Alex95 & Afirus KI Pack

This pack contains an extensive amount of different cargo types. Including:

7.82 swap bodies 7.82 tanktainers/swap bodies 30ft tanktainers/containers 40ft containers/silos and high cube's 45ft containers and high cube containers 20ft containers/tanktainers

All these containers are accurate, even the small height variations for nonstandard containers (mostly 782/715 or 30ft containers). This will give KI trains a lot more of a realistic look. For example, the 40ft silo contains almost 10 different variations that match real-life. Most users will probably not even notice it, but we wanted to go the extra mile ;).

The cargos are generated randomly but stay the same once you place them. You can use the rail vehicle number to select a container type by changing the number(s) after the "_". This way the cargo will always be the same every time you run the scenario. So, you can now finally recreate almost every specific European train consist!

Installation

To install the pack, simply install the .rwp with Rail Utilities or extract the .rwp using 7zip and copy the files to the Assets folder.

Usage

Activate the provider and products like this:

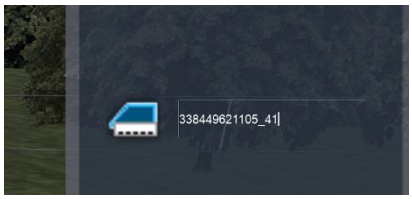
You can limit the cargo types you want to load by only selecting the ones you are going to use. The cargo types have been split because the cargo's have the largest hit on performance. Even though the package is only a few 100MB, this will help in the future when we add more cargo packs.

Make sure "KI Wagon Essentials" is always selected if you are going to use any wagon from the pack.

The next step is to place a wagon:

As you can see, everytime you place a wagon, a random cargo type is choosen. Although the cargo is randomly chosen at the beginning. It will not change when you reload the editor or run the scenario. Lets now look at how you can change the cargo type manually:

In this image you can see that the wagon number has a rail vehicle number ending with "_41". Number 41 determines the type of cargo. At the end of this document, you can find a full catalog of all cargo numbers per cargo type.



If a wagon has multiple cargo slots, you can add multiple: "_", for every cargo:

And that's it. Now you can save the scenario. Play it and you will see the same cargos every time.

Performance

A lot of time has gone into improving performance, ensuring that a lot of wagons from this pack can be placed without creating memory leaks, creating overhead on the GPU, CPU, or RAM. Wherever possible objects have been merged into one to improve draw calls. Textures, as well as geo files are reused where possible, and finally, the script used to select random skins has been tweaked and tested for more than a year and contains a lot of tricks to prevent lag spikes, and bugs/inconsistencies in Railworks itself.

Considerations

To optimize performance, we decided to put as many containers as possible in one texture. This allows users to place the most objects without crashing the game. Unfortunately, this makes it so that you can't place individual cargos containers. In version 2 or 3, the ability to add cargos as scenery objects will be included.

Assumptions

The script, blueprint and geometry files are optimized based on the following (optimistic) assumptions I have made.

1. The game is relatively optimized for atlas textures and reusing/caching resources
2. Textures are only loaded when needed
3. All Bin files in a Provider/Product (and possibly geo files) are loaded when a scenario starts (BAD)
4. Reskin blueprints are optimized

Requirements

1. At least an 8GB VRAM GPU
2. Fast RAM, DDR4 or better

Tips for scenarios to get best performance/memory usage

Limit variation use

To get the best performance, try to use the least number of unique vehicles (blueprints). Try to use the same vehicle and change the cargo using the rail vehicle number. Each variation loads its own textures and has their own set of materials (which slows down the game).

Use the static variants

Use the static variants for non-moving wagons, which do not require LUA scripts to work, which should improve performance a lot. It also minimizes the number of unique objects in the scenario, increasing fps.

Avoid the more complex cargo variants

Unfortunately, the way to get the random skins to work was to have duplicates of them. After the initial load, the other variants are removed via LUA scripting. Having lots of these variants could lead to large lag spikes when loading the wagons. You will have noticed this already when TS loads a tile when driving in a scenario. The large LUA initialization spikes are already mitigated as much as possible because our script chooses a random delay for each repaint do distribute the load across multiple frames instead of large spikes on initial load.

So, try to limit the cargos with complex geometry (tanktainers, silos, etc.). If you are running into performance issues or crashes, these are the first ones you want to remove.

Adding cargos to other wagons

Adding cargos to other wagons can be done simply, only by editing the .bin files. You can the Sggmrss files as an example. Make sure you change the following things:

1. Update the autonumbering list (create a new dcsv file with a list of numbers for the container combinations you want for your wagon).
2. Make sure you use an empty version of the third-party wagon you are adding the cargos to.
3. Update the Mass so it includes the cargo weight
4. Update the Children and add a child for each cargo slot. They must follow the following naming convention: Cargo_A, Cargo_B, Cargo_C, Cargo_D. Most of the time you will only use 1 cargo slot, so use Cargo_A. Again, the Sggmrss wagons (CH-WASCO) serve as an example.
5. Update the script component to point to: AlexAfirus\KI Sggmrss90\wagon_script_rnd
    a. (optional) Copy the script to the same folder as the .bin file so the user doesn't need to activate AlexAfirus\KI Sggmrss provider/product.

Notes to repainters

When adding a new blueprint for a reskin. Make sure it does not conflict with our naming convention. So do not add a 40ft_3.xml repaint to an already existing 40ft_1.xml & 40ft_2.xml as it may create a conflict when we add new repaints ourselves in the future. Preferably use something like 40ft_[a name for my reskin in lowercase].xml or put it in a separate folder (even better).

Make sure to use "Reskin Blueprints", we do so for our own repaints as well. This means there is only one geopcdx file. It also makes it so that we can change the original CH-WASCO model in the future and your reskin will automatically use the new GeoPcdx file.

Cargo repainters

The cargo's use Atlas textures. These are large textures that hold multiple repaints. Each wagon randomly chooses one randomly. The cargo it chooses is derived from the rail vehicle number. A rail vehicle number is automatically selected by Railworks when you place a wagon in a scenario. Railworks chooses this number based on a list that is provided in the Blueprint under "RailVehicleComponent" -> "Numbering List". If you look at a list, you will notice that the numbers and in. _2 or _13 (or for multiple cargo per wagon: _2_1 or _13_26). These numbers correspond to the liveries on the atlas texture. The first number starts at the bottom left of the texture, Number 2 is the one to the right of the first one and then it counts bottom left to top right.

Note that most cargos contain small variations, the 30ft containers for example differ significantly in height between the smallest and largest containers. Check the original blender file and note down the number of the variant you want to make the repaint for ("cargo_3", or "cargo_20"). Now use the same trick as explained above by counting from bottom left to top right. to get the repaint location for that specific object in the atlas texture.

It might be that you are not able to fill the entire Atlas of textures, this is not an issue. You can tweak the number list to exclude the numbers of repaints you were not able to fill.

You can also add the same combination multiple times to the numbering list to increase the chance of a specific repaint spawning.

These techniques are all used by us so you can just check out the existing blueprints and numbering lists as an example.

Wagon repaints

The CH-WASCO model (at least for the Sggmrss) serves as a base model. The other wagons use "Reskin Blueprints".

Questions

Feel free to ask question, send feedback, or contact me via railworks@esstudio.nl