TSWTools

Toolkit for Trainsim World

Users guide

Rudolf Heijink

Version 0.3 alpha

Preface

Introduction

The last years I created a number of manuals and tools for DTG TrainSimulator. Mike Simpson, the author of the world famous RWTools has been an important source of inspiration to me. I never tried to copy his work, but I found some niches specifically for scenario authors that are not covered by RWTools. Mike announced he will not create such a toolkit for TSW. So I decided I could claim the TSWTools name with a lot of respect and thankfulness to Mike. I cannot but admire his perseverance in reverse engineering undocumented features without any support from DTG.

Here it is, the third alpha edition of TSWTools. It's not doing much yet, but the start is there.

Acknowledgements

All anonymous members of the TSW community for sharing their experience and helpfulness.

The creators of Inno setup for providing a free installer.

Disclaimer

This guide is provided "as is". The author is not liable for the consequences of the use of this guide or the LuaCreator application. 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.

Front page image

One of my own screenshots made at the Sandpatch route. It still has a "wow" effect on me seeing all the details.

Rudolf Heijink

Contents

List of figures

1 Introduction

There is not yet much you can do to enhance your experiences with TrainSim World, but few things are useful and justify a tool. TSWTools will do this for you:

- Unpack all .pak files and make a local unpacked copy.
- **NEW:** View uasset file contents using the UModle application
- View the key bindingsfile, used in the game
- **NEW:** you can now edit most game settings in the tool, including enhancements like Viewdistance and extended sound levels. You make sets and save them outside the game
- **NEW:** Make backups of your user settings and progress and restore them
- **NEW:** Manage you additional liveries and add them to the game or remove them again
- A screenshot manager, that combines steam and TSW screenshots for easy selection.
- A game launcher, using a settings file and optionally start Sandpatch Radio.
- Browse through the file locations

NEW: This version has a redesigned user interface, using WPF technology.

1.1 Next version

In the next version I will provide a better integration of the livery manager. Also I will see if it is possible to use your own screenshots as launch images. The third improvement will be an installer, which you can use to add livery paks to the game or to the livery library.

Maybe, if the content editing tools are available, I may provide some support functions, but nothing can be promised.

1.2 New in this version

1.2.1 Version 0.3

New functions:

1. A complete redesign of the user interface
2. Make a backup of your local game files (settings, progress, last played, screenshots ….)
3. A livery manager. You can store your additional liveries in a library and install them from this library.
4. A tool to edit most game settings, including some Unreal settings recommended by community members
5. Improved unpack tool
6. Improved game launcher
7. An interface to the UModel application, which you can use to view UASSET files
8. Error log screen

1.2.2 Version 0.21

- Fixed an annoying bug in the options menu that prevents to select the steam and TSW installation directory.
- Unpacking all TSW .pak files now processes all paks in parallel and shows a “busy” indicator while it is working.
- You now can unpack a single TSW game file (.pak type).

1.2.3 Version 0.2

- Unpacking game files also works for DLC .pak files (all files will be unpacked)
- The screenshot viewer is improved, it will now handle your user id properly in most cases
- You can now save and restore several settings, e.g.
- A game launcher, which will load your options set and start Sandpatch Radio if you want that.

1.2.4 Version 0.1

This is the first version. It does some basic stuff:

- You can easily unpack the game .pak files.
- You can view the files at the game location as well as the unpacked files
- There is a simple screenshot manager
- You can view the input mappings (but not edit them)
- You can view and edit the game settings outside the game.

2 Installation

2.1 Installation procedure

The game comes with an installer. Installing it is straight forward. The first time you run TSWTools, you must set some options in the game to enjoy all functions. See section 5.1

It has been tested using Windows 10, but likely will work in other environments as well. TSW Tools is written in C# and requires the latest .NET version to run.

You need to install the Unreal game engine and the UModel tool for some functions.

In order to read the manual, you need a pdf reader (by reading this it is clear to me you have that).

It is recommended to install a good text editor as well. Notepad++ is a good solution.

Check out www.hollandhiking/trainsimulator for download links.

2.2 Folder structure

At the next page you can see the folder structure for the data TSWTools may store. The folders are created automatically after your first login. In rare cases this does not work, you can create them manually.

Backup is used to store saved backup sets.

Liveries contains the liveries. You are free to make a further subdivision.TSWTools will scan all subfolders to locate content.

Manuals is the place to go for documentation. In the root folder, you should find this manual, the starters guide and licence information. Inside the **RouteGuides** folder you can add subfolders as much as you see fit.

OptionsSets will contain the Options collections as you create them.

Temp is for temporary files. You can delete the contents if you need to.

Unpack contains the unpacked .pak files, each of them in its own folder structure. In the folder **UnpackedAssets** the UModel application should store its exports.

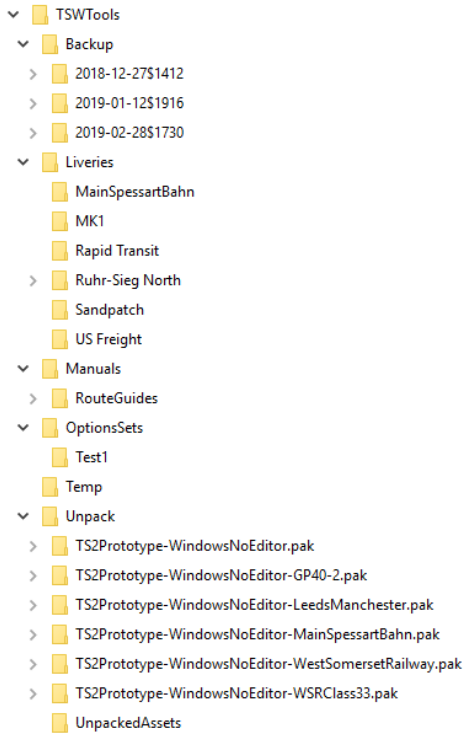


Figure 1 Folders used in TSWTools

2.3 Database

TSWTools uses a database to preserve some information for you. You can find it in the TSWTools root folder and it is called TSWTool.db. If it is missing, TSWTools will create it again for you, but all data will be lost.

If you want, you can open the database and see for yourself what is inside it, use the freeware tool you can download here:

https://sqlitebrowser.org/

3 User interface principles

In this chapter a short “buttons training”. This may help you to understand how the new user interface is designed and how you can use it. LuaCreator is a fairly complex application. It is not 100% fool proof, so handle it with a bit of care. The user interface also has more or less hidden features, you may consider convenient.

Note: this chapter is copied from another tool I created, called LuaCreator, which uses the same principles for it user interface. I did not yet update all examples specifically for TSWTools.

3.1 Modal versus modeless windows

For Windows application, there are two ways to open a new window: **modal** or **modeless**.

A **modal window**, also called dialog form, needs to be closed using either the **OK button**, which usually saves data or a **Cancel button**, which cancels all changes made in the form. Access to all previously opened windows is blocked, while you are working with this window/dialog.

A **modeless window** spawns from its parent window. You can navigate freely between the windows you have open.

The advantage of modeless windows is a much larger flexibility and freedom during use. Freedom has its price, your desktop may soon be cluttered with a large number of open windows, and it is up to you to keep track of them. If data between these windows is related, changing data in a modeless window may cause inconsistency in other windows. Fortunately, WPF has some useful technologies to update all relevant windows automatically.

In the new WPF version of LuaCreator I decided to switch from Modal windows to modeless windows, with a very small number of exceptions. The main reason to do so, is that you always have access to the information you need. This works, because the logic in LuaCreator is (almost) completely separated from the logic to show data on the screen.

However, you can create situations where inconsistencies in data may occur. In this case, close the application and restart it.

You can close each window separately (using Alt+F4 or the cancel button in the top right position), all spawned windows will stay open. It is not a very good idea to close the startup window, at the moment there is no method to reopen it again, without closing TSWTools.

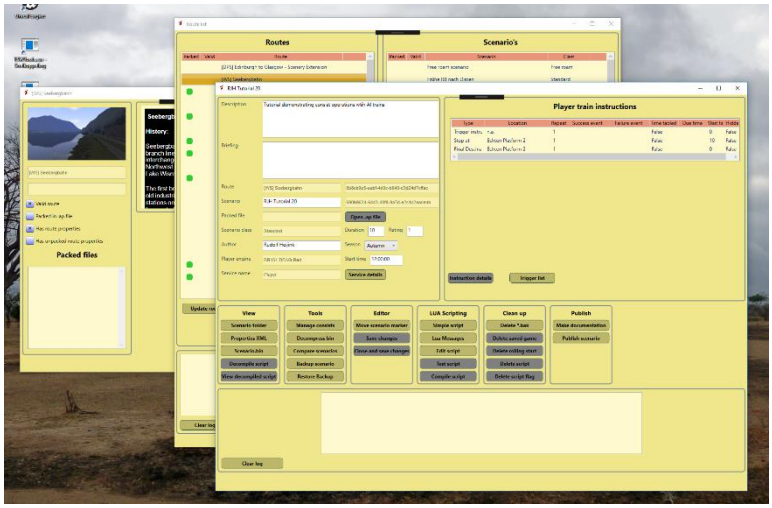


Figure 2 Modeless windows

3.2 Impact of the window size

I created the new WPF version using a fairly large screen, with a resolution of 2450x1440 pixels. As I understood it, WPF would scale nicely for other screen sizes. Unfortunately this is not true.

So, when I tested TSWToools at my laptop screen (1920x1280 screen), the windows did not fit on the screen, which is not workable. For the short term I adapted some screens to fit better and adapt the content size a bit. I also added scroll bars to each window. Therefore if you work at a smaller screen it should be possible to use LuaCreator by scrolling. A consequence of all this is that full screen mode maximizes now to 09% of full screen.

For the next version, I will improve this a lot. This may take time, because for these scaling issues no tutorials exists and it seems little knowledge is documented. So I will need to find out how to do this exactly. Don't worry, I have some ideas.

Note: When working on a smaller screen e.g. a laptop screen, you may need to use the scroll bars to be able to use larger windows.

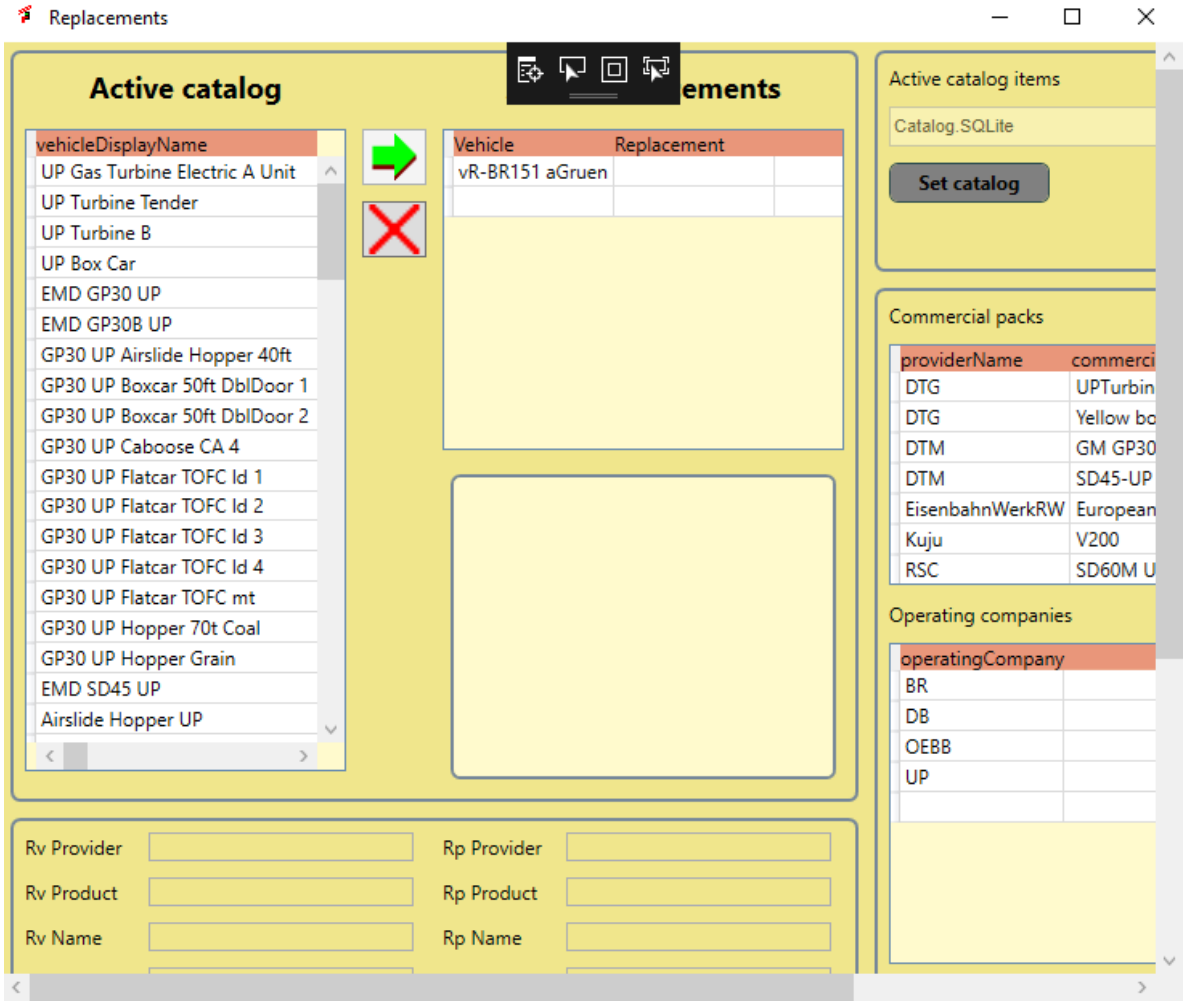


Figure 3 Example of a window with visible scroll bars

3.3 Controls

TSWTools uses a large number of controls. WPF makes it relatively easy to give them similar look and feel. In LuaCreator simple styling is used, later these may be replaced by more advanced and fancy options. For now, it is mainly functional, so it is helpful to understand the concepts that are used.

Figure 4 Buttons

Buttons are used to start an action. TSWTools uses colour codes to tell you more on the significance of a button:

- **Pink** is used for a **Cancel button**. This is mainly useful for a modal window and results in abandoning the planned changes.

- **Green** is used for the **OK button**, which usually makes changes permanent. For a modal window, an OK Button will close the window as well.
- **Grey** is used for a **Disabled button**. Pressing it has no effect, mostly be cause condition is not met. For instance, You cannot show Scenario Properties if no scenario is selected.
- **Dark Khaki** is used for **Normal buttons**. They will execute the indicated action.

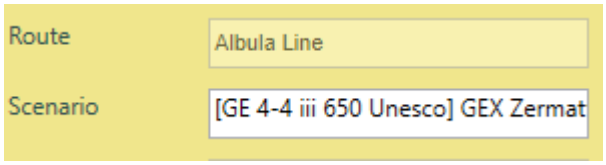


Figure 5 Text Box

In Figure 5 you see a four elements. The left column contains the meaning of the right column (in this case Route and Scenario). The right column contains the actual value from the game data. You see the **Border line** for the text. If the background is **yellow**, this means you cannot change the text, if the background is **white**, you can edit the text in this field.

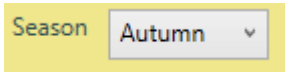


Figure 7 Combo box

A variant of this principle is a **combo box**, which shows a predefined set of values, e.g. the seasons in this example. The background is white, so you can edit it.

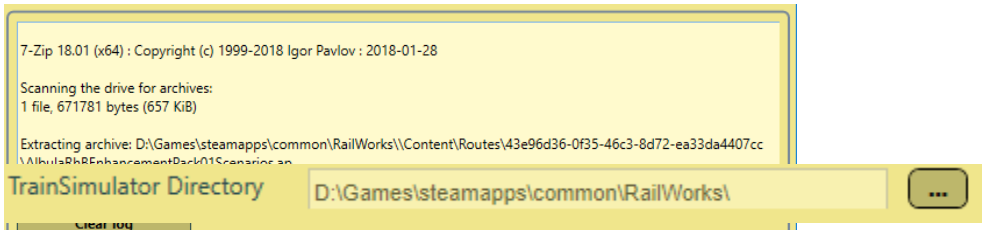
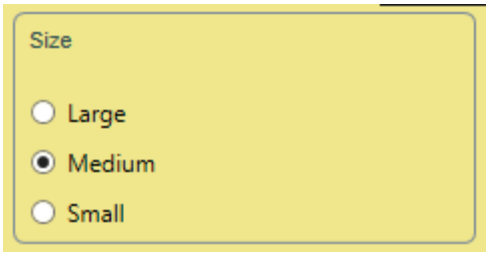


Figure 8 File Dialog

Figure 9 Result Text Box

A special case is the **Result Text Box**. This Text Box has more than one line and it will automatically scroll to the last line when its contents changes. In many cases there is a Clear Button nearby that wipes all text. It is used to inform you about results of actions where needed. It's use is not always consistent yet, This will be repaired in next versions. On most windows you will find one.

A **File Dialog** also is a special case. It is used to select a **file** or **directory**. The **actual value** is shown in the Text Box, but as you can see you cannot directly edit it. If you want to change it, press the button at the right side showing three dots. This will open the well-known standard windows dialogs for selecting files or directories.



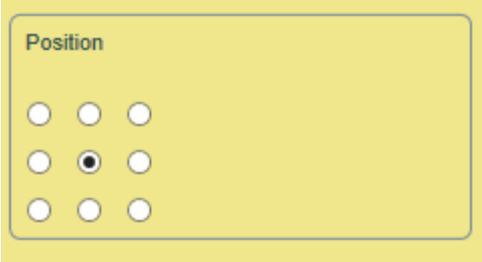


Figure 9 Radio buttons

Radio buttons are sets of small round buttons you can check, but you can check only on out of a set. A set is surrounded by a thin border line. In one case, you see there is not any text. In this case each button represents a position at the screen for a Lua message. This will be explained later.

Tooltips are short help texts. You may see them at some screens. I intend to add them to all windows. Unfortunately, it is not possible to capture them easily.

TSWTools works a lot with **tables**. I have used two different technologies for tables:

1. **List View**
2. **Data Grid**

The bad news is that you cannot see which one I used. Data Grids off a bit more flexibility than List Views. In the example below, you see the **Route List Window** which has two **List Views.**

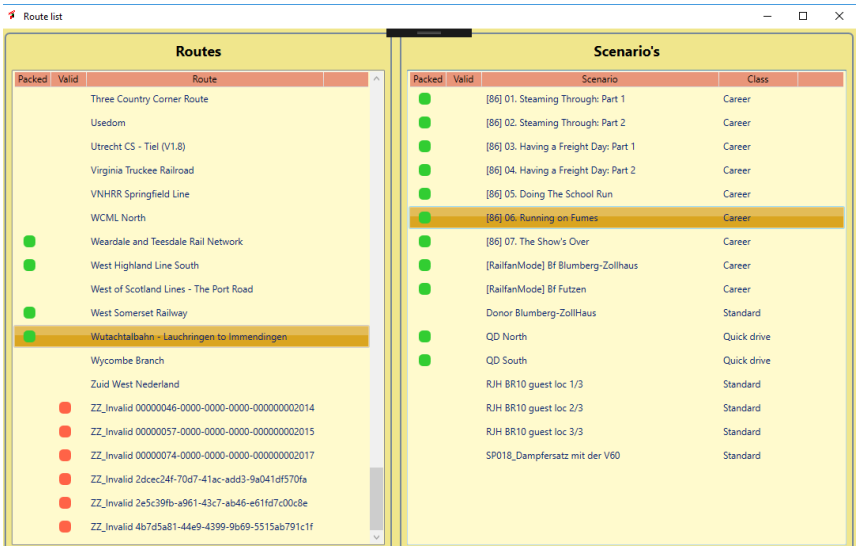


Figure 10 Two coupled List Views

Remarkable features (all of these are also valid for a Data Grid):

- The first line with the **Pink** background contains the explanatory **column headers**. You can make the columns wider or smaller if you need.
- A **selected line** is highlighted by giving it an **Orange** background colour.
- If the **number of rows** exceeds the space allowed for the table, **scroll bars** will be shown automatically.
- The **green** and **red blobs** act as a **check mark**. In this case a green blob means the route is packed in .ap file The red blob means the route is not a valid route.

A **Data Grid** has one additional feature:

By **clicking** a column at a header row you can sort the table using this column.

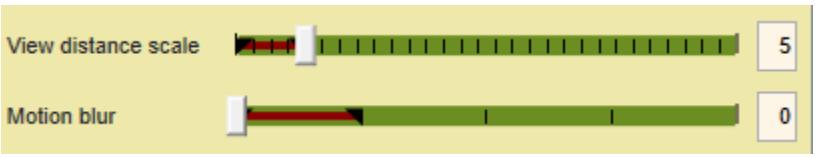


Figure 11 Slider control

The **Slider control** is used to select a value in a range. The brown area is a cosmetic feature and represents a recommended range. Please make sure to consult the documentation on the interpretation of this range. The tick marks show the granularity of the settings. In most cases the selectable values are restricted by the software. At the right side, you seen an TextBox. You also can type the value directly in the TextBox, but this is not the recommended practice.

As a last control, I will introduce the tab control. This looks a bit like a button. What it does is that you can hide parts of a window and show only one of them. For instance, the options window shows either all file locations or the other options. The main advantage is that it saves screen space and makes the window more compact.

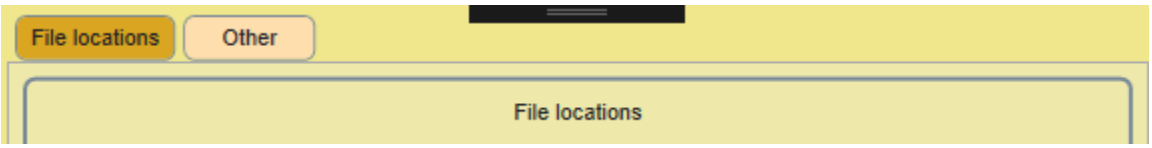
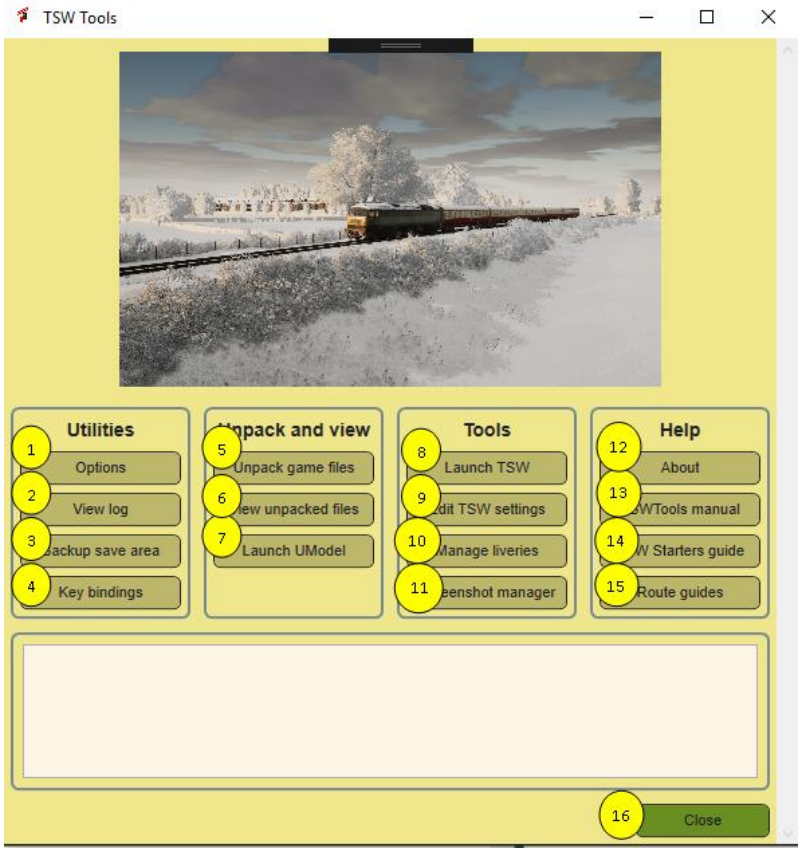


Figure 12 Tab control

4 Main screen

In Figure 13 the TSWTools main screen is depicted. Each function will be introduced in the subsequent sections.



1. Options, tool options. You need to set the options before you can use TSWTools
2. **New** For errors, TSWTools provides an error log, which you can open here if needed.
3. **New** Backup tool for the saved user data, screenshots etcetera
4. View the key bindings in a neat table format
5. Unpack the game .pak files using the unreal unpacker (takes a lot of time!)
6. View the unpacked files using explorer
7. **New** Interface to the UModel toolkit, to view uasset files
8. Game launcher
9. **New** Edit the game options and save options as a set
10. **New** Manage your additional mods and liveries
11. **Improved** The screenshot manager
12. About this tool
13. Open the TSWTools manual
14. Open the TSW Starters Guide
15. Open route guides
16. Close button, terminates TSWTools

5 Utilities

5.1 Options Dialog

Before you can use TSWTools you need to set the options. Normally you only need to do this once. Options are stored in the registry.

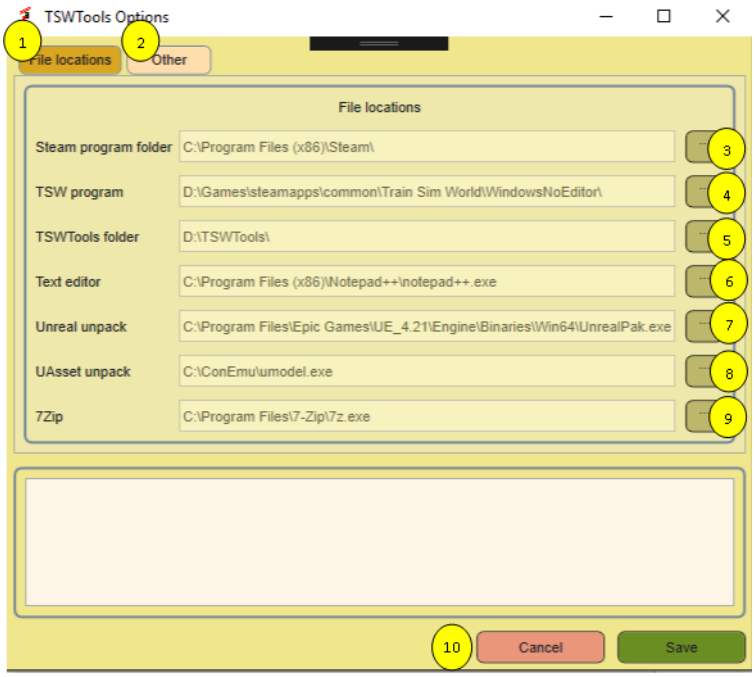


Figure 14 Options dialog, File Locations tab.

1. **File locations tab**. The options window is a modal window and it has two tabs. The active tab has a brown colour. The not active tab is red. File locations is the tab for setting all file locations.
2. **Other tab**. The second tab is for other settings.
3. **Steam program folder**. Set here the folder where you installed steam. This is needed for the screenshot manager. This NOT always the folder where steam installs games. In the example you see that at my computer Steam is installed at the C drive, but TSW is installed at the D drive.
4. **TSW Installation folder**. Click at the three dots to open an Open File Dialog. Make sure your path ends with "WindowsNoEditor" otherwise it will not have the desired effect and some functions of TSWTools will not work.
5. **TSWTools folder** is the folder where TSWTools will install its datafiles, e.g. the unpacked game. So this folder requires a lot of space. See section 2.2 for details.
6. **Text editor**. Notepad is used as a default, but I recommend to choose a better option, e.g. Notepad++
7. **Unreal unpacker**. Is the unpacker in the Unreal Engine. If you have the engine installed by using defaults, you probably do not need to do anything. Check out mu website for some additional information on installing the Unreal engine.
8. **UAsset unpack**. Here you need the UModel program. Please check out my website for the download location and additional information.
9. **7Zip**. I think this is not used now, but will be in near future. Here you can provide the location where 7Zip is installed. You need to point to **7z.exe** and not to one of the other programs that come with 7Zip.
10. **OK button** saves your changed settings, **Cancel** will cancel all updates in settings

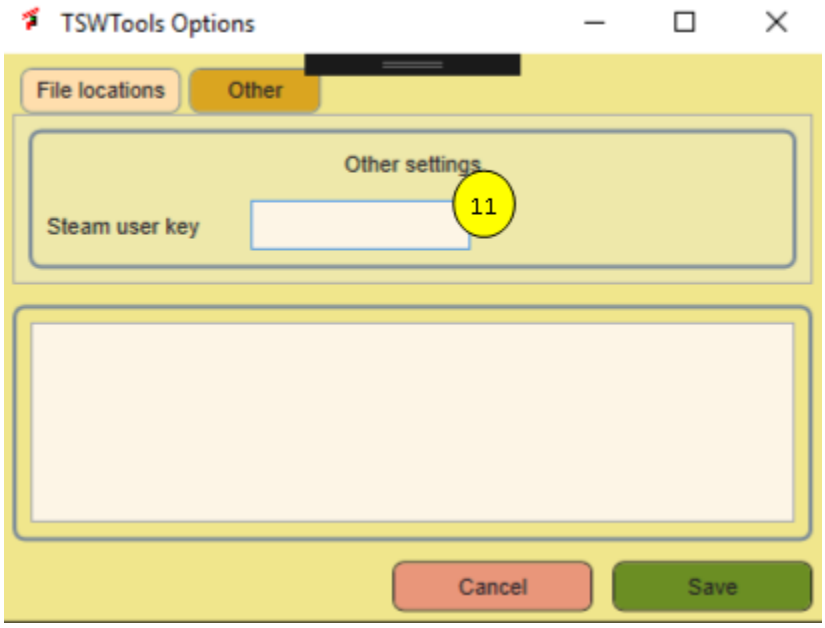


Figure 15 Options, Tab Other visible

11. **Steam User Id**. You need this ID to get the steam controlled screenshots. TSWTools tries to guess the correct id by inspecting your hard disk, but in case it does not work, you can set it manually.

5.2 Log viewer

The Log Viewer tells exactly where errors occur and include the error message. It is a non-modal screen and you can leave it open when needed.

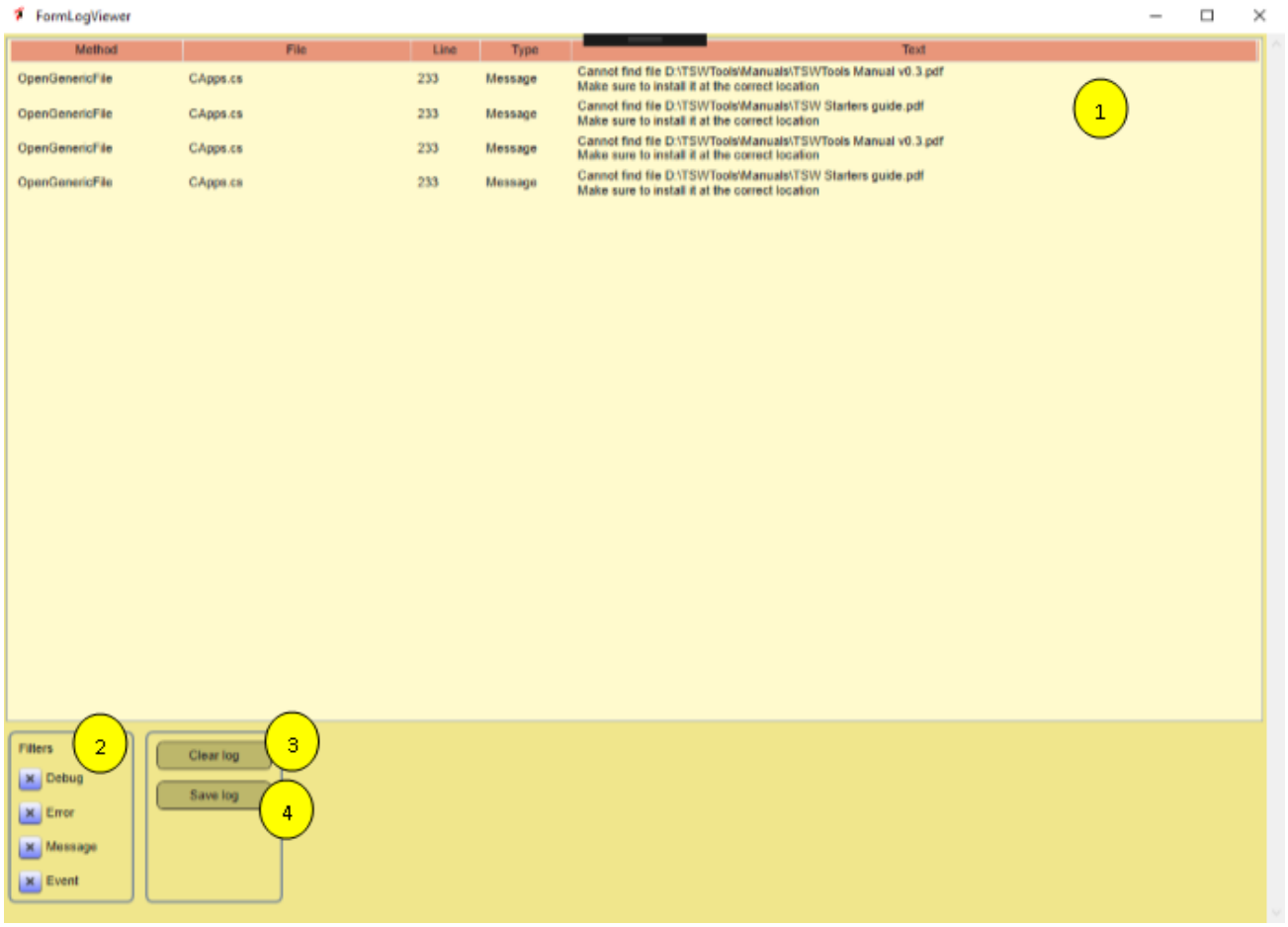


Figure 16 Log viewer

1. Here you find the actual log. It will be updated for each event automatically, no need to refresh it.
2. You can filter what you want to see. Default is all message types on.
3. This button will clear the contents of the log
4. Click here to save the logfile. You will be asked to provide a file name.

Note: if you want to report a bug, always send me a logfile. This helps me a lot to find out what is going wrong.

5.3 Backup save area

I happened once to me that the saved game data was corrupted. I deleted it and lost all progress. In Figure 17 the backup window is depicted.

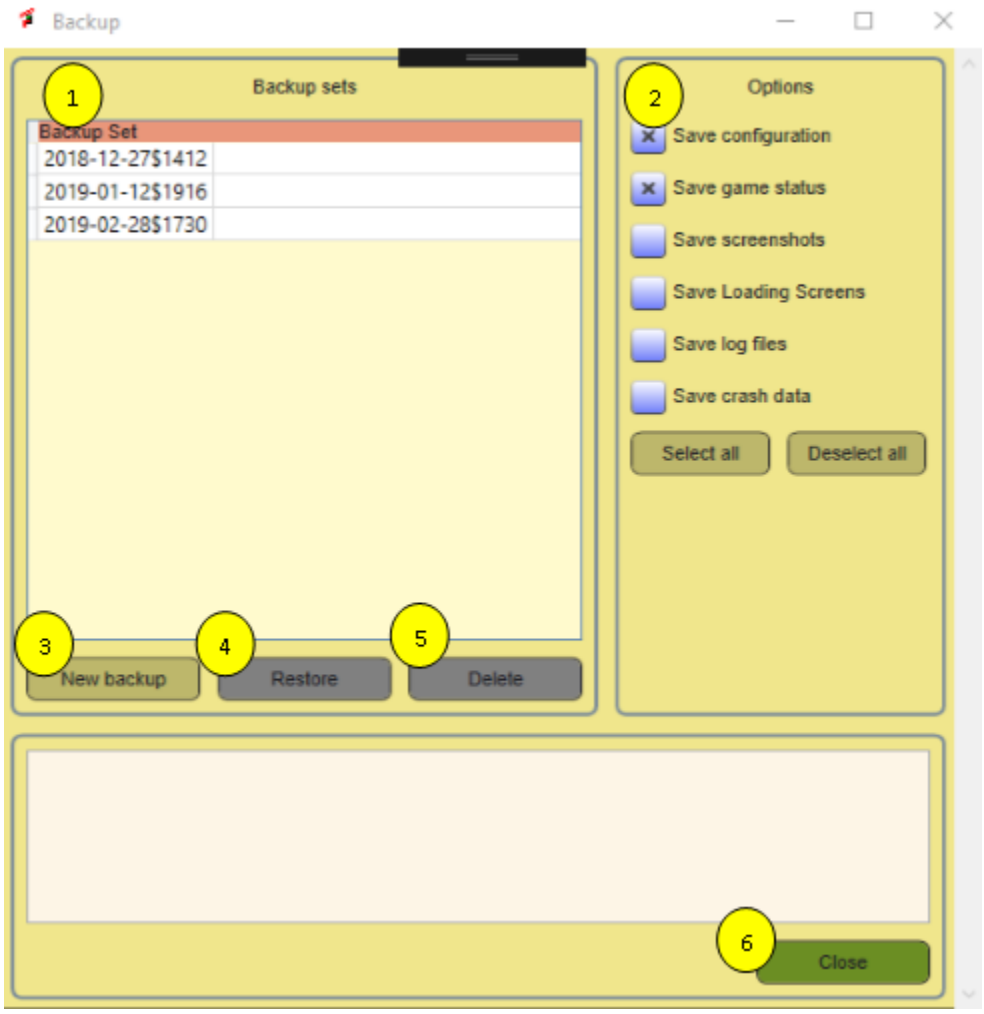
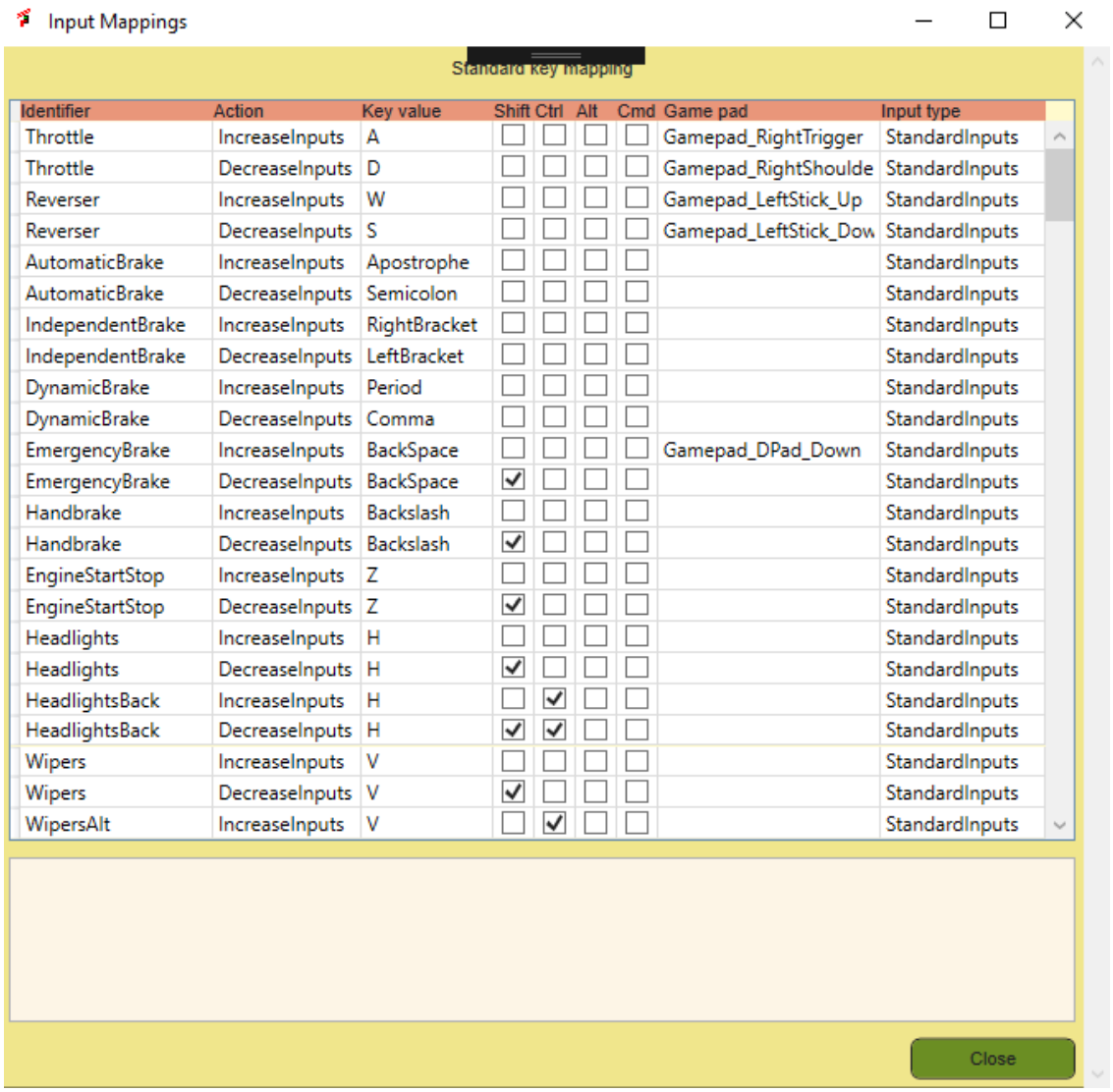


Figure 17 Backup window

1. This list all previous backups. Each backup is stored in a separate folder, named with the date (yyyy-mm-dd) and the a dollar sign and the system tine in four digits. It is likely I will add the option to set a comment text in a next version. So, ask for it if you appreciate that!
2. Options, here you can select what you like to include in the backup. I recommend to use the two checked items as a bare minimum. Optionally you can add screenshots (may consume a lot of disk space!). There a re buttons to select the all or deselect them all.
3. Press this button to create the backup. It will NOT ask for further confirmation!
4. If you select an existing backup, you may restore it. Please be warned, you will NOT be asked to confirm this!!!
5. You also can delete backups to free disk space.
6. This closes the window.

5.4 View key bindings



Identifier	Action	Key value	Shift	Ctrl	Alt	Cmd	Game pad	Input type
Throttle	IncreaseInputs	A	☐	☐	☐	☐	Gamepad_RightTrigger	StandardInputs
Throttle	DecreaseInputs	D	☐	☐	☐	☐	Gamepad_RightShoulde	StandardInputs
Reverser	IncreaseInputs	W	☐	☐	☐	☐	Gamepad_LeftStick_Up	StandardInputs
Reverser	DecreaseInputs	S	☐	☐	☐	☐	Gamepad_LeftStick_Dow	StandardInputs
AutomaticBrake	IncreaseInputs	Apostrophe	☐	☐	☐	☐		StandardInputs
AutomaticBrake	DecreaseInputs	Semicolon	☐	☐	☐	☐		StandardInputs
IndependentBrake	IncreaseInputs	RightBracket	☐	☐	☐	☐		StandardInputs
IndependentBrake	DecreaseInputs	LeftBracket	☐	☐	☐	☐		StandardInputs
DynamicBrake	IncreaseInputs	Period	☐	☐	☐	☐		StandardInputs
DynamicBrake	DecreaseInputs	Comma	☐	☐	☐	☐		StandardInputs
EmergencyBrake	IncreaseInputs	BackSpace	☐	☐	☐	☐	Gamepad_DPad_Down	StandardInputs
EmergencyBrake	DecreaseInputs	BackSpace	☑	☐	☐	☐		StandardInputs
Handbrake	IncreaseInputs	Backslash	☐	☐	☐	☐		StandardInputs
Handbrake	DecreaseInputs	Backslash	☑	☐	☐	☐		StandardInputs
EngineStartStop	IncreaseInputs	Z	☐	☐	☐	☐		StandardInputs
EngineStartStop	DecreaseInputs	Z	☑	☐	☐	☐		StandardInputs
Headlights	IncreaseInputs	H	☐	☐	☐	☐		StandardInputs
Headlights	DecreaseInputs	H	☑	☐	☐	☐		StandardInputs
HeadlightsBack	IncreaseInputs	H	☐	☑	☐	☐		StandardInputs
HeadlightsBack	DecreaseInputs	H	☑	☑	☐	☐		StandardInputs
Wipers	IncreaseInputs	V	☐	☐	☐	☐		StandardInputs
Wipers	DecreaseInputs	V	☑	☐	☐	☐		StandardInputs
WipersAlt	IncreaseInputs	V	☐	☑	☐	☐		StandardInputs

Figure 18 Key bindings or input mappings

This is not a tool doing much, but it gives you a nice overview of most input mappings. Anyway, it helped me to discover some undocumented features. The mappings are sorted alphabetically. In order to use this function, you need to unpack the game files first.

You can change the sort order by clicking at the column headers. It does NOT reflect the custom changes to keyboard mapping. I do not yet have any clue where this information is stored, probably in the status data, which I cannot open.

6 Unpack and view

6.1 Unpack game files

This function will unpack the game files. It cycles through all .pak files it can find and uses the unreal unpacker to do the job. It may take a lot of time, during which TSWTools appear to hang. Please have a lot of patience.

For this function to work, you need to install the Unreal Engine.

1. Here you have a list of all installed active .pak files. As you may notice, the first one is a livery I have installed now. The game core .pak file is NOT shown here.
2. Because unpacking is time consuming, it is done in a separate process. Every second this process reports is it is still running back to the window. If this square is green and shows the word “ready” it is waiting for an unpack order. During unpacking it will have an orange background and the text “busy”. I would like a more fancy animation, but that is not yet working properly.
3. Click this button to unpack the game core.
4. Click this button to unpack all DLC, **including the game core**.
5. In 1 you can select one or more .pak files. This will enable this button, which does what it says.

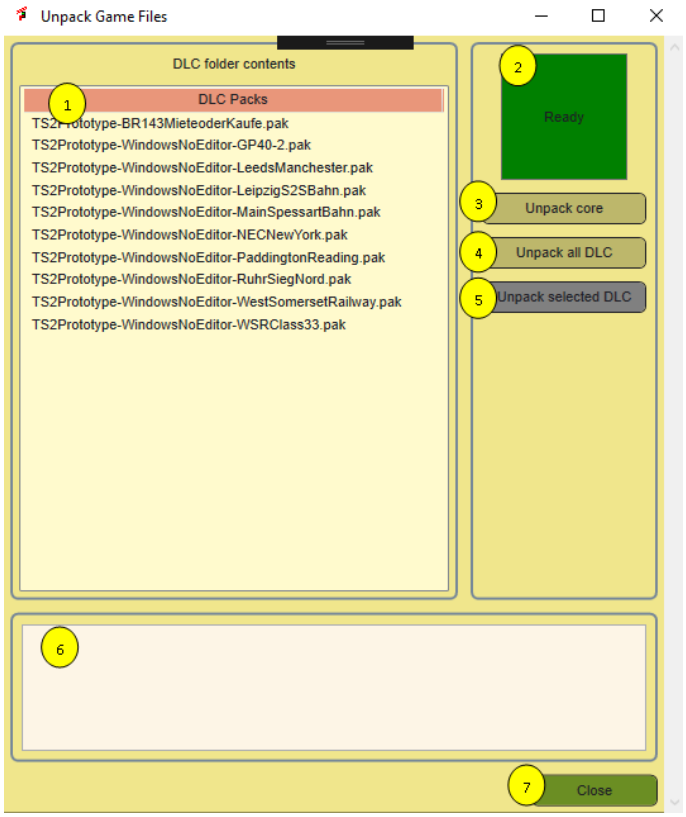


Figure 19 Unpack tool window

6. Fore the moment you can see here a number that is updated every second, so you have some idea that it is still running.
7. Closes the window.

The unpacked files will be stored in the TSWTools folder. See section 2.2 for details.

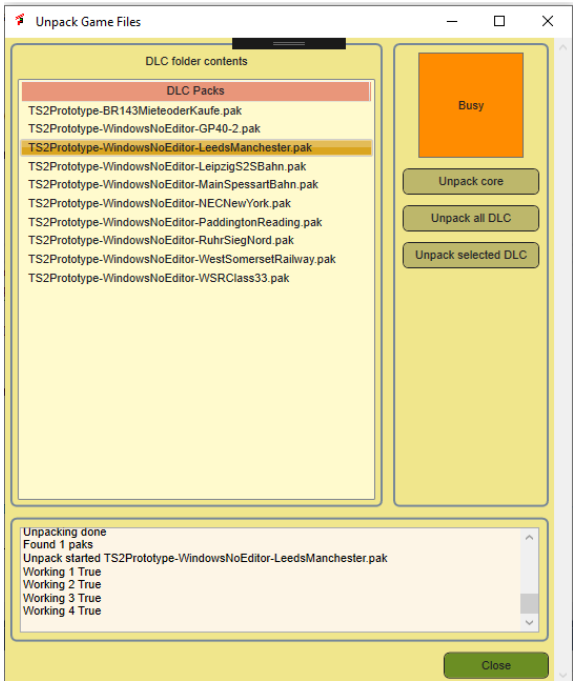


Figure 20 Unpacker while working

6.2 View unpacked files

This opens windows explorer, so you can browse through the unpacked files.

6.3 Launch UModel

UModel is a reverse engineering toolkit for Unreal games. It has it's own user interface, but I think my solution makes it a bit easier to work with the tool.

Note: this function is experimental. Please give me feedback. Until now I have not been able to do anything useful with it.

1. Here you find a list with UModel options that seem most relevant to me, for easier reference.
2. This is a view on thee unpacked assets. You must select a .uasset type file here (likely) and press the Add files button.
3. Here you will find the output of UModel. This textbox will be much wider during execution. I still need to find a better style for this user interface. Coming in the next version…
4. Once you selected an option, this button will insert it at the command line.
5. This line contains the input path you selected in the Files block. You can edit it manually if you like.

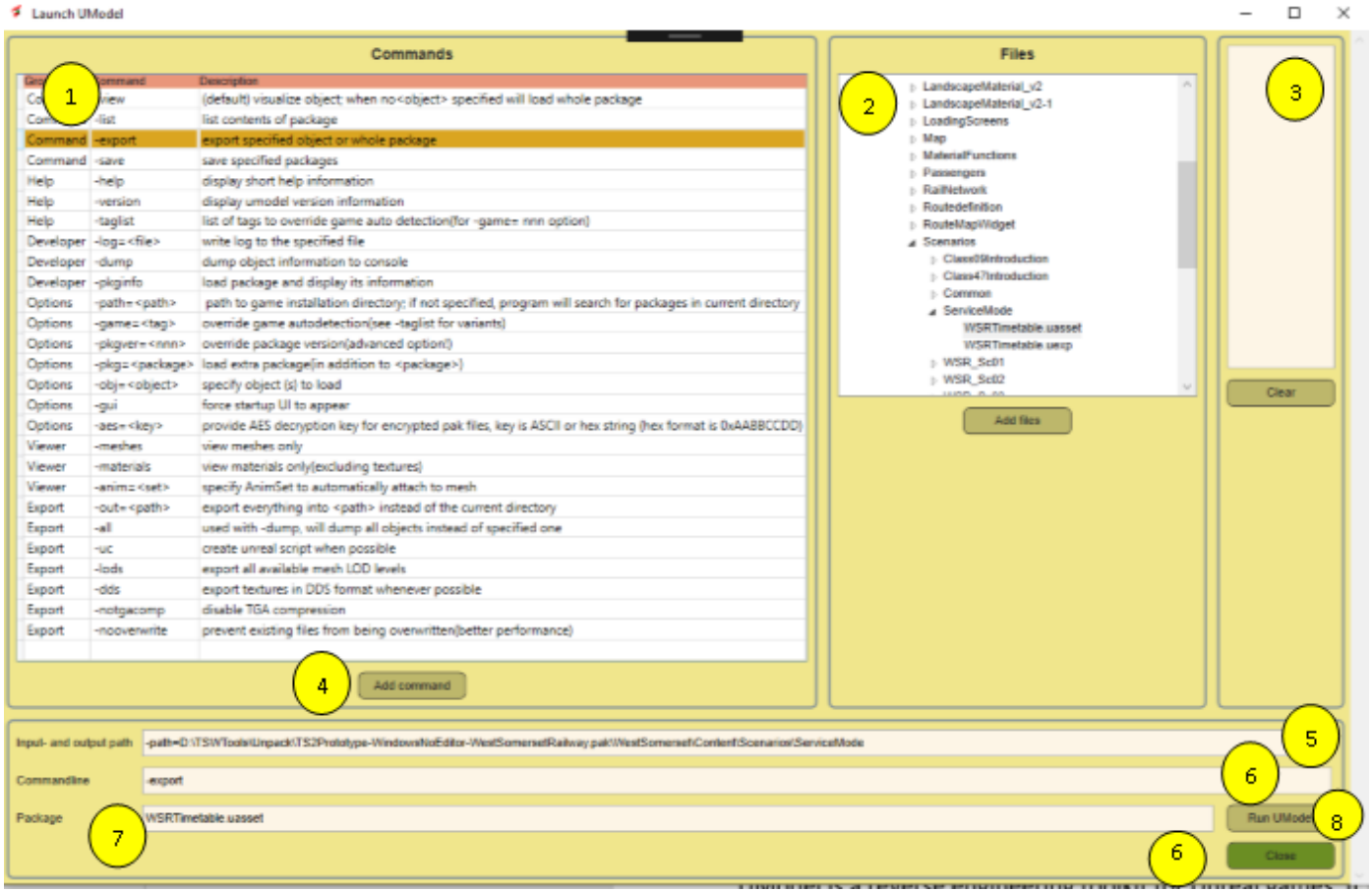


Figure 21 UModel interface for viewing the contents of uasset files

6. Here all options you added will be placed. You can (and sometimes must) edit this manually.
7. TSWTools will split the path you entered in the Files block and place the filename here. UModel interprets this as the "package".

The UModel application is called as follows: Input and output path + commandline+ package.

The -out parameter is always set to the path in the TSWTools folder:

```
<TSWToolsFolder>\Unpack\UnpackedAssets
```

Give it a try and let me know please if you have success or if you encounter errors. For your convenience, the UModel help file is included in appendix B of this manual.

7 Tools

7.1 Game launcher

TSWTools now has a game launcher. It is useful for two reasons:

1. You can load an options set before launching the game
2. You can turn on Sandpatch Radio
3. You can select liveries, using the liveries manager

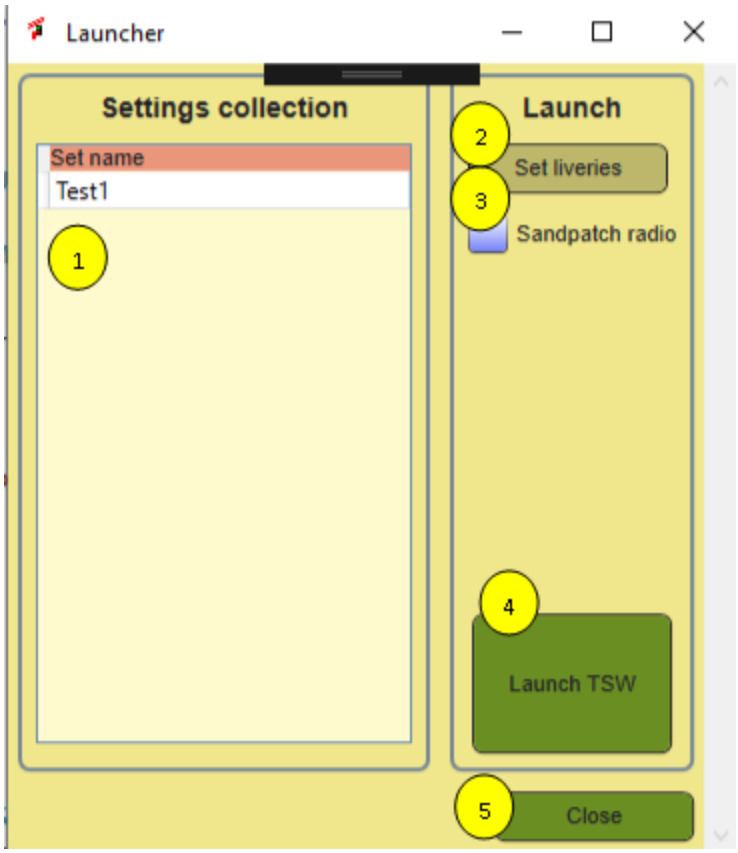


Figure 22 TSW Launcher

It would be nice if we could set some more options, e.g. choose a route or avatar during startup… If you discover any of such functionality, let me know.

Functions in more detail:

1. Select one of the previously prepared settings files. These are stored in the folder **{TSWTools}\OptionsSets**
2. **Set liveries,** this will open the Livery manager. In future I will provide a better integration to make it easier to use.
3. Launch Sandpatch Radio. It will open a browser screen, unfortunately not minimized. I did not bother to support the other players. Let me know if you think there is a good reason to support them. It is hard coded in this version. In a future version I will support more rail radio stations and make it more flexible.
4. The launch button. It will start Sandpatch radio, load the options set if selected and start TSW.
5. Closes this window.

7.2 Edit TSW Settings

7.2.1 Introduction

TSW has a lot of different setting you can adjust in game. This has a number of restrictions:

1. The settings are not always well organized
2. Especially for sound the range for adjustment is too limited. People complain about low sound volumes
3. Some interesting settings are not directly accessible, though you can edit the engine.ini file.

4. It may be useful to have more than one settings set, e.g. one using imperial units and one for metric, a low resolution and a more high resolution set depending on route and how powerful your computer is.

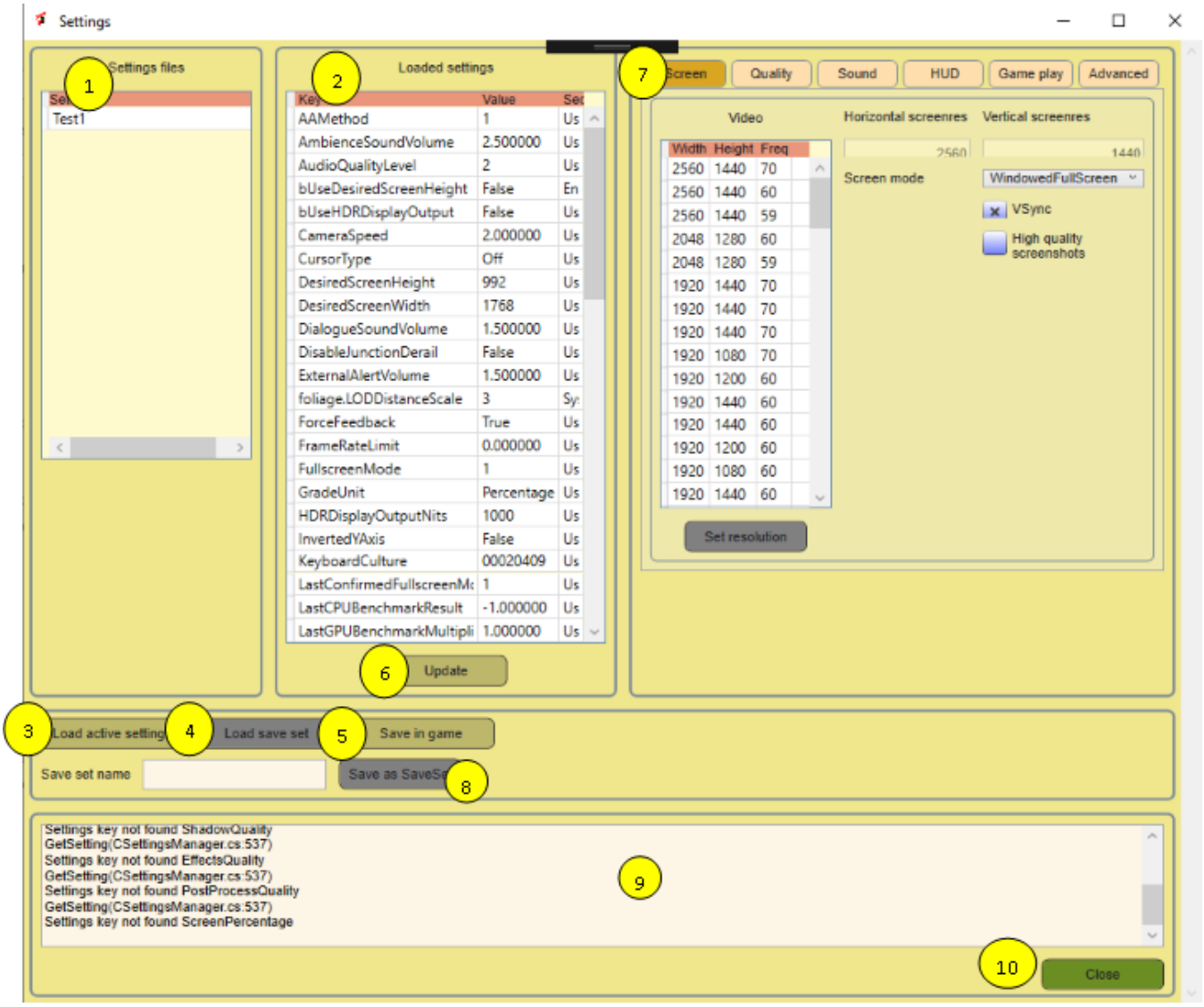


Figure 23 Settings editor

TSWTools solves this problem for you. It allows to edit most settings, including a number of Unreal settings not covered in game. It allows to save the settings files in the TSWTools folder and load sets in the game during launch. The window is a bit complicated, so please read the instructions carefully. It has a main window and a number of tabs for different types of settings.

The common functions are:

1. A list of all available settings collections (technically each collection is stored in a separate folder).
2. The actually loaded collection of settings in the editor.
3. With this button you can load the settings that are now actually set in game
4. With this button you van load a saved set, you must first select such a set in 1.
5. Saves the edited setting as active game settings

6. Updates the loaded settings from what you changed in the tabs (nr 7)
7. Here you see a number of tabs, each of them will reveal a subset of the editable collection. This will be covered in more detail later.
8. Saves the set you are editing now as a saved set. You must provide a **save set name** in the textbox to enable the button. I require you provide at least three characters in the name.
9. Provides informative messages
10. Closes the settings editor. Note that it will NOT warn you for unsaved changes.

In game, the settings are stored in two different files:

The TSW specific settings reside in **GameUserSettings.ini**. The Unreal settings are stored in **Engine.ini** .

TSW has a bit of strange behaviour. Each setting has a default value. If the default value applies, the setting is not defined in the .ini file. You do not need to worry about this. The Settings function in TSWTools can handle this and knows the default value, but it will always create an entry for the setting. Also settings are organised in groups. TSWTools knows about these groups and will make sure this is working.

If DTG decides to add a new setting, this should not be a problem. You cannot edit it, but its value will be preserved, due to the way this functionality is managed.

In the next sections, all supported settings are explained where necessary. For detailed information on what each setting will do, please read the TSW Starters Guide, which is included in the TSWTools installation file. I do not want to write this text twice.

7.2.2 Screen settings

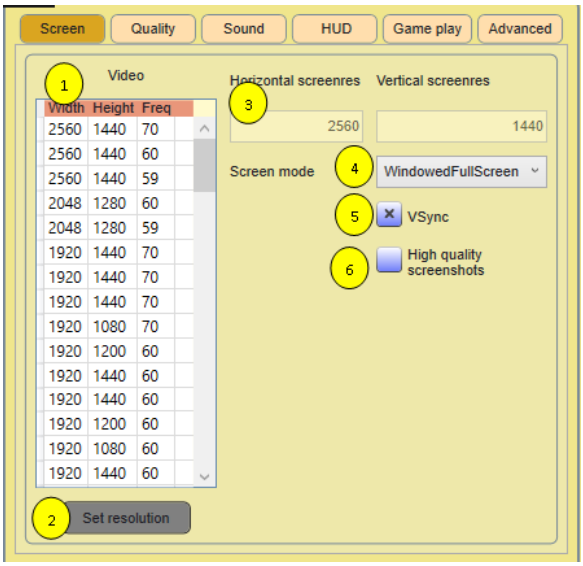


Figure 24 Screen settings

1. Here you get a list of all screen resolutions your screen supports. The frequency is available here, but it is not used as a game setting. By clicking at the column header, you can have some influence on the sort order.
2. Once selected a resolution, click here to activate it.
3. These two boxes show the set screen resolution.
4. Here you can select the screen mode. It is a combo, so no mistakes possible.

5. Turns vSync on or off
6. Turns high quality screenshots on or off. Warning: high resolution is very high resolution and consumes a lot of disk space.

7.2.3 Quality settings

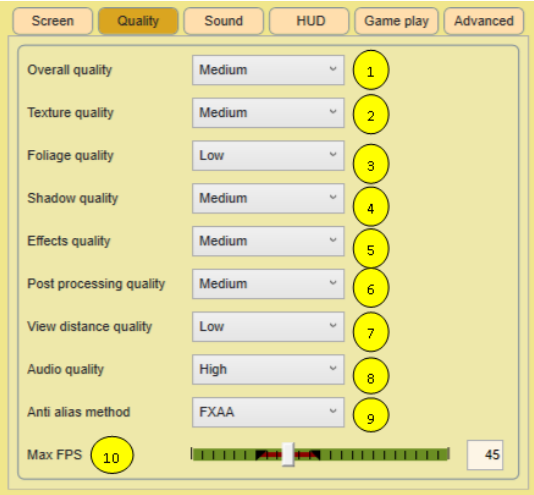


Figure 25 Quality settings

These settings affect the graphics quality. In the **GameUserSettings.ini** file this is not always done consistently. You can select the values Ultra, High, medium and Low for each of them\For the anti-alias method, the technical terms are used. Matt (our DTG expert) recommends FXAA for most cases.

You can limit the number of frames per second. Below 30fps is not a good idea, and 60fps should be enough for high end systems. If you set it to 0, no limit is set.

7.2.4 Sound settings

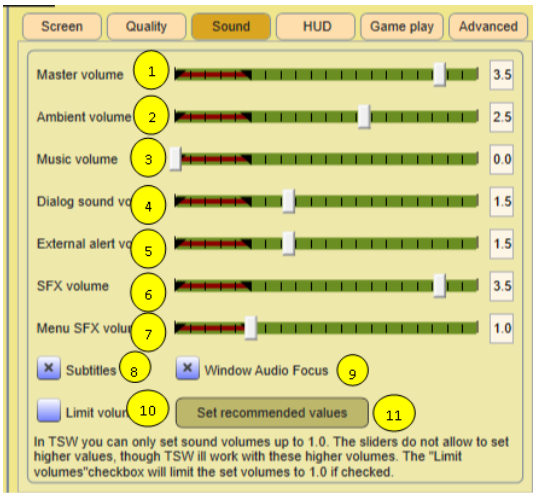


Figure 26 Sound settings

TSW supports sound settings in the range 0.0 to 1.0 For many users this is not loud enough and higher values are recommended by some users. **You can set these higher values here, but you do it at your own risk.**

In brown you see the officially supported range. If you select a higher value, in the TSW settings menu you can only change it back to the supported range.

- At 8 you turn subtitles on or off.
- At 9 you can set windows audio focus, whatever that means.
- At 10 you can limit the settable sound volumes to the supported volumes.
- At 11 you can set the values to what is considered a good practice.

7.2.5 HUD settings

Figure 27 HUD settings

In TSW you have a lot of control on what you want to see at your screen and how you want to see that. See the TSW Starters Guide for all details. In TSW Tools you can apply the initial settings in a more structured way.

1. The objective marker shows the next task and the distance to the next task. You can select either the none, HUD version or a marker or both of them. In game there is an easy toggle function for this.
2. Same as above, for the next signal.
3. You can show or hide the next signal aspect also.
4. Same for the next seed limit, but here there is not an option to hide its value.
5. TSW has a scoring system, you can show the actual score or hide it.
6. TSW has markers in the 3D world where you can start a scenario or tutorial. This setting will hide or show these markers.
7. You can hide or show the compass separately.

A few newer settings are not settable. There is a stop area marker you cannot set and there is a reticule dot you cannot set here. If anyone knows how to do that, please let me know.

7.2.6 Game play settings

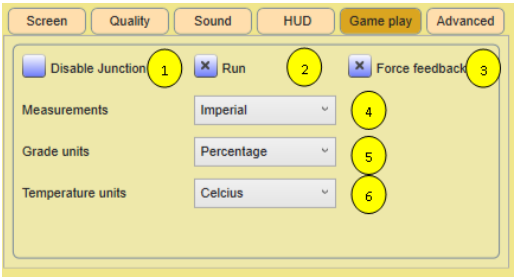


Figure 28 Game play settings

There a few gameplay settings, which are settable here.

1. Set disable junction derail on or off.
2. Set first person mode default to walk or to run
3. Turn force feedback on or off
4. You can select imperial or metric units (Mph or km/h)
5. You can select grade units
6. You can select temperature units

7.2.7 Advanced settings

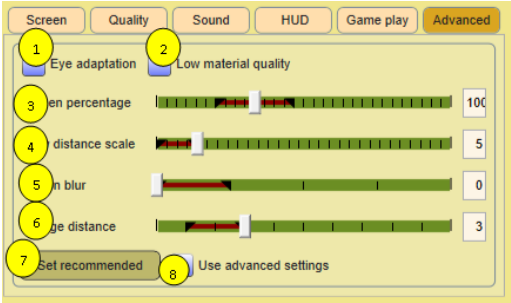


Figure 29 Advanced settings

The advanced settings are all settings supported by the Unreal engine. See the TSW Starters guide for detailed information.

For your convenience, at 7 you can select a recommended setting for all parameters. If you deselect 8, the advanced settings will not be used, except for Screen percentage, which will be set to 100%

For Motion Blur (5) TSW provides a key combi to turn it on or off. Here you can turn it off or select values in the range 1-4 to determine the amount of motion blur.

7.3 Livery manager

There is not yet a content editor for TSW available, but despite that several community members managed to create reskins or small content improvements. Installing the improvements is simple, just place the .pak file in the DLC folder. This approach has one disadvantage, you cannot supply more than one livery at a time. So, if you want to exchange a livery, you must uninstall the old one. TSWTools offers a Livery Manager, that allows you to manage liveries and install or uninstall them with a few clicks. It also maintains a catalogue with all livery packs you have available. For the moment, you need to install liveries in the livery manager folder (see section 2.2 for details) and you can eventually add some useful metadata.

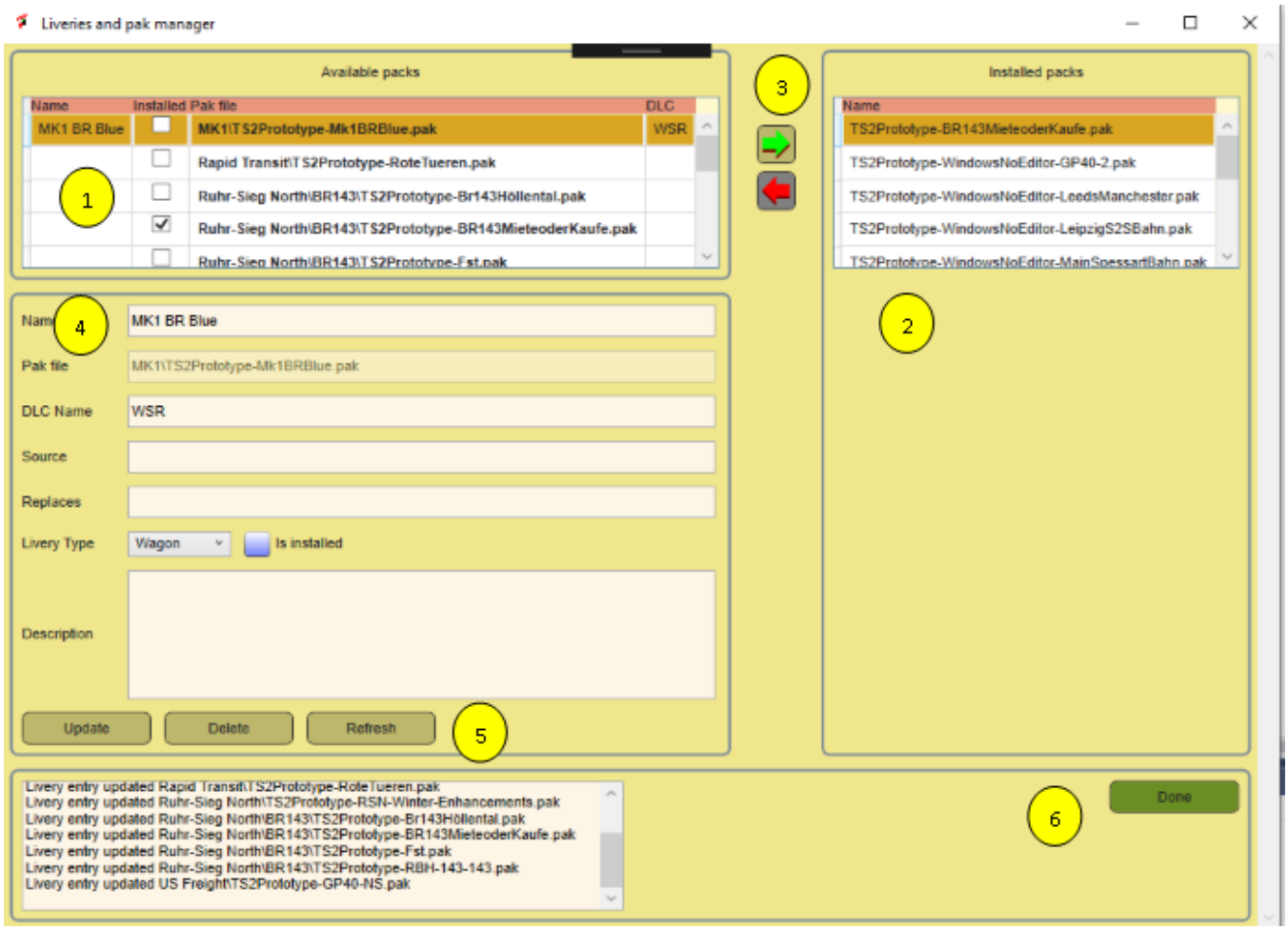


Figure 30 Livery manager

1. A list of all available liveries. This list is stored in the TSWTools database.
2. A list of installed paks, this includes the DLC you purchased. In principle you can treat them as a livery as well, but it does not affect steam, so maybe steam will try to download the DLC again. I have not tested for this.
3. The green arrow button activates installation. The red button activates uninstall function.
4. A set of metadata for your convenience.
5. Update: updates the database, Delete: deletes a livery from disk and database, Refresh, reads the database again for synchronization.
6. Closes the tool

Note: to unstall a pak select a pak with the “Installed”checkbox set. This will activate the uninstall button (red arrow). This may feel not very intuitive. I will see if I can create a better and more natural solution.

7.4 View screenshots

TSW has essentially two screenshot methods:

- You can use the F12 key to get a screenshot including the HUD. Essentially this is the steam screenshot facility.
- You can use Ctrl+F12 to get a TSW screenshot, without any HUD stuff.

Unfortunately the two types of screenshots are stored at different locations and have different size and file format. Therefore TSWTools includes a screenshot viewer that brings them all together. You also can delete screenshots here or make a local copy if you like.

Figure 31 Screenshot viewer

1. Thumbnails of the screenshots
2. Load previous page
3. Load next page
4. Number of available screenshots
5. Larger preview of screenshot. You can do this by clicking at a thumnail.
6. File path of the previewed screenshot
7. Opens a file dialog and copies the selected screenshot to the desired location, while updating the file name.

8. Deletes the selected screenshot. Warning: it does NOT ask for confirmation!

Note: this screen needs some redesign of smaller monitors. This will come in the next version. I also noticed that if you synchronize the Images folder using OneDrive this does not perform very well, so you should avoid that probably. I will see if I can do the loading in a background process.

In the next version I will see if it is possible to use your screenshots as loading screen.

8 Help

8.1 About dialog

The About Dialog informs you about the actual version of TSWTools. It also provides a link to the website where you can download updates and additional tools.

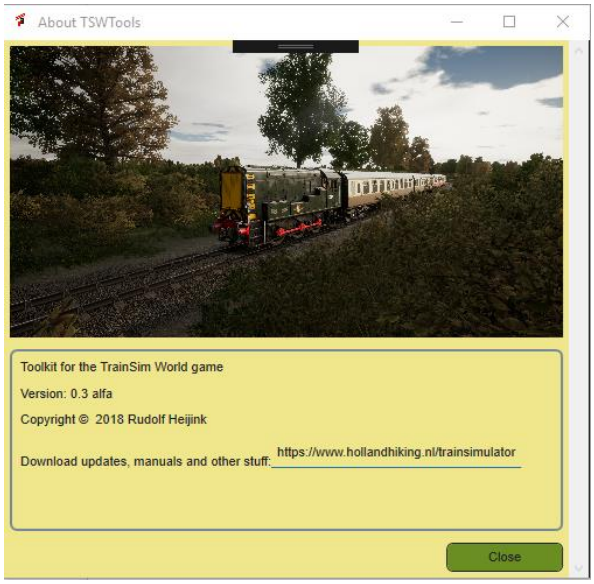


Figure 32 About Dialog

8.2 Open TSWTools manual

Clicking the button should open this manual, which is included in the installer. In case you want to install updates, make sure to select the appropriate folder. See section 2.2 for more information.

8.3 Open TSW Starters Guide

The TSW Starters guide provides a lot of useful additional game information. A versio will be included in the installer. In case you want to install updates, make sure to select the appropriate folder. See section 2.2 for more information.

8.4 Open route guides

In this folder you can install the official game manuals and any additional material as you see fit. Likely, for some routes I will provide additional route guides, which you can download from my website. In case you want to install additional files, make sure to select the appropriate folder. See section 2.2 for more information.

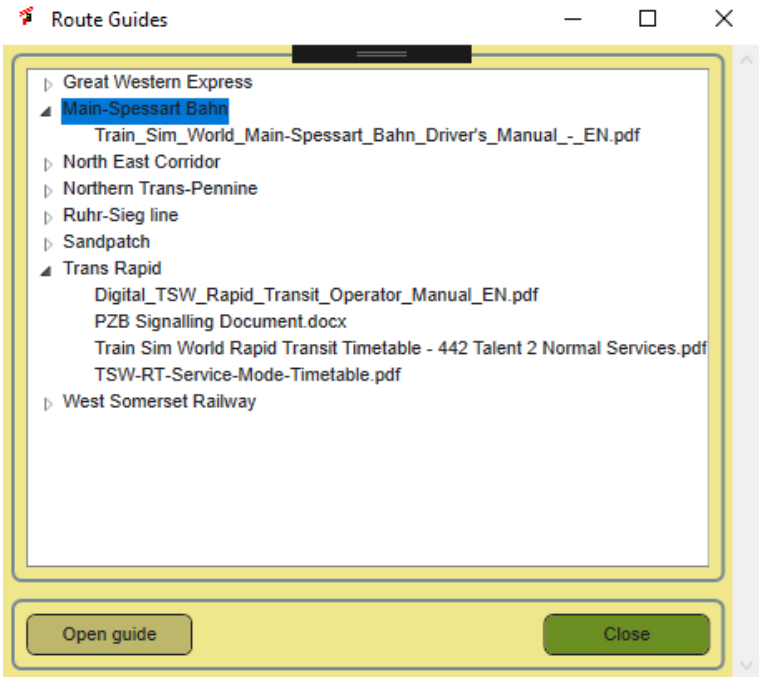


Figure 33 Route guides selector

If you click the button, a simple dialog opens. It represents the folder/file tree for the Manuals/RouteGuides folder. You can select either a directory or a specific document and click the button to open it. TSWTools sues Shell Execute, so it works for all known file types and you are completely free to build your own directory tree.

A. Download locations

All my guides and tools are available here:

Site name	URL's
Holland Hiking (download page for all tools and guides)	http://www.hollandhiking.nl/trainsimulator/

At this site you also will find links to all other tools and manuals mentioned in my guides.

B. UModel command reference

Unreal Engine viewer and exporter

Link to website:

http://www.gildor.org/

```
Usage: umodel [command] [options] <package> [<object> [<class>]]
       umodel [command] [options] <directory>

    <package>       name of package to load - this could be a file name
                    with or without extension, or wildcard
    <object>        name of object to load
    <class>         class of object to load (useful, when trying to load
                    object with ambiguous name)
    <directory>     path to the game (see -path option)

Commands:
    -view           (default) visualize object; when no <object> specified
                    will load whole package
    -list           list contents of package
    -export         export specified object or whole package
    -save           save specified packages

Help information:
    -help           display this help page
    -version        display umodel version information
    -taglist        list of tags to override game autodetection (for -game=nnn
option)
    -gamelist       list of supported games

Developer commands:
    -log=file       write log to the specified file
    -dump           dump object information to console
    -pkginfo        load package and display its information

Options:
    -path=PATH      path to game installation directory; if not specified,
                    program will search for packages in current directory
    -game=tag       override game autodetection (see -taglist for variants)
    -pkgver=nnn     override package version (advanced option!)
    -pkg=package    load extra package (in addition to <package>)
    -obj=object     specify object(s) to load
    -gui            force startup UI to appear
    -aes=key        provide AES decryption key for encrypted pak files,
                    key is ASCII or hex string (hex format is 0xAABBCCDD)

Compatibility options:
    -nomesh         disable loading of SkeletalMesh classes in a case of
                    unsupported data format
```

```
    -noanim         disable loading of MeshAnimation classes
    -nostat         disable loading of StaticMesh class
    -notex          disable loading of Material classes
    -nolightmap     disable loading of Lightmap textures
    -sounds         allow export of sounds
    -3rdparty       allow 3rd party asset export (ScaleForm, FaceFX)
    -lzo|lzx|zlib   force compression method for fully-compressed packages

Platform selection:
    -ps3            Playstation 3
    -ps4            Playstation 4
    -nsw            Nintendo Switch
    -ios            iOS (iPhone/iPad)
    -android        Android

Viewer options:
    -meshes         view meshes only
    -materials      view materials only (excluding textures)
    -anim=<set>     specify AnimSet to automatically attach to mesh

Export options:
    -out=PATH       export everything into PATH instead of the current directory
    -all            used with -dump, will dump all objects instead of specified one
    -uncook         use original package name as a base export directory (UE3)
    -groups         use group names instead of class names for directories (UE1-3)
    -uc             create unreal script when possible
    -psk            use ActorX format for meshes (default)
    -md5            use md5mesh/md5anim format for skeletal mesh
    -gltf           use glTF 2.0 format for mesh
    -lods           export all available mesh LOD levels
    -dds            export textures in DDS format whenever possible
    -notgacomp      disable TGA compression
    -nooverwrite    prevent existing files from being overwritten (better
                    performance)

Supported resources for export:
    SkeletalMesh    exported as ActorX psk file, MD5Mesh or glTF
    MeshAnimation   exported as ActorX psa file or MD5Anim
    VertMesh        exported as Unreal 3d file
    StaticMesh      exported as psk file with no skeleton (pskx) or glTF
    Texture         exported in tga or dds format
    Sounds          file extension depends on object contents
    ScaleForm       gfx
    FaceFX          fxa
    Sound           exported "as is"

For list of supported games please use -gamelist option.

For details and updates please visit http://www.gildor.org/en/projects/umodel
```

C. Known issues

There are some issues I am aware of, but not yet solved. These are listed here.

Issue nr	Description	Priority
1	On small screens the user interface is not always looking good	Medium
2	The UModel interface is still experimental	Medium
3	The index for this manual should be updated	Low
4	In combination with OneDrive loading screenshots performs poor	Medium
5	Thumbs are not properly updated when you delete a screenshot	Medium

Index