use..... | 20 |
| minimize the number of needed DLC..... | 60 | visibility | 21 |
| minimum speed | | scenario failure | 32 |
| Stop At instruction..... | 70 | Scenario marker | |
| NameMyRoute | 59 | location | 31 |
| naming scheme | 40 | scenario properties..... | 26 |
| Packager | 63 | quick drive scenario | 36 |
| planning | | scenario success..... | 32 |
| quick drive scenario..... | 34 | scenario type..... | 31 |
| platform marker | 13 | ScenarioProperties.xml..... | 31 |
| Platform marker | 10 | screen icon..... | 16 |
| Player Consist | 34 | screenshot | 66 |
| player instructions | | select tool | 17 |
| Railfan | 48 | service class Special | 70 |
| player spawn point | | Serz | 59 |
| properties | 36 | –ShowDriverlist..... | 69 |
| Player spawn point..... | 10 | Siding marker | 10 |
| Player Spawn Point..... | 36, 42 | signal icon | 10 |
| Player Spawns | | signal passed at danger..... | 61 |
| additional..... | 39 | signal state | |
| points | | scenario editor | 71 |
| career scenario | 33 | simple controls..... | 26 |
| pool of consists | 35 | Starting Location | 34, 35 |
| Portal | 10 | StartingSave.bin | 25 |
| PreLoad metafile | 35 | static consist | 21 |
| priority | 42 | Static Consist Spawner..... | 42 |
| publication site | 60 | Static Consists | 41, 44 |
| publishing scenarios | 63 | Steam Workshop | 60, 66 |
| Quick Drive Consist | | stopping point..... | 13 |
| setup..... | 44 | Stopping point | 10 |
| quick drive scenario..... | 34 | tape measure | 41 |
| create..... | 36 | temporary speed limit | 16 |
| rail vehicle numbers | 28 | testing | |
| Railfan Scenario | 46 | scenarios | 61 |
| Railsim-De..... | 63 | time tables | |
| Railworks America | 63 | testing | 61 |
| Rainfan Invisible | 47 | timetabled properties..... | 31 |
| Relay Scenarios | 46 | Timetabled scenario | 31 |
| Rolling Start | 24 | track infrastructure objects | 11 |
| rolling stock list..... | 61 | track properties | 17, 71 |
| Route | 34 | track type | 16 |
| route plan | 34 | track unevenness | 16 |
| routing for AI | | Treinpunt | 63 |
| RW Austria..... | 63 | UKTS | 63 |
| RW TextEdit | 59 | unhide a hidden instruction..... | 30 |
| RW Tools..... | 60 | unidirectional track..... | 16 |
| RWA | 63 | up the route | 39 |

| | | |
|---------------------------------|----|--|
| upload screenshots | | |
| Steam..... | 66 | |
| using | | |
| saved consist | 20 | |
| Utilities | 63 | |
| Utilities.exe..... | 58 | |
| ValidBuildAndDrive | 45 | |
| ValidBuildRoute | 41 | |
| ValidInScenarios | 47 | |
| Via Command | 39 | |
| visibility | | |
| saved consist..... | 21 | |
| wagon numbers | | |
| saved consist..... | 21 | |
| XML Notepad | 59 | |
| year | | |
| quick drive scenario | 35 | |
| yellow marker | 17 | |