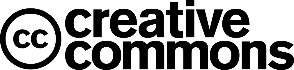


TimetableTool Manual

Rudolf Heijink

Version 0.1, May 2020

***Copyright © 2019 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:***

|  |  |
| --- | --- |
| http://mirrors.creativecommons.org/presskit/icons/by.large.png | **Attribution** — You must give [appropriate credit](http://creativecommons.org/licenses/by-nc-sa/4.0/), provide a link to the license, and [indicate if changes were made](http://creativecommons.org/licenses/by-nc-sa/4.0/). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. |
| http://mirrors.creativecommons.org/presskit/icons/nc-eu.large.png | **NonCommercial** — You may not use the material for [commercial purposes](http://creativecommons.org/licenses/by-nc-sa/4.0/). |
| http://mirrors.creativecommons.org/presskit/icons/sa.large.png | **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](http://creativecommons.org/licenses/by-nc-sa/4.0/) as the original. |
|  | **No additional restrictions** — You may not apply legal terms or [technological measures](http://creativecommons.org/licenses/by-nc-sa/4.0/) 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](http://creativecommons.org/licenses/by-nc-sa/4.0/).
* 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](http://creativecommons.org/licenses/by-nc-sa/4.0/) may limit how you use the material.
* Dovetail Games has been granted a non-exclusive license for commercial use of this guide.

# Preface

***Introduction***

***Images***

Each chapter is introduced with a screenshot I made in game.

***Acknowledgements***

A big thank you to all community members that contributed to the contents of this guide. Many of them will not be aware of their contribution, but lots of information comes from the community forums.

***Frontpage image:***

<Description>

***Disclaimer***

This manual and TimetableTool are provided “as is” the author cannot accept any consequences from the use of Tool and Manual.

The contents is the sole responsibility of the author.

***Contact***

Comments are welcome at [trainsimulator@hollandhiking.nl](mailto:trainsimulator@hollandhiking.nl).

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

If you volunteer to help me developing TimetableTool please contact me!

Enjoy reading!

Rudolf Heijink

Contents

[Preface 3](#_Toc38884867)

[1 Introduction 6](#_Toc38884868)

[1.1 Warnings 6](#_Toc38884869)

[1.2 Donations 6](#_Toc38884870)

[2 Updates 8](#_Toc38884871)

[2.1 New in this version 8](#_Toc38884872)

[2.2 Next version 8](#_Toc38884873)

[2.3 Versions in development 8](#_Toc38884874)

[2.4 Version 0.1 8](#_Toc38884875)

[3 Installation 9](#_Toc38884876)

[4 About timetables 10](#_Toc38884877)

[5 Creating a Timetable 12](#_Toc38884878)

[5.1 Define a route 13](#_Toc38884879)

[5.2 Locations 14](#_Toc38884880)

[5.3 Service directions 16](#_Toc38884881)

[5.4 Services 17](#_Toc38884882)

[6 Reference guide 18](#_Toc38884883)

[7 Trouble shooting 19](#_Toc38884884)

[8 Known issues 20](#_Toc38884885)

[A. Links to documentation and tools 21](#_Toc38884886)

[Index 22](#_Toc38884887)

List of figures

[Figure 1 Classic timetable view as generated by TimetableTool 10](file:///W:\TimetableTool\TimetableToolManual.docx#_Toc38884888)

[Figure 2 Graph view of the complete services timetable for West Somerset Railway 11](file:///W:\TimetableTool\TimetableToolManual.docx#_Toc38884889)

[Figure 3 Route selection and entry screen. 13](file:///W:\TimetableTool\TimetableToolManual.docx#_Toc38884890)

[Figure 4 Completed Route edit form 14](#_Toc38884891)

[Figure 5 A route is selected, which enables some other screens, e.g. locations 14](file:///W:\TimetableTool\TimetableToolManual.docx#_Toc38884892)

[Figure 6 Route layout as shown in the official game manual 15](file:///W:\TimetableTool\TimetableToolManual.docx#_Toc38884893)

[Figure 7 Completed location list for the RSN route. 16](file:///W:\TimetableTool\TimetableToolManual.docx#_Toc38884894)

[Figure 8 Service direction table filled for RSN 17](file:///W:\TimetableTool\TimetableToolManual.docx#_Toc38884895)

# Introduction

TrainSimWorld has a big new feature, Timetable services, where you can choose from up to 350 drives on a single day. In the present version, there is only one timetable, but I believe some day DTG will make it possible to develop our own timetables. West Somerset Railway may be a good example, what about this:

* A day with only some maintenance activities
* A diesel heritage day
* Separate summer/winter time tables

And whatever you can think of. Designing all services for such a day, requires new skills and tools. I believe a tool to plan your services ahead and plot them into one or more timetables is absolutely necessary. You simply cannot test 25 one hour during services properly.

Therefore, I have the idea of creating a timetable tool for some time. Now I take the opportunity to combine this with learning a lot of new C# programming skills.

This first version is still a bit primitive. It works, but it is not yet optimized for easy use. I definitely will add a number of shortcuts, but first I need a stable base and demonstrate the principles on how this all works.

I included two completely worked out timetables:

The services timetable for Heavy Haul (Sandpatch) and the timetable for WSR. This shows how it works and how to use it, while being simple enough to reveal errors fast. These two timetables will be installed automatically

## Warnings

You can use TimetableTool for any game timetable. It is NOT intended to create timetables for real world application and is probably not suitable for that purpose. I cannot accept any liability if you try to use this for anything in a real world.

The first version is only meant to get your feedback. It may not be compatible with future versions and you may need to retype all data, though I will try to avoid such a situation.

## Donations

Until now I never asked for donations for my software. I do not need to make money with them, but as I get more experience, and applications get more complex, my expenses are increasing. For example, I use some development tools that are not free and need to do some additional courses etcetera.

* Jetbrains Resharper, helps me a lot to improve code quality.
* GhostDoc is a tool that helps me to document the source code.
* Courses to become a better programmer

Finally, I will need to purchase code signing certificates. This makes sure you can trust my software.

Therefore, now ask you to donate me if you sue this software:

* For professional developers, I ask **Euro 25 per year**. This is cheap
* For non-commercial use, a one time **Euro 5** donation makes me happy.

# Updates

## New in this version

This the initial version. It has all core functions working, but the user interface is not yet great and you definitely will like more views.

## Next version

I will need to do a lot of code cleanup (refactoring) to make the code better readable and maintainable. I also will create a few shortcuts to make it possible to work faster, maybe I will introduce a workflow to guide you through all steps to create a timetable.

## Versions in development

The source code is publicly available for free at Github. There you can see the most recent versions, but keep in mind they may have bugs and you need to compile the code by yourself. If you want to help me, let me know!

Technologies used:

1. SQLite database
2. Visual Studio 2019 Community Edition, Net Core3.1
3. C# 8.0
4. Caliburn.Micro MVVM library
5. Dapper for Database Access
6. Squirrel to create installation packages
7. Word and PDF for the user manual

## Version 0.1

This version is the initial version. It is not meant for operational use. Just for review.

# Installation

You just need to click the executable to install. When you first run TimetableTool, a database is created which contains two TrainSimWorld timetables:

1. The West Somerset Railway
2. Heavy haul (Sandpatch)

This allows you to get started right away.

TimetableTool will check for updates on a weekly basis, when you run the program and download and install them automatically.

# About timetables

At the moment, TimetableTool will create two different views on a timetable. The first one is the classic timetable, as you can get them in the well-known timetable booklets. You need to create one timetable for each direction and these will not be combined. An example is shown in Figure 1.

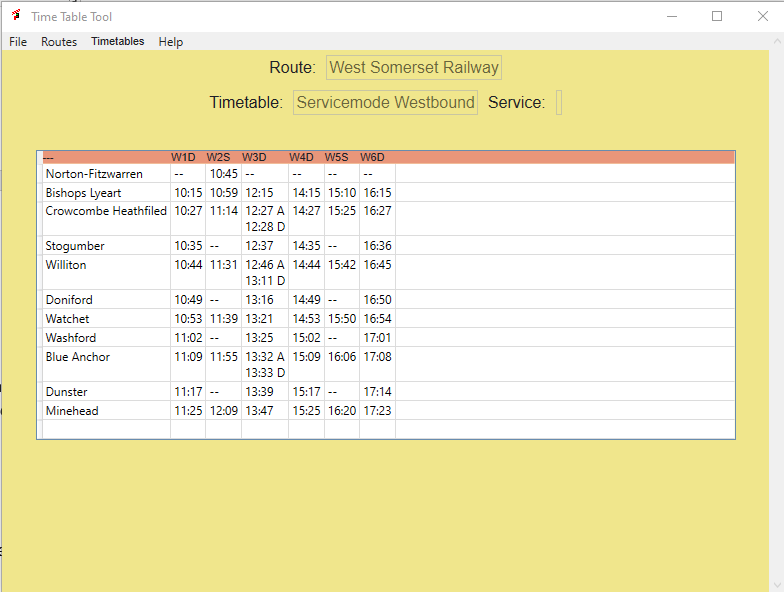
You can export this as a.csv file to process it further outside the tool. You are completely free to represent a number of additional locations. As you can see, if a train stops longer than one minute, separate arrival and departure times can be included. If a train should not stop, automatically two dashes are included to represent this situation.

Figure Classic timetable view as generated by TimetableTool

There is second view, representing the timetable in a graphical way. In this representation you normally include both directions, but it is possible to create a timetable that shows just a single drive or all trains in one direction. It is up to you. This is a really powerful tool, which is also used by real world time table designers.

* It shows the density of trains at a track.
* It shows where trains meet, so how many trains do you actually see?
* It shows conflicts on single track areas

An example is shown in Figure 2.

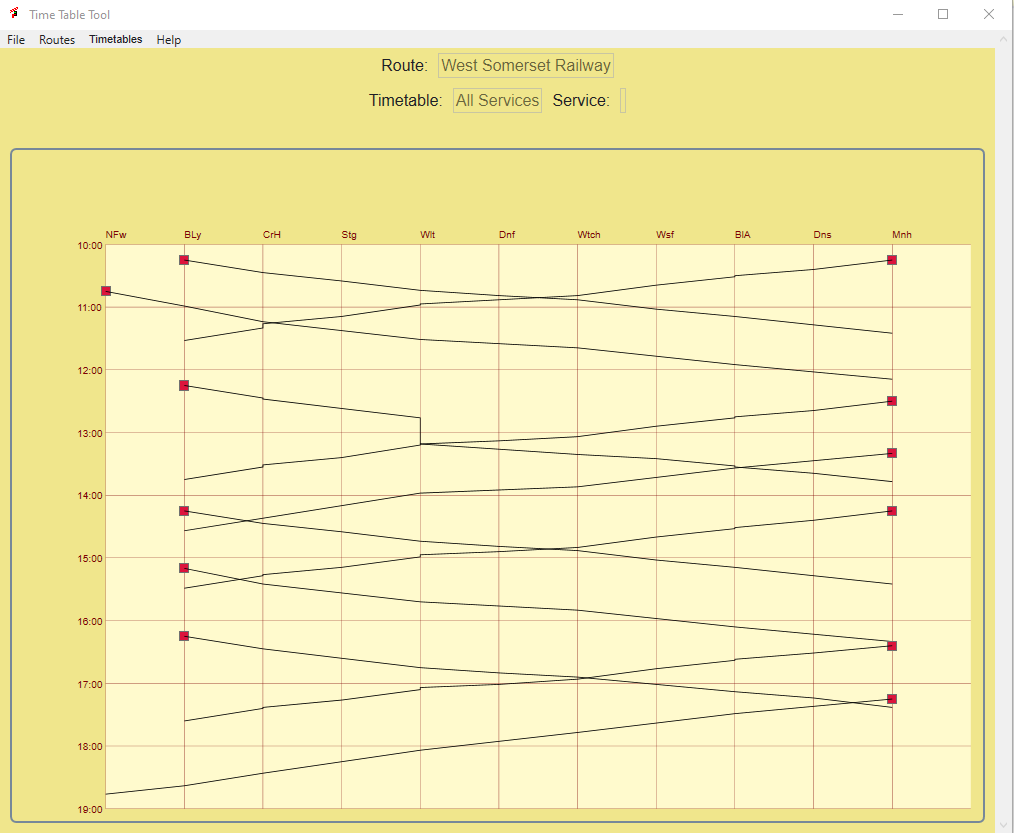
The red squares denote the starting points. At the top you see abbreviations for stopping locations. If you move the mouse over the red squares (which are buttons), you will see a tooltip with additional information. The same happens if you move over the stopping locations.

Figure Graph view of the complete services timetable for West Somerset Railway

Please note, this is an alpha product. I am very proud of the results until now, but I want to make it more interactive and new representations of timetables will be created. If you have specific wishes, let me know.

Some ideas:

* Timetable per stopping location, showing all arrivals or departures
* An Ebula like view, which gives specific information to the driver about speed limits, gradients signals and so on.
* Views that show how rolling stock is moving over the day.
* Views adding crew plans

I only tested this using TSW routes, but you can use it for other train simulators as well and possibly also for bus simulators or other simulation games that require scheduling.

It is NOT intended for real world timetable development. I do not have the professional skill to develop something for that.

# Creating a Timetable

TimetableTool is a design tool for timetables and works mostly bottom up. I have specifically timetable mode for TrainSimWorld in mind. This requires the ability to repeat a drive several times on a day. This way of working is supported by TimetableTool. At the moment, it requires al little bit additional work, but I will someday provide some more shortcuts to make it go faster.

For now, you need to perform the following steps:

1. Define a **route**. All timetables are bound to a route.
2. Add an number of named **locations** to the route. A location can be a station, a siding or any other point of interest. You also must define an **order** for the locations.
3. Define **directions**. For instance, you may drive from west to east or the other way round. The tool must be able to distinguish between the different directions. In most cases two directions should do.
4. Define **services**. A service is a container to hold stops, driving times but not a fixed departure time. This make sit possible to run a service multiple times. Defining services is to most work, so I included some optimizations.
5. A **service instance** makes the service a real drive, by giving a service one or more departure times. In the graphs you see service instances.
6. Bundle a number of service instances in a **timetable**. A timetable is what you finally want to see. You can have as many time tables as you like! It is nothing more than a specific view on the data.

You need to execute these steps in the order shown above. TimetableTool will force you to do so, by disabling menu items that are missing prerequisites. In the rest of this chapter, I will clearly indicate these requirements.

|  |  |
| --- | --- |
|  | This chapter has a tutorial nature. In the reference section will cover a lot more details on how it works what you can do. |

## Define a route

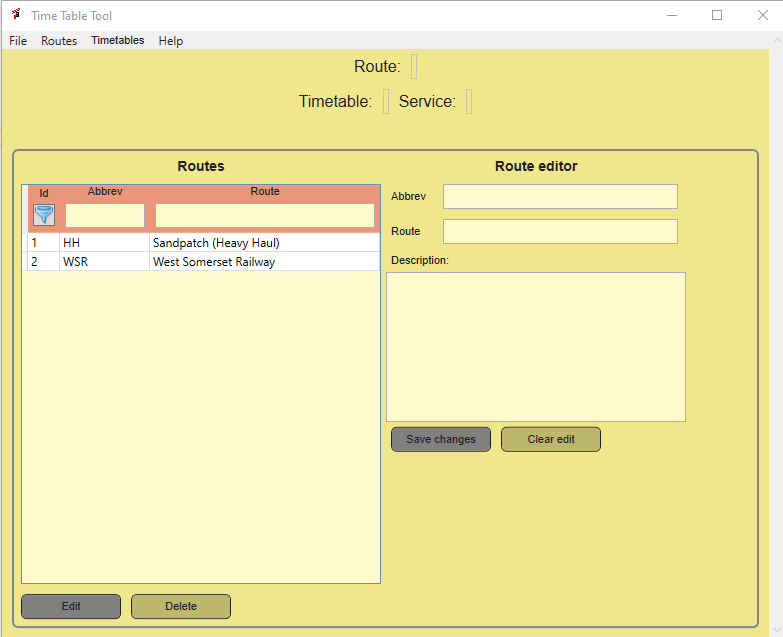
In Figure 3 you see the screen as it looks like when you first open TimetableTool. For this tutorial, we will create a small new timetable for the Ruhr-Sieg-North route.

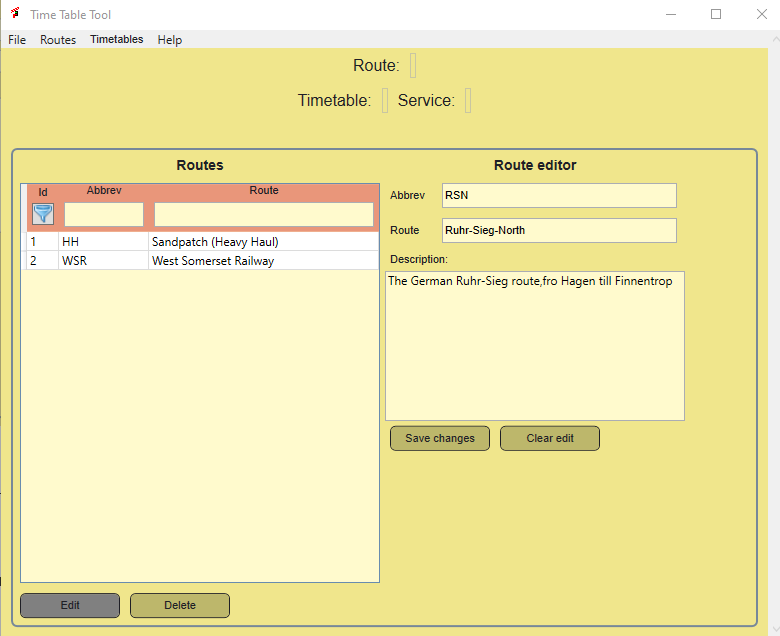
Figure Route selection and entry screen.

At the top of the screen, you can see the name of the currently selected route, service and timetable. As you see, nothing is selected and in the left pane you see the routes for the demo timetable. We can ignore this for now.

To enter a new route, complete the form at the right. All three fields are required. Once you have done, the Save changes button is enabled and you can save the route into the database. It will appear right away in the route list in the left panel.

|  |  |
| --- | --- |
|  | The Delete functions have not yet been implemented, so once your route is saved, you can edit it, but you cannot delete it at the moment. |

Now, you need to select the route you just created, to enable the locations screen.

Figure  Completed Route edit form

## Locations

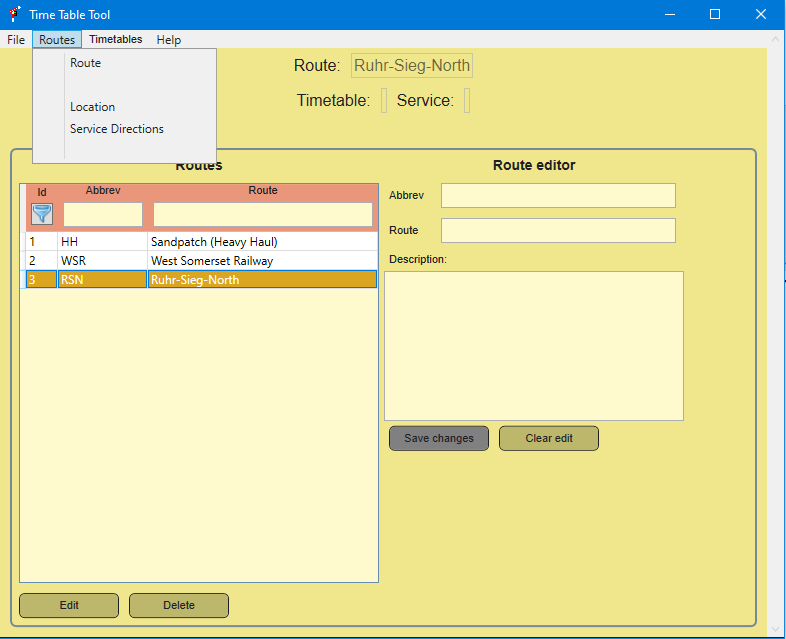
If you selected a route, at the top level, in the Routes menu the menu-item Location will be enabled. In the route table you see the orange background, indicating the route is selected and the name of the selected route is shown in the text above your screen.

Figure A route is selected, which enables some other screens, e.g. locations

No you can open the locations screen, using the menu. The locations shown are filtered by route. As you just created the route, there are no locations yet, so the table is empty.

Because we use a database, and you can add locations afterwards, you need to tell TimetableTool in which order trains may pass the locations. You do this by assigning each location a number, called Order. Locations will be sorted according to this number. The value is not critical, it must be a number, that is all. I recommend to increase location numbers by 10.

|  |  |
| --- | --- |
|  | In the reference guide I will explain how to work with branches, like in the Rhine-Ruhr Osten route. For now we keep it simple. |

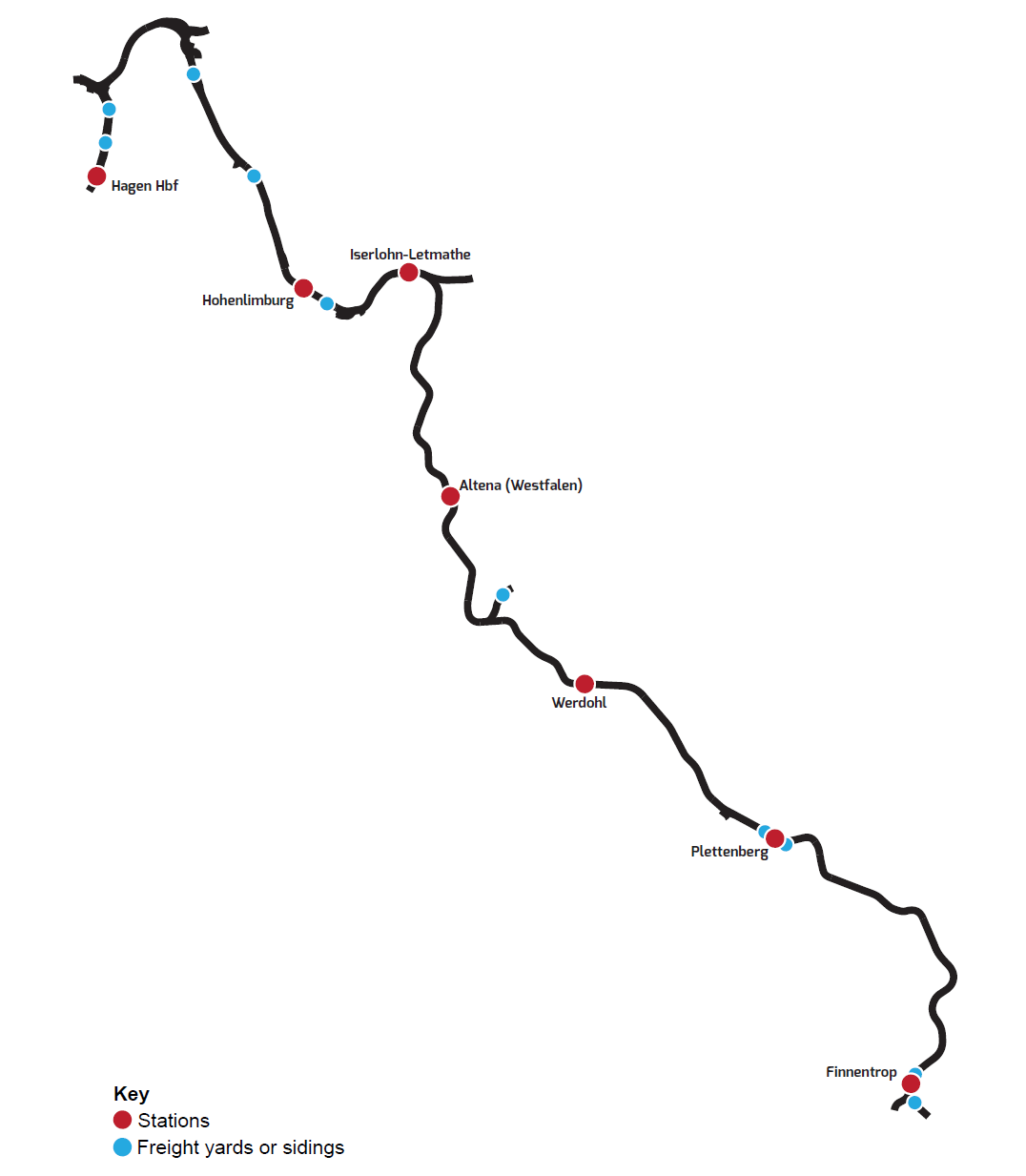


Figure Route layout as shown in the official game manual

In Figure 6 you see the route as presented in the game manual. Freight yard did not get names. This is not really bad. You can change names for locations later, if they are incorrect, without having serious impact on the timetables.

I decided to include the line ends as locations. AI trains may depart from Finnentrop to the line end. I also included the freight yards, because we will need them to add freight services. It does not matter where you start, I decided to start at Hagen and then number southward to Finnentrop.

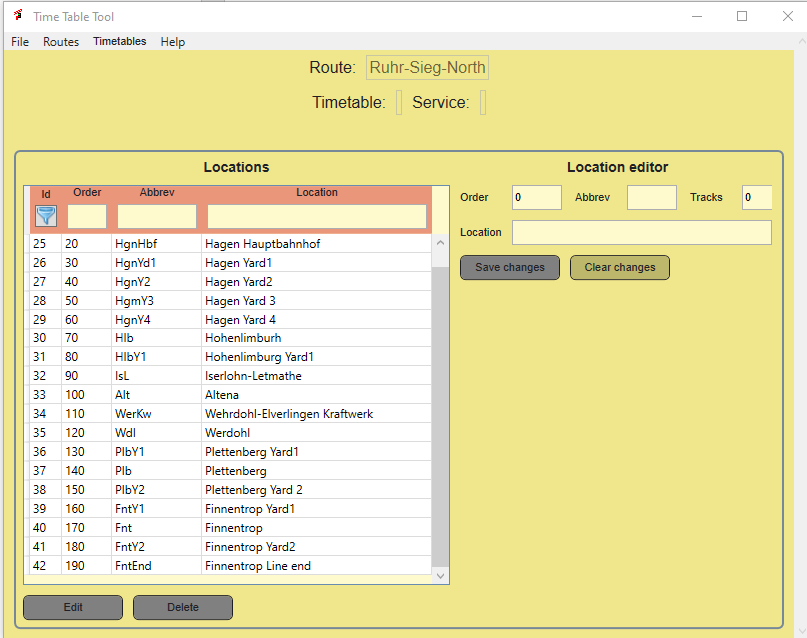
The completed list may look like Figure 7, depending upon your choices. I used fantasy abbreviations, but if you know the official abbreviation, you can use them as well. There are no restrictions, but I recommend to keep them short, less than 6-6 characters. You also must fill the number of tracks. The basic idea is that this can be used to have an idea about the capacity of the location. This is especially interesting for single track routes. I am not yet sure if I want to keep this information.

Figure Completed location list for the RSN route.

I increase the order by 10, to make it easier to add a location later.

There is no need to select one of the locations. You can now proceed to set up the service directions, which is straight forward.

## Service directions

Service directions refer to a route. So you must have selected a route in order to access this form. You can open the form through the Routes menu

It is straight forward, create a name and an abbreviation. These are only used for reference. Internally the database identification is all TimetableTool needs and you need to tell if the direction should follow the order of the locations from low to high or from high to low (descending order).

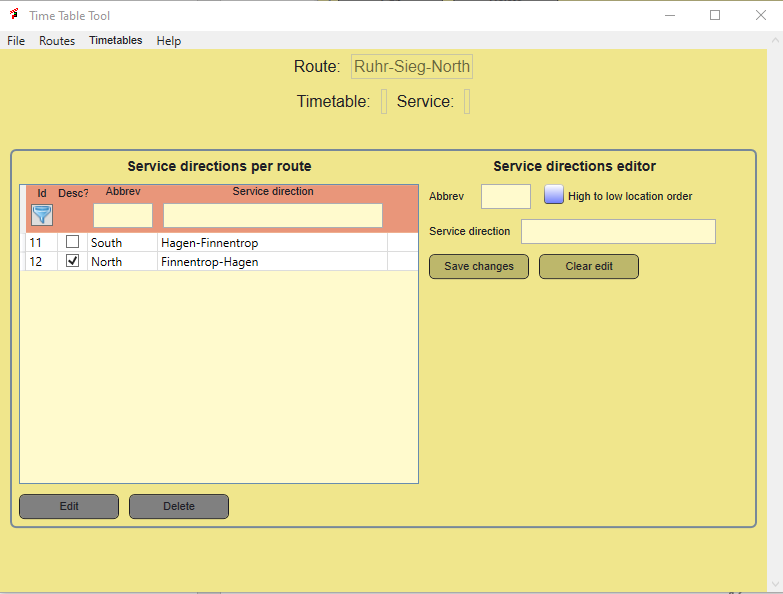
Normally two directions are sufficient. As you see, in the abbreviation I use the global compass direction, which is described more precise in the name. The “Desc” field or High to Low Location numbers tells TimetableTool to start with high location numbers for trains from Finnentrop to Hagen.

Figure Service direction table filled for RSN

Once this is done, the route details are set. We now can define services.

## Services

I order to add services, you must have selected a route. The, at the Timetable menu you can click at the services menu item. Do NOT confuse this with the services instance menu item!. The services page allows you to define a set of re-useable services.

For RSN I used the timetable you can download here as basis:

<https://www.railsim-fr.com/forum/index.php?/files/file/1449-tsw-fiches-horaires-ruhr-sieg-nord/>

We will create 3 services in each direction, but note they can be re-used.

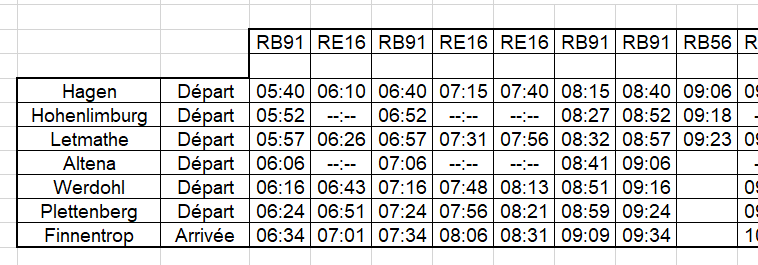
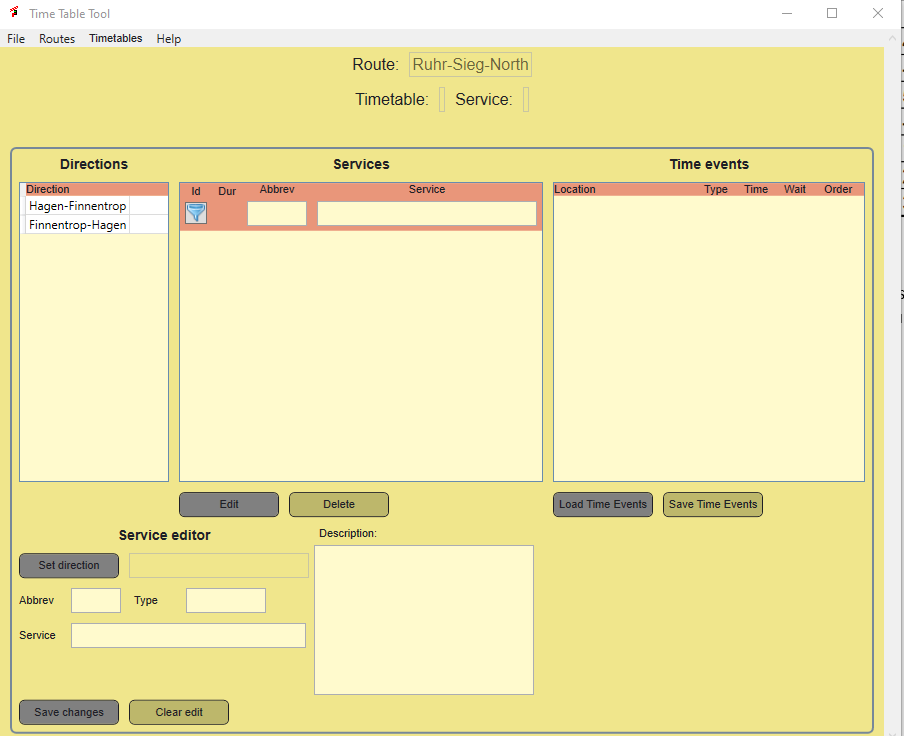
When you have look at this timetable, You see there actually are just three services: RB91, RE16 and RB56. The others are just copies. This step is by far the most complicated procedure. TimetableTool has a shortcut here to make it much easier to create these service, as you will see soon.

Figure Services form

Figure Part of the North-South passenger timetable for RSN

As you see, the window is divided in three tables at the upper part and has an edit window at the lower part. At the left, you see the directions that are available. You must tie each service to a direction, which is done by selecting a direction and then add it to the service, using the Set Direction button in the Service editor.

The other fields are trivial. You select an abbreviation (e.g. RB91), a descriptive name and there is room for a more extensive description. All fields are mandatory. If they are all set, you can save the new service, using the Save changes button.

Now you can select the service. This step will fill the third column with all locations. You can directly edit this table, which makes it a lot faster that the old way. The old way is still available, see the Reference Guide chapter for details.

I will demonstrate the RB56 service first, because it is very simple.

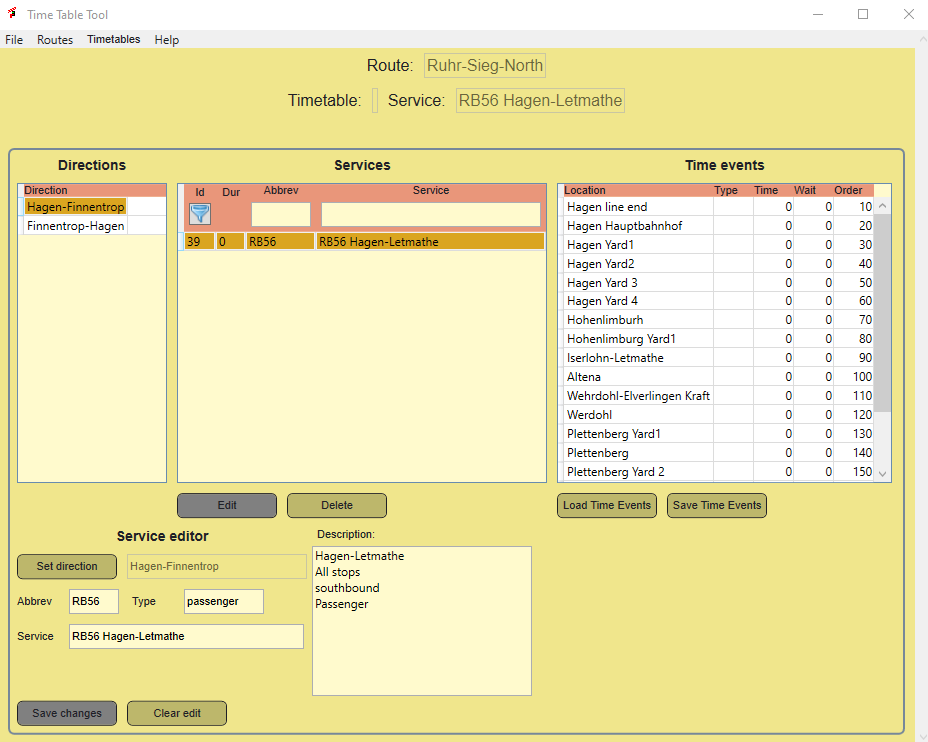
In Figure 11 you see the results of the first steps. A service is created and saved. Then the service is selected and prepared to edit (Edit button). Once you have done that, the button Load Time Events is enabled. If you press that button, it will load either the previously defined time events or load all locations, so you can create time events. The filed type should have the values passenger or freight.

Figure RB56 service defined and time events loaded

This is great. We must have a closer look at the time events table. This is where most work is needed and where it happens:

|  |  |
| --- | --- |
| Field name | Description |
| Location | You see the location name here, but behind it are all details in the location. Locations are sorted by order and take into account the service direction. |
| Type | The type of the event. If you leave it empty, the location will NOT be saved as Time Event. I use following codes:   * S=start, where the service starts * E= end, where the service terminates * H= halt passenger stop * P= pass, pass through with a set time * -- = keep the Time Event at this location, but it is not scheduled. |
| Time | The relative arrival time from the last scheduled location till this location in minutes. For the first location, it always should be 0. |
| Wait | The waiting time at the location. This is optional. If it is zero, TimetableTool assumes arrival time is equal to departure time. If a value is set, separate arrival and departure times will be generated (as you can see in the WSR timetable). |
| Order | This is NOT the order of the locations, but the order of the Time Events. It is generated automatically, but you can change it and changes will be preserved. The nice thing is that it takes the service direction into account. If you have set to descending checkbox, it will reverse all. |

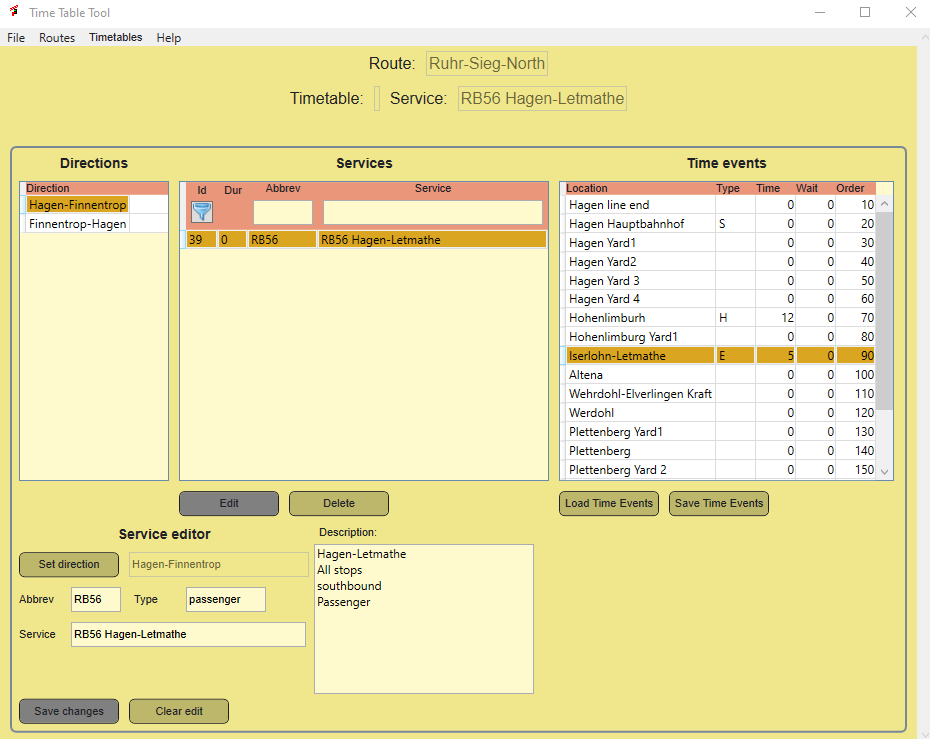
You now can edit the time events directly in the table. For the service we create, the result should look like this:

Figure Completed time events for service RB56

You see, I set Hagen Hbf as starting point, giving the Type the value S.

Then, you need to calculate that it takes 12 minutes to drive to Hohenlimburg. This location gets H for Type and 12 for time.

Finally, it takes 5 more minutes to drive to Letmathe, where the service ends, so I make that clear by putting S in the Type.

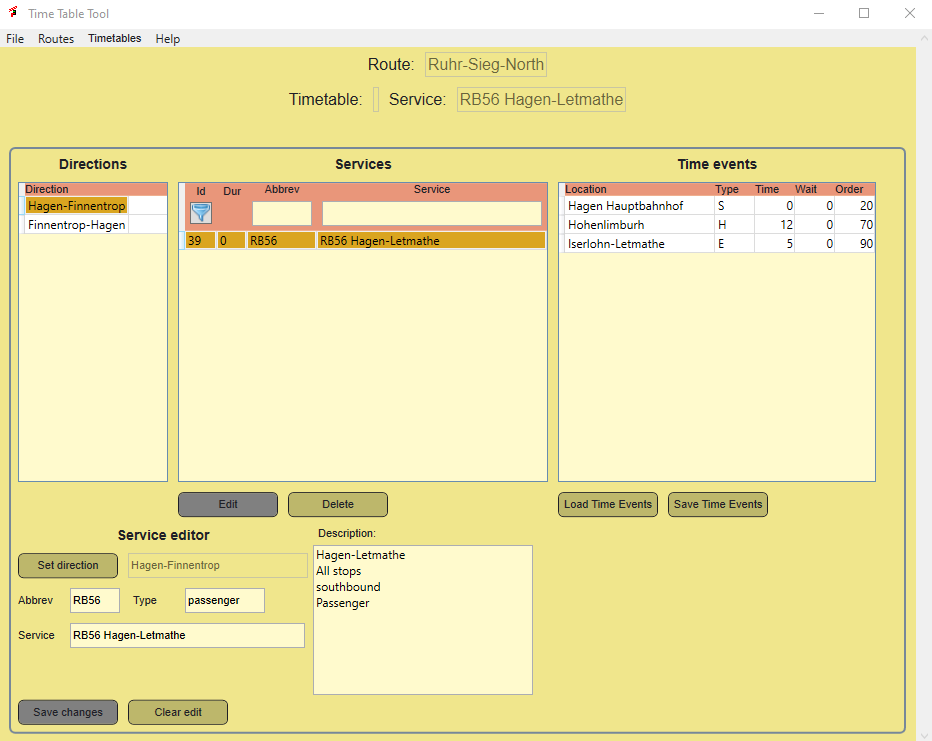
Now you should save the Time Events. Resulting in this screen:

Figure Saved time events

You see now that all time events we no longer needed are kicked out again. The service should show the total driving time, but this is at the moment of writing not working well. It is calculated but not shown right away.

|  |  |
| --- | --- |
|  | Service type and Time event type are not used at the moment. It is possible that they will get more clear definitions. I recommend to use these fields as described here. |

I the same way I will now create the other two services and show just the time events:

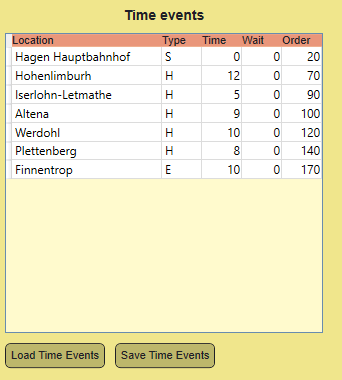
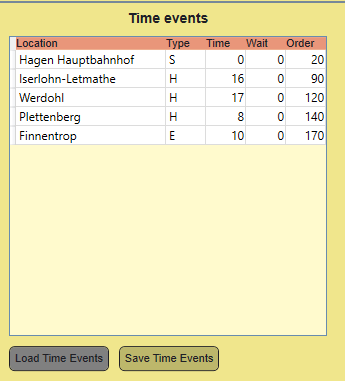
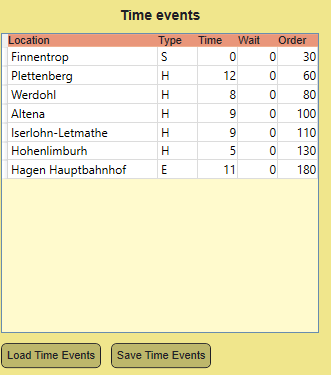
For the total demo, I will add three services for the other direction as well. I see now, DTG has used the same service names. That is not a very good idea, there I changed the abbreviation to include the direction.

Figure RB91 en RE16 services

So, RB91 will be RB91S and RB91N.

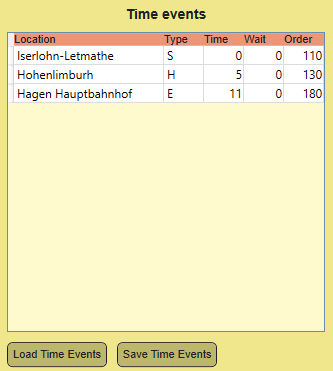


Figure Two of the three northbound services

# Reference guide

# Trouble shooting

# Known issues

1. Links to documentation and tools

This guide, other guides for TSW and TSWTools are available here:

|  |  |
| --- | --- |
| Topic | URL |
| Holland Hiking | <http://www.hollandhiking.nl/trainsimulator/index.php> |
|  |  |
|  |  |
|  |  |
|  |  |

# Index

**Geen indexgegevens gevonden.**