

*.inc Files...For the Common Man

by Sanjay Das

Updated 4/19/24

Introduction

In both Microsoft Train Simulator (MSTS) and Open Rails Train Simulator (ORTS), *.wag and *.eng files are essential files that allow the simulators to identify an asset as either a locomotive (*.eng or “engine” file) or a piece of rolling stock (*.wag or “wagon” file). These files contain all the necessary information—about physics, sound, cab view files, etc—that are to be used by that locomotive or car.

However, in some ways, the *.eng and *.wag files represent a certain sense of redundancy. In the days of MSTS, all the physics for each individual locomotive had to be contained within its own *.eng file, and likewise every car must have all its physics contained within its own *.wag file. At first, this may not seem like it poses any problem, but depending on the size of your locomotive and rolling stock collection, it can be problematic in some ways.

The first issue is in fleet editing. Let's say you have a fleet of 20 or so nearly-identical freight cars. Each freight car is exactly alike in every way with the exception of only a few minor differences—different paint schemes, road names, road numbers, etc. However, the majority of the contents of each of the 20 *.wag files generally contain identical parameters with congruent values. Thus, if you wanted to change one of these “shared” physics parameters across the entire fleet—for example, increase the braking force,—you come to the realization that you have to edit each *.wag file *individually*, which can become immensely tedious and time-consuming. Additionally, these duplicated parameters among tens of dozens of *.wag and *.eng files can eat up many megabytes of disk space in no time at all.

Thankfully, ORTS now has a better solution—using *.inc files. In this text, you will learn what an *.inc file is, how it works, and how to write your own files.

What is an *.inc file?

An *.inc file is a remote file that can contain common *.eng and *.wag parameters used by multiple locomotives and/or cars. One way of thinking about it is that it is essentially the *.eng and *.wag file physics equivalent of sound and cab view “aliasing.” These files don't even have to be in the same subfolder as the locomotives or rolling stock that use them—the only thing ORTS needs to know is the file path of the relevant *.inc files that each vehicle will use.

How *.inc files work

An *.inc file works by replacing a block of physics code within an *.eng or *.wag file with an “**Include**” statement (example: **Include (commonphysics.inc)**), which states the file path of the *.inc file from which to extract the otherwise-missing data. When ORTS loads a locomotive or car, it reads the beginning of the main *.eng or *.wag file (Fig. 1), until it gets to the “**Include**” statement. It then searches for the *.inc file in the specified file path, after which it reads the data from the *.inc file, and

inserts the *.inc file's data into the locomotive or car's physics data. When ORTS reaches the end of the *.inc file, it then returns to the main *.eng or *.wag file and continues reading its data until it either finds another “Include” statement (at which point the *.inc file reading cycle starts again) or reaches the end of the main *.eng or *.wag file.

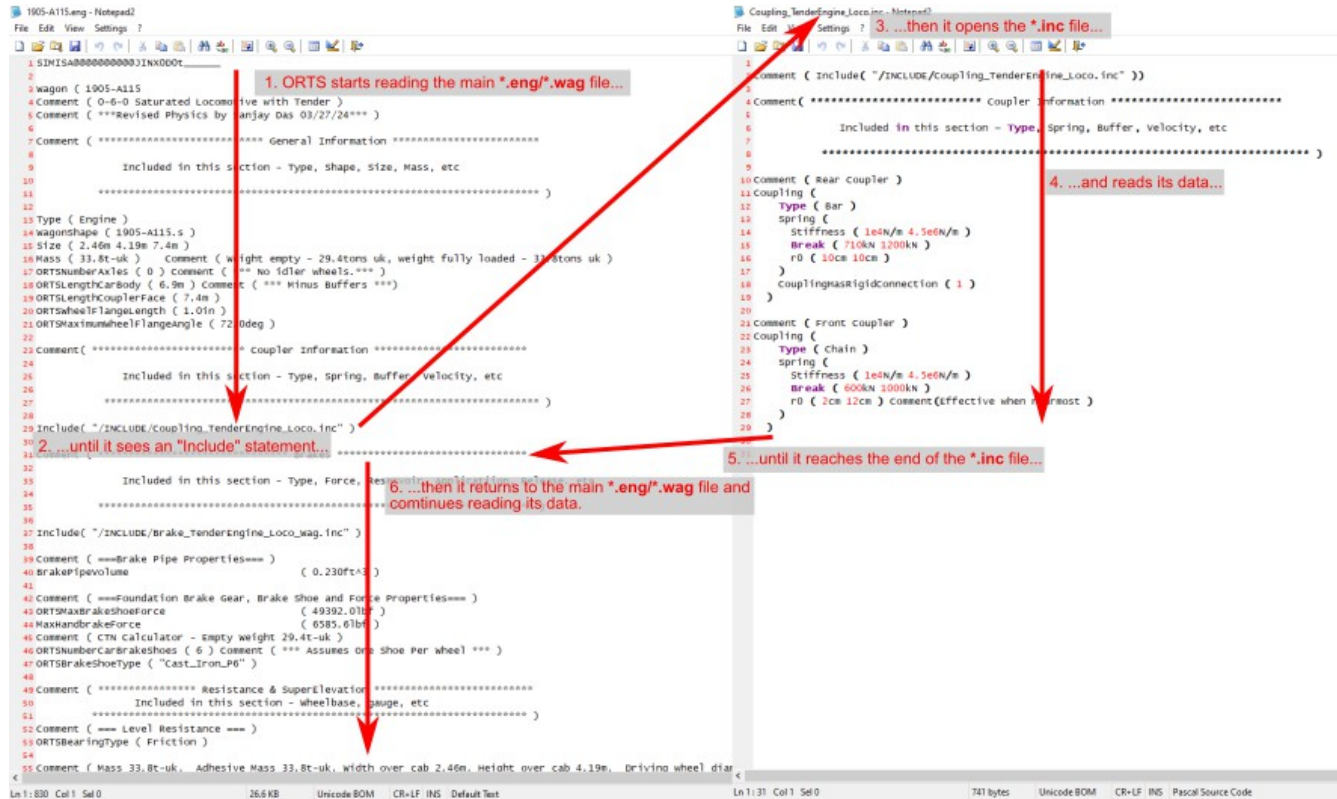


Fig. 1: Flow diagram of how ORTS works with *.eng/*.wag and *.inc files.

It should also be noted that ORTS treats **Include ()** statements in one of two ways:

1. As previously described, if it finds an **Include ()** statement within the main *.eng or *.wag file.
2. Within the *.eng and *.wag files contained within the subfolder “OpenRails”, which usually contain ORTS-exclusive parameters, in order to maintain the main files' compatibility with MSTs, and are tied into the main *.eng or *.wag file with an **Include ()** statement pointing to the main *.eng or *.wag file (example: **Include (“myloco.eng”)**). This method will not be discussed in this text, but more information can be found on the ORTS Online Manual.

How to Create Your Own *.inc files

Creating an *.inc file can be done in many ways, all you need is Windows Explorer and a text editor. Here is one method, which is the way I personally create new *.inc files:

1. In Windows Explorer, go to the location of the *.eng or *.wag file you want to extract data for your *.inc file from, highlight it, then right-click it and from the drop-down menu, left-click “Copy”.

2. Go to the location where you want your *.inc file to be located, right-click any empty white space in the Windows Explorer window, and from the drop-down menu, left-click “Paste.”
3. Highlight the newly-pasted copy of your *.eng or *.wag file, right-click it, and from the drop-down menu, left-click “Rename.”
4. Rename your *.eng or *.wag file copy, changing the filename from either “<filename>.eng” or “<filename>.wag” to “<newfilename>.inc”, and hit **ENTER**.
5. When you hit **ENTER**, Windows will give you the following message: “If you change a file name extension, the file might become unusable. Are you sure you want to change it?” Left-click “Yes” to accept this change.
6. Open up the *.inc file with your favorite text editor, and make any necessary changes (see below).

General *.inc File Guidelines

For the most part, the rules for writing and editing *.eng and *.wag files also apply to *.inc files. However, there are a few additional guidelines that need to be followed in order for the *.inc files to work properly:

- The first line of the *.inc file **MUST be left blank!**
- The “SIMISA@@@@@@@@@JINX0D0t_____” header that ordinarily appears at the top of every *.eng or *.wag file **MUST be omitted!**
- While not mandatory, it helps that an “**Include**” statement with the *.inc file's file path be located somewhere within the *.inc file so it may be easily copied and pasted between *.eng or *.wag files as needed. In the *.inc file, this “**Include**” statement should be placed within “**Comment()**” tags so as to prevent confusing and/or crashing ORTS when it reads the files. I usually write this “**Comment(Include ())**” statement on the second line (top) of my own *.inc files.
- *.inc files are really only meant for “common” parameters that are shared between several locomotives or cars. Parameters that vary between individual locomotives and cars, such as the shape file and name parameters, should not be included in the *.inc file, and should remain within the individual *.eng or *.wag files. I generally only include data in an *.inc file that is shared by at least two vehicles.
- *.inc files are meant for use in ORTS only. Thus, if any **Include ()** statements are present in the body of the main *.eng or *.wag file, this renders the associated files as ORTS-exclusive files. Attempting to open such files in MSTs will cause the game to crash. However, if using an OpenRails subfolder that contains supplementary ORTS-only data (see page 2), this data will be ignored by MSTs.
- ORTS reads the **Include ()** statements in order from top to bottom. When ORTS reads a new *.inc file, any duplicate data that was found among the *.eng , *.wag or any previously-read *.inc files is overwritten by the data from the most recent *.inc file. For example, if you have a **DoesHornTriggerBell** parameter in the main *.eng file set to (0) and another **DoesHornTriggerBell** parameter in an *.inc file set to (1), then depending on the placement of the **Include ()** statement, one parameter may override the other. To prevent this, it is necessary to either eliminate any duplicated parameters among multiple files, or (preferably) place the **Include ()** statements somewhere near the end of the main *.eng or *.wag file.

The same rules for file paths used in cab view and sound aliasing also apply to *.inc file “**Include**” statements. In other words, your *.inc file doesn't have to be located within the same subfolder as the *.eng or *.wag file(s) that use them, provided **the full file path** is given.

If the *.inc file is located within the same subfolder as the *.eng or *.wag file(s) that use it, then the file path would be specified as:

Include (“filename.inc”)

If the *.inc file and the associated *.eng or *.wag file(s) are in different subfolders within the **TRAINSET** folder, then it needs to be specified like this:

Include (“../SUBFOLDERNAME/filename.inc”)

Alternatively, you can also specify a location of a separate subfolder within the subfolder that contains the *.eng or *.wag files:

Include (“/INCLUDE/filename.inc”)

As with cab and sound aliasing, the file paths **are not case-sensitive**, but the file and folder names **must be spelled correctly** in order for the *.inc files to work properly!

A Few Examples of *.inc File Application

To illustrate how *.inc files can be applied to any piece of rolling stock, I will share with you what I do whenever I retrofit any locomotive or car I download to incorporate *.inc files.:

First, I will go to that locomotive or car's subfolder within the **TRAINSET** folder and create a new subfolder labeled “**Include**”. I will then copy and paste copies of one of the *.eng or *.wag files from which I want to extract data from (see pages 2-3), and make the necessary modifications.

A typical passenger or freight car might thus contain the following *.inc files:

*.INC FILE NAME	CONTENTS
/Include/GeneralData.inc	Vehicle type, size, derail coefficients , external *.sms file to be used. The shape file and name parameters are usually omitted, but in the case of a freight car with separate empty and loaded physics, the shape file may still be included here.
/Include/Coupling.inc	Contains all the coupler information. However, in most cases I may instead use the AAR-Type_E.inc file from North American V Scale (NAVS).
/Include/Brake.inc	Contains all the data for the braking system. If variations in braking force exist within the same family of cars, the general air brake data would be contained in a “Brake_General.inc” file, while the brake forces would be contained within a separate “BrakeForces.inc” file.
/Include/Lights.inc	Contains all the lighting code (if the car family is

	so equipped).
/Include/PassengerView.inc	Contains the passenger view information (if the family of cars is so equipped) including adding alternate viewpoints .
/Include/SpecialFX.inc	Contains data for particle-based effects—steam heat, stovepipe smoke, etc. (If the car family is so equipped)
/Include/Resistance.inc	Contains resistance data , including the Davis coefficients, wind resistance, bearing type, and rigid wheelbase.

In the case of freight cars that have separate empty and loaded physics variations, then the list of ***.inc** files would comprise the following:

*.INC FILE NAME	CONTENTS
/Include/GeneralData.inc	(same as in first table)
/Include/Coupling.inc	
/Include/Brake.inc	
/Include/Lights.inc	
/Include/SpecialFX.inc	
/Include/Resistance_Empty.inc	Same as “ Resistance.inc ” from first table.
/Include/Resistance_Loaded.inc	The resistance parameters for the car in its fully loaded state, which is set up as a static FreightAnim (albeit without a shape file unless it's an open-top or flat car).
/Include/Loads/<loadname>.inc	For open-top freight cars that have multiple visible loads. Each load is given its own *.inc file, which is contained in an additional subfolder labeled “Loads”. These, too, are set up as static FreightAnim parameters.

Here's what the ***.inc** file structure for a steam locomotive (tender engine assumed) might look like:

*.INC FILE NAME	CONTENTS
/Include/Loco/GeneralData_Wag.inc	Vehicle type, size, derail coefficients , external *.sms file to be used, intake points, etc. The shape file and name parameters are usually omitted, and remain in the main *.eng file. This is the data that would be located within the Wagon () section of the *.eng file.
/Include/Loco/Coupling.inc	Contains all the coupler information for the loco. Implies the loco has a permanent “Bar” coupling

	on the rear and either an “Automatic” or a “Chain” coupling on the front.
/Include/Loco/Brake_Wag.inc	Contains all the data for the braking system that would be located in the Wagon () section of the *.eng file—list of brake equipment, application/release rates, brake forces, etc.
/Include/Loco/Resistance.inc	Contains resistance data , including the Davis coefficients, wind resistance, bearing type, rigid wheelbase, and trailing locomotive resistance.
/Include/Loco/Lights.inc	Contains all the lighting code for the locomotive.
/Include/Loco/SpecialFX.inc	Contains data for particle-based smoke and steam effects .
/Include/Loco/GeneralData_Eng.inc	Locomotive type, number and size of driving wheels, weight on driving wheels, and sand capacity parameters. This is the data that would be located within the Engine () section of the *.eng file.
/Include/Loco/Boiler.inc	Contains all the info for the locomotive boiler —evaporative surface, superheating, volume, maximum boiler pressure, etc.
/Include/Loco/Cylinders.inc	Contains all the info for the locomotive cylinders —diameter and stroke, efficiency, crankpin angle difference, etc.
/Include/Loco/Firebox.inc	Contains all the info for the locomotive firebox —fuel calorific value, grate area, maximum feed rate, etc.
/Include/Loco/Fuel.inc	Contains the tender coal and water capacities and (if so equipped) water scoop data .
/Include/Loco/Brake_Eng.inc	Contains all the data for the braking system that would be located in the Engine () section of the *.eng file—main reservoir volume and air pressures, compressor data, charging rate, automatic and independent release rate, etc.
/Include/Loco/CabControls.inc	Contains all the parameters for the cab view controls, as well as the cab view *.cvf and in-cab *.sms file to be used.
/Include/Loco/Description.inc	Contains the description of the locomotive (if there are multiple locomotives within the same family).
/Include/Loco/Procedures.inc	Contains the operating procedures for the locomotive.
/Include/Tender/GeneralData.inc	Vehicle type, size, derail coefficients , external *.sms file to be used, etc. The shape file and

	name parameters are usually omitted, and remain in the main *.wag file.
/Include/Tender/Coupling.inc	Contains all the coupler information for the tender. Implies the tender has a permanent “Bar” coupling on the front and either an “Automatic” or a “Chain” coupling on the rear.
/Include/Tender/Brake.inc	Contains all the data for the braking system—list of brake equipment, application/release rates, brake forces, etc.
/Include/Tender/Resistance_Loaded.inc	Contains resistance data , including the Davis Coefficients, wind resistance, bearing type, and rigid wheelbase, and the like, when the tender is in its fully loaded state.
/Include/Tender/Resistance_Dynamic.inc	The resistance parameters for the tender in its empty and loaded states, which is set up as a dynamic FreightAnim . Also included is the coal load shape file (if the tender has visibly depleting coal) and the fuel and water intake points.
/Include/Tender/Lights.inc	Contains all the lighting code for the tender (if so equipped).
/Include/Tender/SpecialFX.inc	Contains data for particle-based effects—steam heat, water scoop, etc. (If so equipped)

A typical diesel locomotive might have these *.inc files (diesel-electric assumed):

*.INC FILE NAME	CONTENTS
/Include/GeneralData_Wag.inc	Vehicle type, size, derail coefficients , external *.sms file to be used, intake points, etc. The shape file and name parameters are usually omitted, and remain in the main *.eng file. This is the data that would be located within the Wagon () section of the *.eng file.
/Include/Coupling.inc	Contains all the coupler information for the loco.
/Include/Brake_Wag.inc	Contains all the data for the braking system that would be located in the Wagon () section of the *.eng file—list of brake equipment, application/release rates, brake forces, etc.
/Include/Resistance.inc	Contains resistance data , including the Davis coefficients, wind resistance, bearing type, rigid wheelbase, and trailing locomotive resistance. Also includes dynamic load information for the fuel and water tanks as well as fuel and water intake points
/Include/Lights.inc	Contains all the lighting code for the locomotive.

/Include/SpecialFX.inc	Contains data for exhaust and other particle-based effects .
/Include/GeneralData_Eng.inc	Locomotive type, number and size of driving wheels, weight on driving wheels, and sand capacity parameters. This is the data that would be located within the Engine () section of the *.eng file.
/Include/PrimeMover.inc	Includes all the data for the diesel engine , including fuel consumption, oil pressure, engine temperature, fuel tank capacity, etc.
/Include/TractionMotors.inc	Includes all the data for the traction motors , including horsepower, tractive effort curves and the like.
/Include/Brake_Eng.inc	Contains all the data for the braking system that would be located in the Engine () section of the *.eng file—main reservoir volume and air pressures, compressor data, charging rate, automatic and independent release rate, dynamic braking (if so equipped) etc.
/Include/Loco/CabControls.inc	Contains all the parameters for the cab view controls, as well as the cab view *.cvf and in-cab *.sms file to be used.
/Include/Loco/Description.inc	Contains the description of the locomotive (if there are multiple locomotives within the same family).
/Include/Loco/Procedures.inc	Contains the operating procedures for the locomotive.

This may not be ideal for your purposes, but it gives you an idea of how *.inc files may be utilized. Depending on your locomotive and rolling stock fleet, there may be additional *.inc files you will want to create for other standardized parameters.

The use of *.inc files and **Include ()** statements can, of course, be used in any MSTs-compatible file using the “SIMISA” header, and not just *.eng or *.wag files. However, the similar guidelines (see pages 3-4) apply.