ETCS Brake Model Tool 2.2
User's Guide

Thomas Brunnengräber

2024-11

This product or its documentation has been created using

- Java™ Platform, Standard Edition Development Kit (adoptium.net)
- NetBeans IDE (netbeans.apache.org)
- LaTeX(www.tug.org/texlive)
- GIMP (www.gimp.org)
- Inkscape (www.inkscape.org)
- Notepad++ (www.notepad-plus-plus.org)
- JCuda - Java bindings for NVIDIA® CUDA™ (www.jcuda.org)

and other free software.

This product includes software developed by the JDOM Project (www.jdom.org).

This product makes use of the Liberation Font provided by Red Hat, Inc.

Contents

1 Introduction

1.1 Intention and Functions of ETCS Brake Model Tool

ETCS Brake Model Tool is intended for calculations related to ETCS brake models in accordance with [Subset 026-3]. It supports the following functions:

Calculation of fix brake models: Based on the nominal forces or decelerations and the probabilistic characteristics of the train brakes, the on-board correction factors are calculated. Two algorithms are implemented: by Monte-Carlo simulation and by combinatorial logic.

Calculation of scenarios: For a given set of train model (fix or flexible), track profile, national values and optionally normal service brake model, the brake distances and supervision limits are calculated.

Comparison of scenarios: Two scenarios can be compared with respect to resulting brake distances and supervision limits.

Limits chart: The speed and distance limits related to one or two scenarios are displayed graphically.

Export: Export of all calculated data in text format (`XML` or `CSV`). Export of charts as vector graphic (`SVG` format) or bitmap (`PNG` format).

1.2 Terms and Abbreviations

Table 1: Terms and abbreviations

Term	**Meaning**
AD	Deceleration
BMI	Brake Model Index, determines which of the brake models created for a fixed train shall be used for calculations if no brake is inhibited by trackside
EBCL	Emergency Brake Confidence Level

Continued on next page

Table 1: Terms and abbreviations

Term	**Meaning**
project member	Data structures necessary for calculations. The following *project members* are implemented: • *flexible train models* • fix train models in three variants: – direct input of the correction factors – calculation of the correction factors by Monte-Carlo simulation in accordance with [prEN 17997-1] – calculation of the correction factors by combinatorial logic • *normal service brake models* • *sets of national values* • *track profiles* • *scenarios* • *libraries* containing some values that can be used for the Monte-Carlo simulations

In ETCS Brake Model Tool and thus also in this document the names are shortened to T_{bu} instead of T_brake_build_up and $AD(V)$ instead of A_brake_emergency(V).

1.3 Conventions

- A term in *slanted letters* indicates a term with a certain meaning within ETCS Brake Model Tool, e. g. a type of data objects.
- A term in **bold letters** indicates a menu, command or button name.
- A term in 'single quotation marks' indicates a fixed term not directly related to ETCS Brake Model Tool.
- A term in "double quotation marks" indicates a name or a quote. It is also used to indicate, that a term or statement is not literally correct (e. g. a simplification or common but imprecise wording).

1.4 The Desktop

After starting ETCS Brake Model Tool, the graphical user interface will be started and the latest project will be loaded automatically.

The desktop has the following areas:

- The menu bar.
- The tool bar.

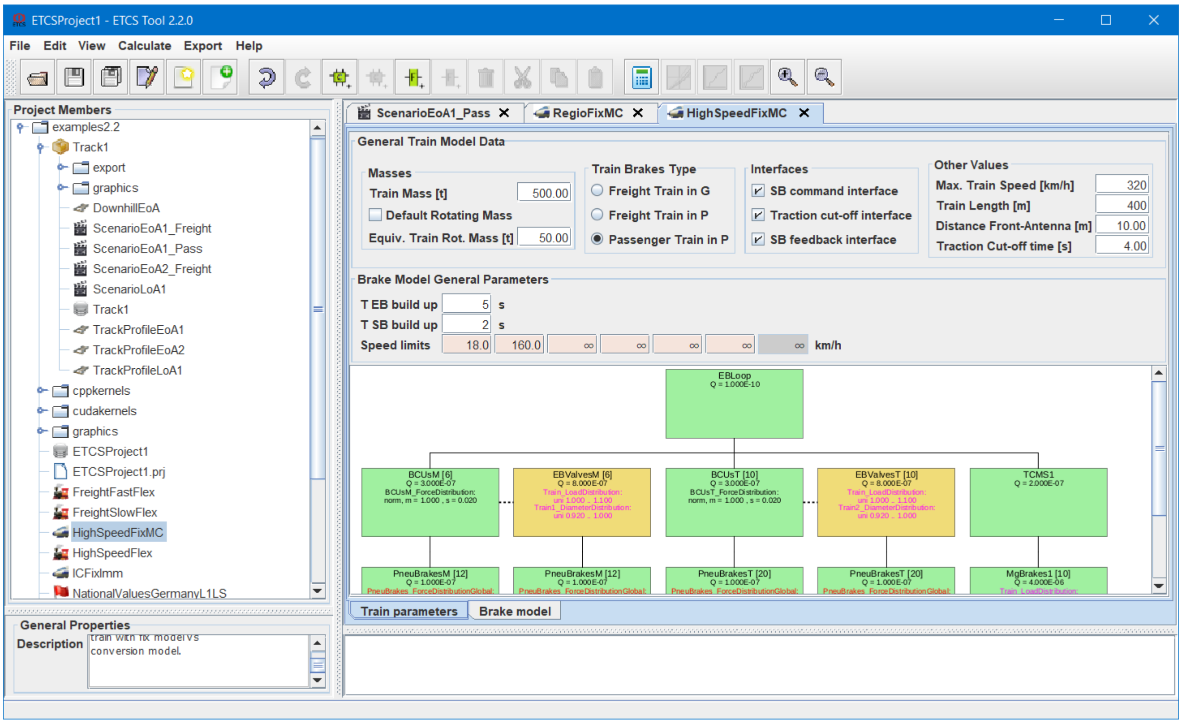


Figure 1: The desktop

- The *member tab pane*, with the active member presented in the active tab. If you click on some *project member* in the project tree or if you create a new *project member*, all its data will be shown in the member tab.
- If no *brake failure model* or *brake component* is selected:
  - The project tree showing the files in the project directory and sub-directories.
  - A panel showing some (optional) description of the member currently shown in the member tab.
- If a *fix train model by combinatorial logic* is active and a *brake failure model* is marked: A panel showing the properties of the marked *brake failure model*, see figure 9. If you click the mouse outside the *brake failure models* table or a component block, the project tree is shown again.
- If a *fix train model by Monte Carlo simulation* is active and a *brake component* is selected: A panel showing the properties of the marked *brake component*, see figure 17. If you click the mouse outside the *brake failure models* table or a component block, the project tree is shown again.
- A message output window displaying hints, warnings or errors occurring during file operations and calculations.
- A status bar displaying hints, if an action could not be performed.

1.5 Floating Window

If you right-click in the heading of a model graphics tab, you can select to show the model graphics tab in a separate floating window. This is helpful to compare two models, in particular if you've got a second physical screen. Multiple model tabs can be moved to the floating window and back again. Note that in the floating window, not all editing functions might be available.

2 Projects

The *project* organizes all data necessary to calculate one or more *scenarios*. Therefore the first action after starting ETCS Brake Model Tool is either opening an existing *project* or creating a new one. Only one *project* can be open at a time.

A *project* that has not been saved after the latest modification, is marked with an asterisk '*' in the window title.

2.1 Packages

All *project members* are organized in *packages*. Each *package* can contain one or more *project members*. Each *package* contains exactly one *library*.

There is always one *global package*. The *global package* has the same name as the *project* and is located in the project directory, i. e. all member files in the project directory are members of the global package.

There might be any number of *local packages*. Each *local package* is located in an immediate sub-directory of the project directory. The name of a *local package* is given by its directory name.

A *scenario* has access to the other *project members* of its own *package* and the *global package*. A *scenario* of the *global package* has access to its own *package* only, accordingly.

In general, you should keep the *global package* as empty as possible, i. e. create most data in *local packages*. In particular, you should create a separate *local package* for each scenario you want to analyze. Only those *project members*, that need to be referred by *scenarios* of different *packages* should be contained in the *global package*.

A new package is created by **File – Create new Package**. You will be asked for the name of the new package. A sub-directory with the given name will be created in the project directory.

2.2 Project Members

ETCS Brake Model Tool provides *project members* of the following types:

- *flexible train models*
- fix train models in three variants:
  - direct input of the correction factors
  - calculation of the correction factors by Monte-Carlo simulation
  - calculation of the correction factors by combinatorial logic
- *normal service brake models*
- *sets of national values*
- *track profiles*
- *scenarios*

A new *project member* is created by **File – Create new Member**. The "Create New Member Dialog" will open, where you can select the *package* the new member shall belong to, and the name and type of the new member.

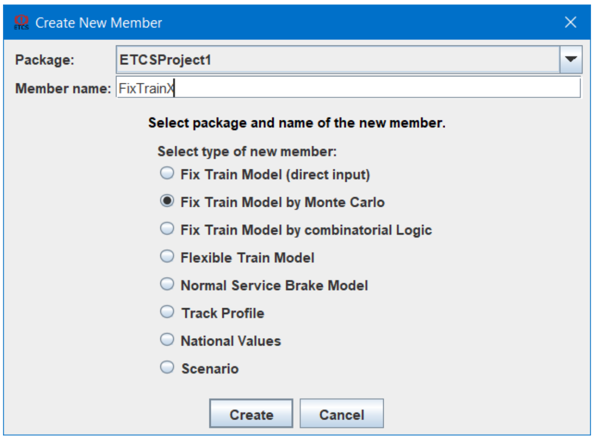


Figure 2: The create new member dialog

All member files found in a package directory will be loaded when opening a *project*.

2.3 Files

The following files are created and handled within a *project*:

- One project (`.prj`) file per *project*.
- One library (`.lib`) file per *package*.
- Optionally one or multiple *project member* files per *package*.

All files are text files following `XML` schemes. Therefore they can be read and their information can be interpreted and even changed manually (if someone considers this useful).

In addition certain evaluation results, intermediate results and graphics can be exported to files. Those files are described together with the related export command, see section 9.6.

2.3.1 Project Files

The project properties as entered and shown in the *project properties dialog* are stored in the `.prj` file. The directory containing the project file is the project directory, and also contains the files of the *global package*.

2.3.2 Member Files

Each *project member* has a name, that must be unique within the *package*. The name of the *project member* is the same as its file name.

The *flexible train model* data is stored in one text file in XML format containing the information shown in the *flexible train model properties panel* (see figure 5). The file name of the *flexible train model* file must be extended by **.flex**.

For each of the three variants of fix train models a specific XML format is used. In case of a *fix train model (direct input)*, the data is stored in a file extended by **.fiximm**. In case of a *fix train model by combinatorial logic*, both the information shown in the *fix train model panel* (see figure 7) and the resulting brake model data are stored in a file extended by **.fixcomb**. In case of a *fix train model by Monte Carlo simulation*, both the underlying train model entered by the user (see figure 17), the simulation results and the final brake model data are stored in a file extended by **.fixmc**.

The *normal service brake model* data is stored in one text file in XML format containing the information shown in the *normal service brake model panel* (see figure 23). The file name of a *normal service brake model* file must be extended by **.nsbm**.

The *set of national values* data is stored in one text file in XML format containing the information shown in the *national values panel* (see figure 24). The file name of a *set of national values* file must be extended by **.nv**.

The *track profile* data is stored in one text file in XML format containing the information shown in the *track profile panel* (see figure 25). The file name of a *track profile* file must be extended by **.tp**.

The *scenario* data is stored in one text file in XML format containing the information shown in the *scenario panel* (see figure 26). The file name of a *scenario* file must be extended by **.scn**. Note: The data calculated for a scenario is not saved in the *scenario* file. It can be exported for further use, see section 9.6.

2.3.3 Files not belonging to the Project

If you want a certain file to be excluded from the project, but not delete it completely, you can remove it by **File – Remove active Member**. This will unload the *project member* and add `_ignore` to its file extension.

You can add an existing file, whose file name is extended by `_ignore`, to any *package* by **File – Add existing Member**.

Of course you can also add, rename or delete files using a file system tool, such as Windows Explorer, but you shouldn't do this while ETCS Brake Model Tool is running in order to avoid inconsistencies.

2.4 Libraries

There is one *library* in each *package*. In the *library* you can define alias names for numeric values.

In *fix train model by Monte Carlo simulation* you can use the alias name instead of the particular value. Thus it is possible to modify the parameters of many components by changing just the numeric value in the library. In addition, an alternative value can be defined. This

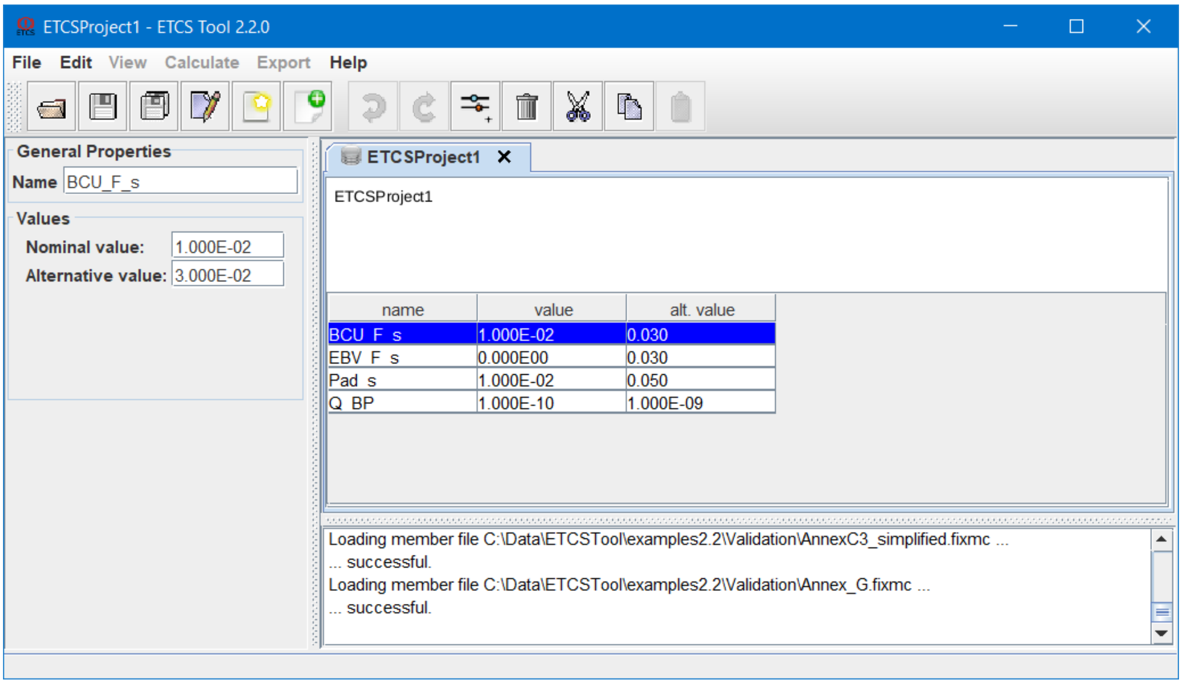


Figure 3: The library panel

alternative value is used in addition to the nominal value when you perform a Monte-Carlo simulation with alternatives, see section 4.4.9.

2.5 The Project Properties Dialog

In the *project properties dialog* all options relating to multiple models of the project can be set. This information is stored in the project file in the project directory (extension `.prj`).

2.5.1 General Tab

Path: The storage location of the project.

Name: A user defined identifier of the *project*. The name is displayed in the title of the ETCS Brake Model Tool window.

Description: An optional description of the *project*.

The new name and description have to be acknowledged by the **OK** button.

2.5.2 Charts Tab

Select the color and line style for each line shown in limit charts.

The default colors and line styles are re-activated by pressing **Set to default styles** .

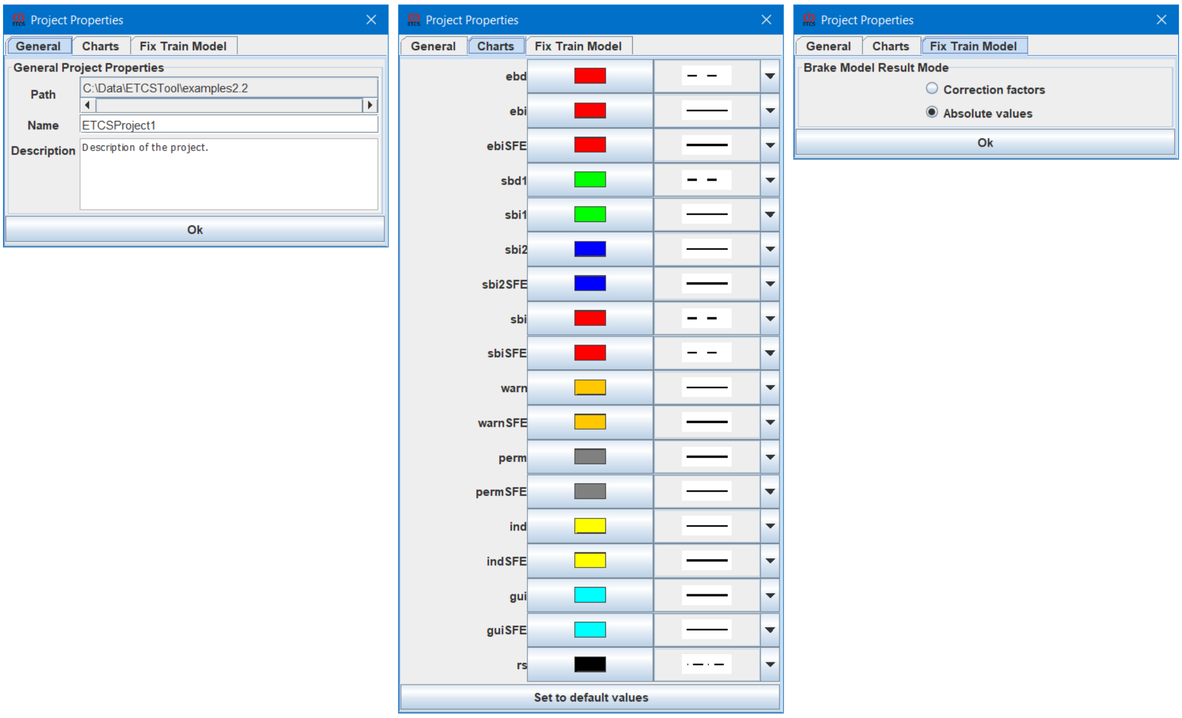


Figure 4: The project properties dialog

2.5.3 Fix Train Model Tab

Select whether the correction factors `Kdry_rst` are exported as correction factors actually, or as absolute deceleration values.

3 Flexible Trains (Conversion Model)

The train model describes all physical parameters of the train, that are necessary to calculate the brake intervention limits and derived limits.

For trains for which no suitable fix train model is stored in the ERTMS/ETCS on-board, the conversion model has to be used. These trains are called flexible trains. The values $AD(V)$ are calculated based on

- the brake percentage (sometimes denoted by λ, therefore the conversion model is often called Lambda-model)
- the brake position, which is one out of

  FG Freigt train in brake position "G" (slow brake force build-up)

  FP Freigt train in brake position "P" (fast brake force build-up)

  PP Passenger train in brake position "P" (or "R", because for "R" the same conversion is used)

- the train length (because brake activation time increases with train length)

3.1 The Flexible Train Model Properties Panel

The data that is necessary for the conversion model is entered in the flexible train model properties panel, see figure 5.

All properties of the *flexible train model* are stored in the flexible train model file (extension **.flex**).

3.1.1 General Train Model Data

Data shown in this panel is necessary for both fix and flexible train models.

Masses

In order to consider the effect of track gradients, the quotient of the train rotating mass and the train mass is necessary.

If the train masses are either constant or determined by the train control or entered by the driver, they can be stated here.

If the rotating mass is unknown, select default rotating mass. In that case, also the train mass will not be used.

Train Brakes Type

Select the type of train brakes. In case of a passenger train with brakes different from UIC type brakes, select "Passenger Train in P" as well.

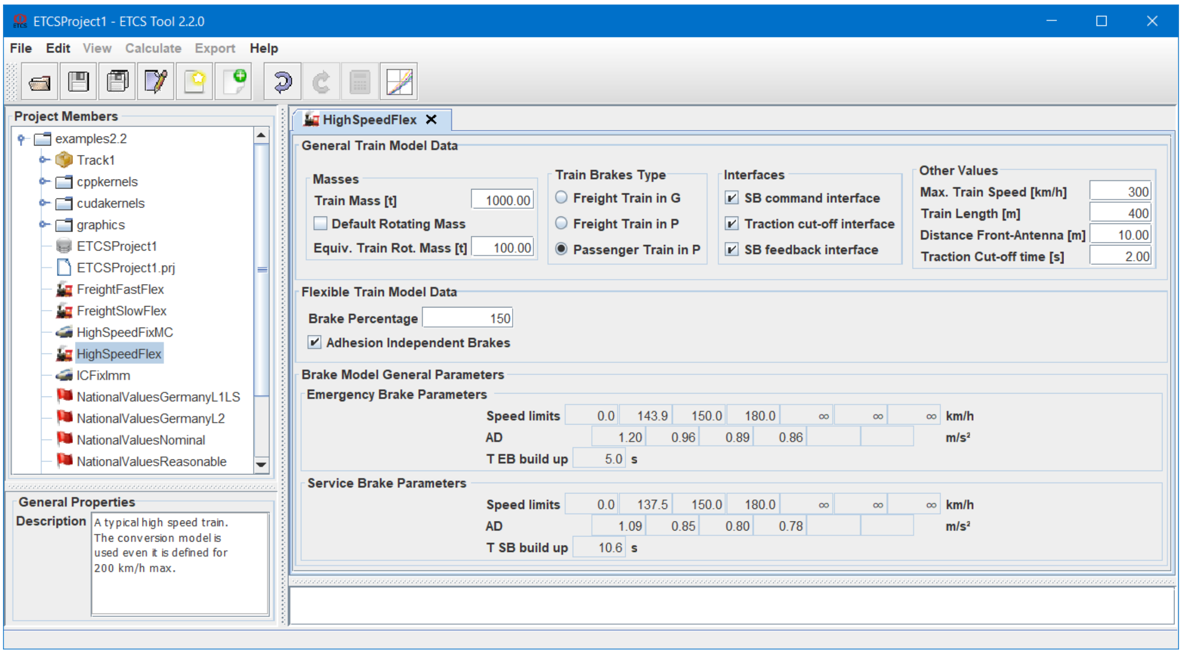


Figure 5: The flexible train model properties panel

Interfaces

Select whether the technical means are implemented for the ERTMS/ETCS on-board to order a full service brake or to cut off the traction independently from other actions (such as emergency brake command).

Other Values

Maximum train speed, the train length, the distance of the Eurobalise antenna from the front end and the time to cut off the traction after commanding the emergency brake need to be stated here.

3.1.2 Flexible Train Model Data

State the brake percentage as entered by the driver here and select whether the train is equipped with adhesion independent brakes.

3.2 Calculation of the Brake Model Parameters

The conversion model defined in [Subset 026-3] is applied by **Calculate – Calculate Brake Models** or the calculation button.

The conversion is automatically performed whenever it is needed in a *scenario*.

The result of the conversion model is shown in the "Brake Model Parameters" section of the train model panel.

4 Fix Train Models

4.1 Explanations valid for all kind of Fix Train Models

For trains that usually operate in a foreseeable train configuration with known brake characteristic (like EMUs or DMUs), the brake model parameters can be determined in the development phase and stored in the ERTMS/ETCS on-board. Those trains are called fix trains or gamma trains. The values $AD(V)$ are calculated and/or measured in the design phase of the train, whereupon the limits $V_1, \ldots, V_m$ can be selected in such way, that the actual deceleration curve $a(v)$ is suitably approximated. The brake model defined like this does not yet consider

1. deviations due to component tolerances,
2. losses of brake capacity due to failures of the brake components (either already existing but not yet detected failures or failures arising in the moment of the emergency brake request),
3. extensions of the distance due to bad adhesion conditions.

The effects of point 1 and 2 are modelled in correction factors Kdry_rst, that need to be defined for each speed range and ten Emergency Brake Confidence Levels, EBCL 0 to EBCL 9, leading to Kdry_rst(V,EBCL). Thus there is in fact not only one brake model but ten for the same physical train configuration. The brake model for EBCL=0 is the nominal brake model, the brake models for EBCL>0 are called Safe Brake Models. Which of the ten models is to be used is determined by the track-side, sending the required confidence level in variable M_NVEBCL.

In [Subset 026-3] no correction factor is defined for the brake build-up time T_{bu}, nevertheless some parts of the brake system (namely the ep-brake system) only effect the build-up time. Therefore it makes sense to correct also this parameter by a factor KT_rst(EBCL).

To make things even more complicated, the brake model can change while running not only due to a change of the EBCL, but also because some special brakes can be inhibited and re-enabled by track-side command. Therefore multiple brake models need to be defined for the different configurations of the train brake system, out of them exactly one needs to be selected by the vehicle control (or brake control) while running according to the status of the special brakes. Each of these sets of brake models (consisting of 10 models each, one for each EBCL) gets a number, which is named Brake Model Index (BMI) in ETCS Brake Model Tool.

The BMI needs to reflect the status of the special brakes (as required in [Subset 026-3]), but obviously it can also encode additional information such as unavailable parts of the brake system that are already known (e. g. as the result of the start-up brake test), or the train configuration (one consist, two consists coupled etc.).

The three lower bits of the brake model index have a fixed meaning, since they shall always code the status of the special brakes as defined in [Subset 034]. Therefore their name, sequence and size (1 bit) cannot be changed in ETCS Brake Model Tool.

To get complete here, it shall be mentioned, that also the effects of point 3 are considered in the calculation of braking distance, therefore correction factors Kwet_rst(V) are defined for each speed range. In contrary to the Kdry_rst(V, EBCL) they do not depend on the EBCL and thus Kwet_rst(V) is identical for each nominal brake model and the related 9 safe brake models.

Thus finally the deceleration used in the calculation of the braking distance is given according to [Subset 026-3], paragraph 3.13.6.2.1.4 as

$$\begin{aligned}\text{A\_brake\_safe}(\text{V}, \text{d}) = {} & \text{A\_brake\_emergency}(\text{V}, \text{d}) \cdot \text{Kdry\_rst}(\text{V}, \text{M\_NVEBCL}) \\ & \cdot \big(\text{Kwet\_rst}(\text{V}) + \text{M\_NVAVADH} \cdot (1 - \text{Kwet\_rst}(\text{V}))\big)\end{aligned}$$

The national values M_NVEBCL and M_NVAVADH don't change within the braking distance *d*, whereas the model may change due to a change of track conditions, indicated by the "d".

Each *brake model* of a fix train model consists of

- its index, encoding the configuration of the brake system (i. e. the train composition, the available brake components etc.)
- 7 nominal deceleration values (EBCL=0)
- 1 nominal brake build-up time (EBCL=0)
- 9 × 7 safe deceleration values for EBCL=1 to 9
- 9 × 1 safe brake build-up time values for EBCL=1 to 9
- 1 × 7 K_wet values
- 1 × 7 service brake decelerations
- 1 service brake build-up time

Fix brake models are created by **Edit – Validate Brake Models** (in case of a *fix train model (direct input)*), or **Edit – Calculate Brake Models** (in case of a *fix train model by combinatorial logic* or *fix train model by Monte Carlo simulation*). In case of *fix train model by combinatorial logic* and *fix train model by Monte Carlo simulation*, the parameters of the fix brake models are shown in a separate tab 'Brake models'. This table displays all values of all *brake models* either as absolute values or as correction factors, see 2.5.

Summary

All values used for the calculation of the braking distance except of the so called 'national values' M_NVEBCL and M_NVAVADH need to be calculated in the engineering process.

Each ETCS emergency brake model in fact consists of 10 brake models:

- The nominal brake model (related to EBCL=0) and 9 safe brake models (related to EBCL=1 . . . 9).
- For EBCL=0 all Kdry_rst(V) are equal to 1, thus AD(V,EBCL=0) = AD(V).
- For EBCL>0 all Kdry_rst(V,EBCL) are less than 1,
  AD(V,EBCL>0) = AD(V) · Kdry_rst(V,EBCL).

The brake model index (BMI) is the same for EBCL=0. . . 9.

The brake model can change while running due to change of EBCL or BMI. The BMI can change while running for different reasons, e. g. due to inhibition/re-enabling of special brakes or brake failures. ERTMS/ETCS on-board should not stop the train if the BMI changes, except if there is no brake model available for the new BMI (should not happen), or the new brake model cannot ensure sufficient deceleration for the given EBCL and the track profile. If someone sees a need to stop the train for the driver to do something, this is outside the scope of ETCS, thus the brake system or vehicle control must stop the train if necessary.

The three options how to determine or define fix train model are described in the following three subsections.

4.2 Direct Input of the Parameters of a Fix Train Model

If you already have the parameters of the fix train model(s), you can create a fix train model by **File – Create New Member** and selecting "Fix Train Model (direct input)". All properties of the *fix train model (direct input)* are stored in the train model file with extension **.fiximm**. The input screen consists of three sections as shown in figure 6.

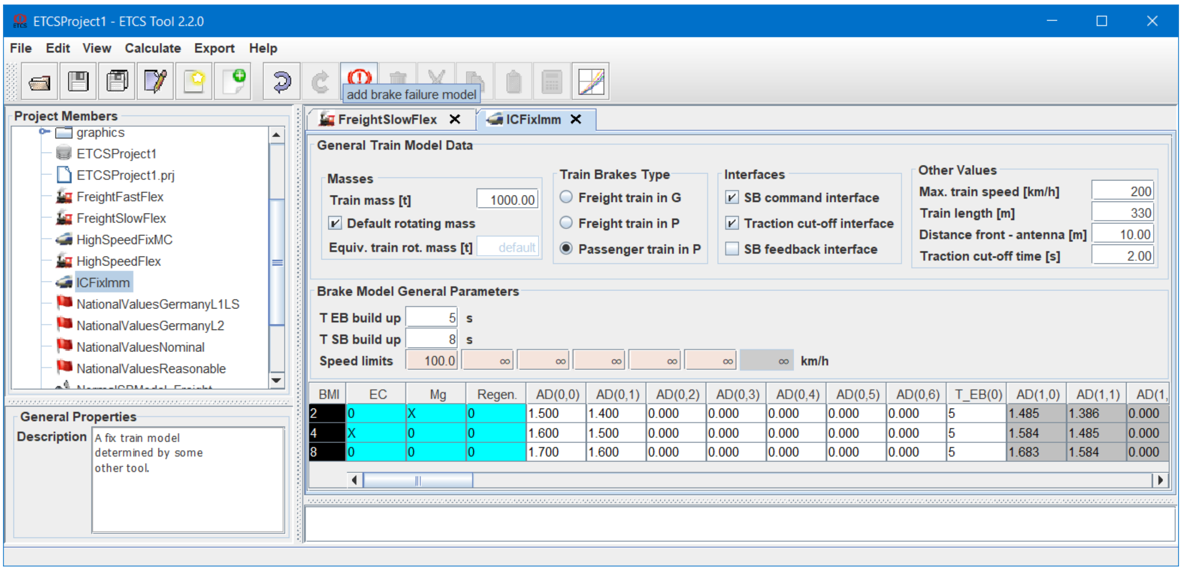


Figure 6: Immediate input of fix train model parameters

4.2.1 General Train Model Data

This data is necessary for both fix and flexible train models. It is described in section 3.1.1.

4.2.2 Brake Model General Parameters

Here you have to enter the brake build up times and the upper speed limits of each section. The last section has no upper limit, i. e. if you want to define two speed sections, you only have to enter one value.

4.2.3 Parameter Table

The third section is the table containing all brake models with all parameters necessary for a fix brake model according to [Subset 026-3].

A new brake model is created by **Edit – Create Brake Model**. After entering the BMI of the new brake model, a new line will be shown. The last three bits of the BMI define whether the brake model includes eddy current brakes and/or magnetic track brakes and/or regenerative brakes. E. g. BMI 13 is binary (1)101b, i. e. it considers eddy current brakes and regenerative brakes according to the definitions in ETCS Brake Model Tool. Maybe your particular ERTMS/ETCS on-board has other definitions, in that case the index might have to be adjusted when transferring the values to the actual ERTMS/ETCS on-board.

In each line, you have to enter the nominal decelerations for each speed section, the correction factors or decelerations for each EBCL, the EB build up times, the correction factors for wet rails and the service brake model.

4.2.4 Validation

The brake models are validated by selecting **Calculate – Validate brake models**. After that, the brake distances as function of initial speed and EBCL can be displayed (see section 4.5) and scenarios can be calculated (see section 8).

4.3 Calculation of Correction Factors by combinatorial Logic

Note: This method of determination of fix brake models is not recommended for new projects anymore, since it is not completely conform to [prEN 17997-1].

ETCS Brake Model Tool is able to calculate all nominal brake models, all safe brake models (the Kdry_rst values for each EBCL) and the Kwet_rst values based on the *master nominal brake model* and some information related to brake failures, given by *brake failure models*, see section 4.3.3.

The panel for entering all necessary data is shown in figure 7.

4.3.1 General Train Model Data

The general train model data is the same as for the flexible train model, see section 3.1.1

4.3.2 Master Brake Model

The *master nominal brake model* defines the speed dependent decelerations and build-up time for EBCL=0 when all components of the brake system are available. It also defines standard deviations for these values in order to model statistical deviations of the physical parameters of the components. It has no specific meaning outside the engineering process, thus it is not

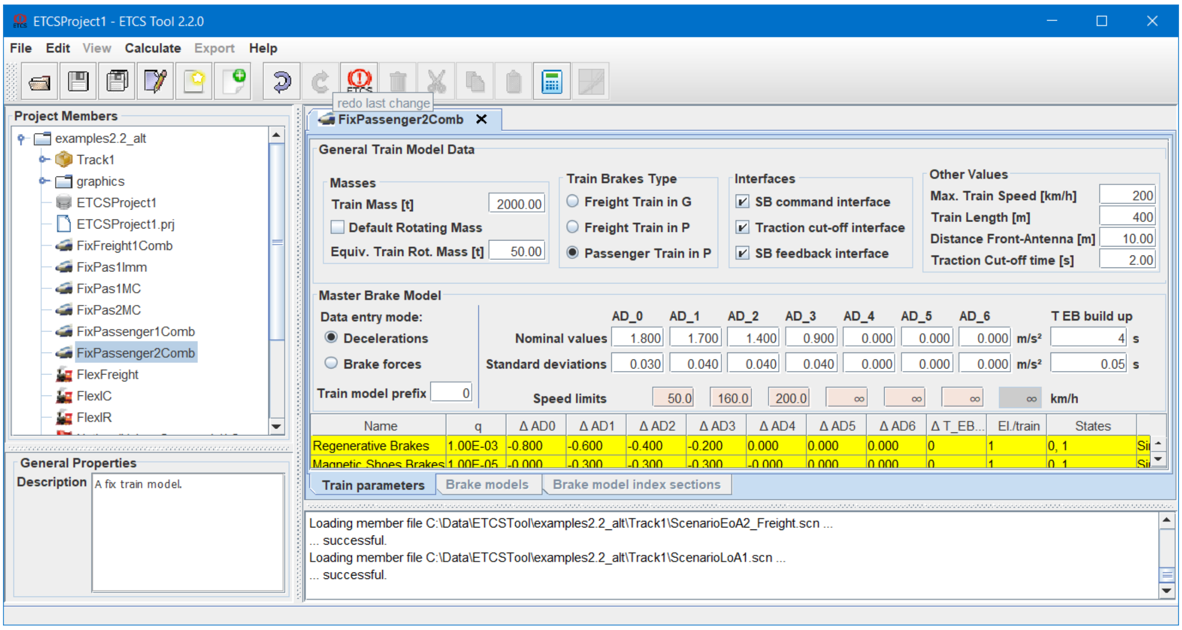


Figure 7: The parameters necessary to determine the fix brake models by combinatorial logic

noticeable in the set of brake models or the brake model index. In the set of brake models created by ETCS Brake Model Tool, it will be the one with the highest BMI.

Data entry mode

The nominal braking capacity and the differences caused by brake failures can be entered in two ways: as deceleration values or as brakes forces. If "deceleration" is selected, the braking capacity is entered in terms of decelerations. If "brake forces" is selected, the braking capacity is entered in terms of brake forces. Since finally always decelerations are needed, also the train mass and the equivalent train rotating mass must be given in case "brake forces" is selected in order to convert the input to decelerations. If the train mass or the train rotating mass is changed, the deceleration values will be recalculated accordingly. Thus don't change the train masses after you entered the correct decelerations.

Train model prefix

The train model prefix is used to distinguish the *brake models* created by multiple independent *fix train models by combinatorial logic*. It is part of the *brake model index* of each created *brake model*, located in the bits above the sections needed for the coding of the brake status.

Nominal values

In the first line the nominal parameters are entered — either decelerations or brake forces. All values are positive values. The deceleration is limited to 2.55 m/s^2. The nominal brake build up time has to be stated as well.

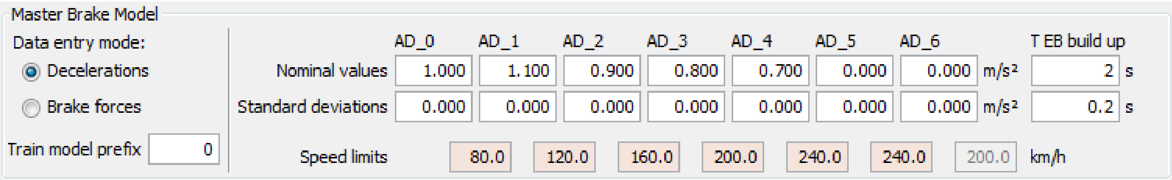


Figure 8: The master brake model parameters table

Standard deviations

Both decelerations and brake build up time depend on a lot of actual physical parameters of a lot of components. Due to the central limit theorem, the overall distribution function will be close to the standard distribution therefore. The related standard deviations are entered in the second line.

Speed limits

The upper speed limits of the seven speed sections. The last speed section has no upper speed limit. If you need less than seven sections, type 'i' into the speed limit text field(s) that you want to set to "infinity" (see figure 7).

In addition to this data, the deceleration effect of each component or group of components needs to be stated.

4.3.3 Brake Failure Models

A *brake failure model* contains all data related to the unavailability of a certain part of the train brake system, that is necessary to consider its effect to the braking capacity of the train:

- the effect on deceleration or brake forces and/or brake build-up time
- the number of parts of this type in the train
- the relation to other failures in the brake system
- the probability of the part being unavailable when demanded
- whether it is independent of wheel-rail adhesion, and the decrease of deceleration under conditions defined in EN 15595, see [Subset 040]
- whether it is used by service brake as well

Since the effect to braking distance is identical, independent from if the unavailability is known before or not, this data is also necessary (except of the probability) and sufficient to calculate all nominal brake models. Thus a *brake failure model* does not exclusively describe the functional safety parameters but also the operational effects. Therefore the inhibition of a special brake is also modelled by a *brake failure model*. If the inhibition and re-enabling would be perfectly safe, the probability of this failure would be zero. But in fact it can never be excluded that the command or status signal fails, so there is always a probability, that the special brake will not work as assumed by the ERTMS/ETCS on-board.

An unavailability that is known early enough (e. g. due to a brake test at start of mission or due to a track-side inhibition) doesn't contribute to this probability. So if the complete system is tested at start-up and the test has a good fault detection and is sufficiently reliable, only those failures have to be considered, that occur within some few hours. Thus this probability is highly depending on the specific train.

All *brake failure models* related to a *fix train model by combinatorial logic* are presented in the brake components table in the middle of the graphics panel.

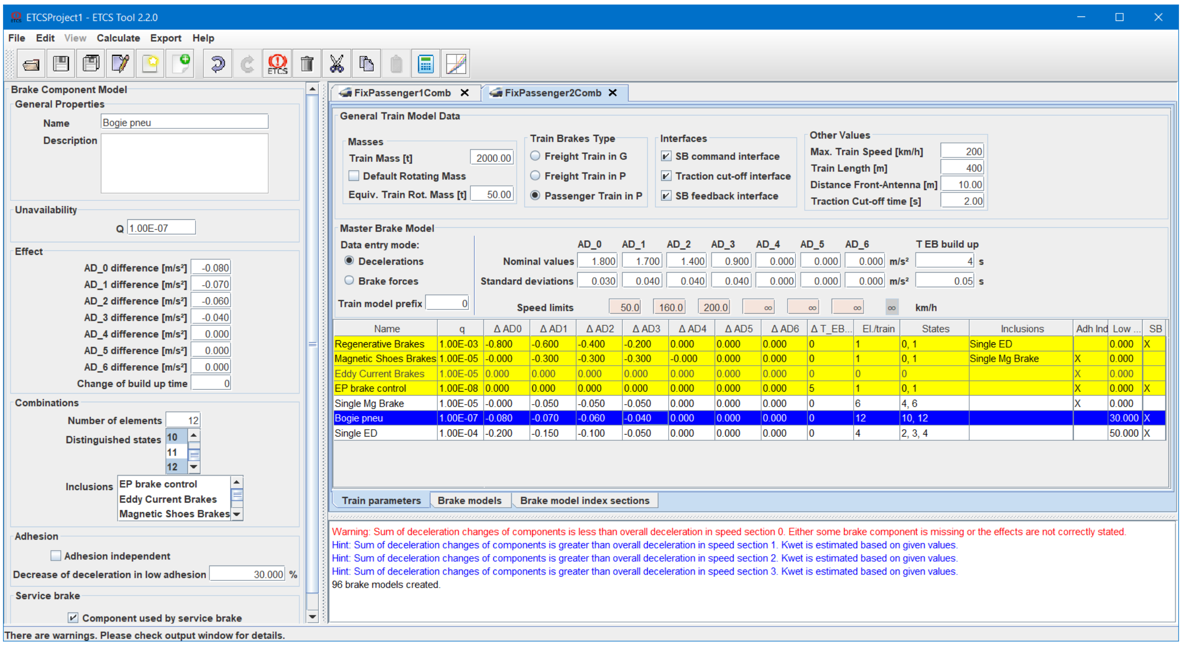


Figure 9: The brake component models table and properties panel

This table displays all values of the *brake failure models*. If a *brake failure model* is selected in this table, its properties can be changed in the properties panel on the left.

4.3.4 The Brake Failure Models Properties Panel

General Properties

Name A user defined name of the *brake failure model*. The name must be unique within the train model.

Description A user defined description of the *brake failure model*.

Unavailability

Describes the probability of a failure on demand, given by the unavailability $\overline{Q}$.

Effect

The effect if this component is not available. Most unavailabilities will affect the deceleration (equivalent to the brake forces). Some components will only or in addition affect the brake build-up time. Since deceleration will always decrease, only negative values are allowed for the changes of deceleration or brake forces. The unavailability of some components might also lead to a decrease of the resulting overall mean brake build-up time [1] , therefore positive and negative values are possible.

Combinations

Number of elements The number of physical or logical elements in the train, that can fail by this failure mode.

Complete independence is assumed between these elements. For global failures, e. g. common cause failures, failures of the overall control system, failures of command or status signals etc., set this value to 1.

Distinguished states Distinguished numbers of available elements. In order to decrease the number of brake models it makes sometimes sense not to assign a separate brake model to each and every possible brake system state, but to group some states. For example if the train has 20 bogies whose pneumatic brakes can fail independently, it makes sense not to have a separate brake model for: no (known) unavailable bogies, for 1, for 2 etc. up to maybe 10, but only one for 0 or 1 unavailable bogies (19 avail.), one for 2 to 4 unavailable bogies (16 avail.), and one for 4 to 10 unavailable bogies (10 avail.). Thus only the states "at least 10 bogies available", "at least 16 bogies available" and "at least 19 bogies available" are distinguished.

In order to add or remove a single number, press the "Ctrl" key while clicking the number you want to add or remove.

Inclusions Tell the algorithm that some other failure (or known unavailability) will have no additional effect if this failure (or known unavailability) occurs.

Only elements existing once per train can include other failures. If the failures of only some elements related to another failure are superseded by this failure, e. g. if a failure of this component affects only 5 out of 10 local elements, create two separate *brake failure models* of this type (with number of elements per train = 1 for each), and two seperate local element components (with number of elements per train = 5 for each).

Adhesion

Activate the checkbox if the brake force created by this component doesn't depend on wheel-rail adhesion.

[1] namely if a brake with big delay fails, the mean delay time will decrease

If the checkbox is deactivated, the decrease of deceleration (corresponding to the increase of braking distance according to EN 15595, see [Subset 040]) shall be entered as well.

Service brake

Activate the checkbox if the component is used also for service brake.

4.3.5 Brake Model Index Sections

The *brake model index* (BMI) is a 16 bit value. This 16 bit value is divided up into sections representing the status (availability) of certain parts (components) of the train brake system, including non-friction brakes such as regenerative brakes.

Each section typically corresponds to one *brake failure model*. The only exception is, if the effect of a *brake failure model* is included in another *brake failure model*, see the example below. The length of each section is determined by the number of distinguished states to be encoded in this section. The sequence of sections is identical to the sequence of *brake failure models* in the *fix train model by combinatorial logic*.

For example a failure of the (global) command signal for the magnetic shoes brakes (Mg brakes) will result in all Mg brakes not applied on demand. Thus the *brake failure model* for the command signal includes (supersedes) the *brake failure model* for local failures of the Mg brakes. Let's further assume that there are 5 Mg brakes installed. The global command failure will result in all Mg brakes failing. The probability of this failure is usually much higher than the independent sporadic failure of 2 (or more) local Mg brakes. Thus it usually makes sense to define the standard case (all elements available) as one state, a sometimes needed partial defect state as a second state, and the complete loss as a third state.

state 1: 0 Mg brakes available (or 1 or 2)
state 2: 3 Mg brakes available (or 4)
state 3: 5 Mg brakes available

By this the number of brake models can be drastically reduced compared to coding each possible number of available Mg brakes (0-5) as a separate state. Since state 1 is already included in the global failure, if the global failure is described in the third *brake failure model* of the *fix train model by combinatorial logic* and the local failure in the sixth, the *brake model index* will be composed like this (assumed the fourth and fifth brake failure model is encoded in one bit):

xxxx xxxx xxLx xGxx

with G being the bit for the global failure, L the bit for the local failure.

The resulting encoding is shown in table 2, "d" means "don't care".

If 3 and 4 available Mg brakes shall be encoded separately, the encoding will be like this:

xxxx xxxx xLLx xGxx

Table 2: Brake model index sections, example 1

No of avail. elem.	L	G
0, 1, 2	d	0
3, 4	0	1
5	1	1

Table 3: Brake model index sections, example 2

No of avail. elem.	L L	G
0, 1, 2	d d	0
3	0 0	1
4	0 1	1
5	1 0	1

with states listed in table 3.

The tab 'Brake model index sections' shown in figure 10 provides information about the bits in the *brake model index*, so that the vehicle side train control software can be developed correctly.

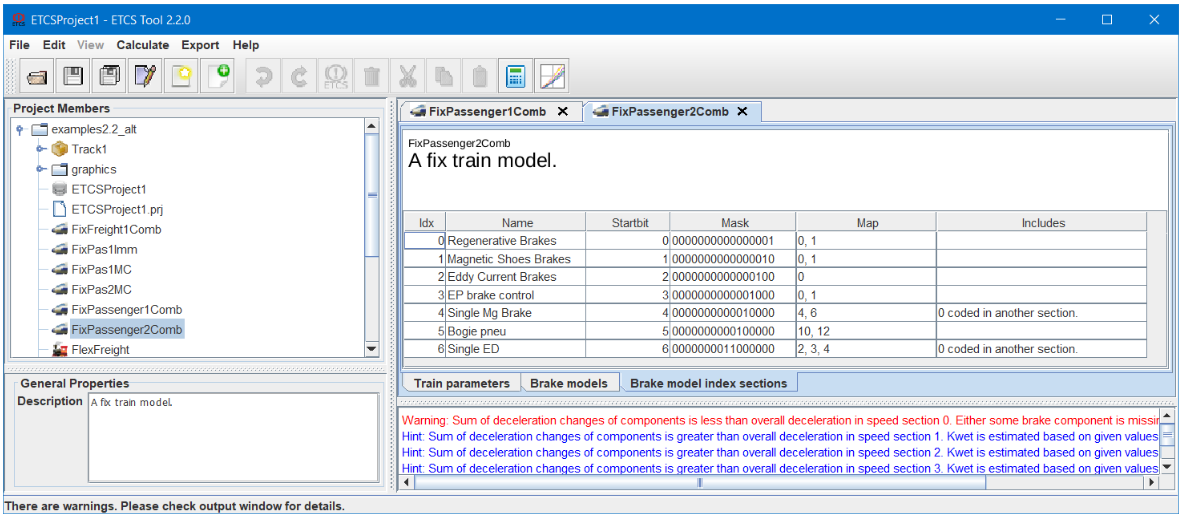


Figure 10: The brake model index sections table

In order to facilitate and ensure correct creation of the *brake model index* by a train control software, ETCS Brake Model Tool can create a C-code template, that will correctly reflect the content of the *brake model index*, see section 4.6.

4.3.6 Brake Models Table

Brake models are created when the **Edit – Calculate Brake Models** command is executed. After that the tab 'Brake models' can be selected. All *brake models* related to the train model will be presented in a table in the graphics panel, see figure 11. This table displays all values of all *brake models* either as absolute values or as correction factors,

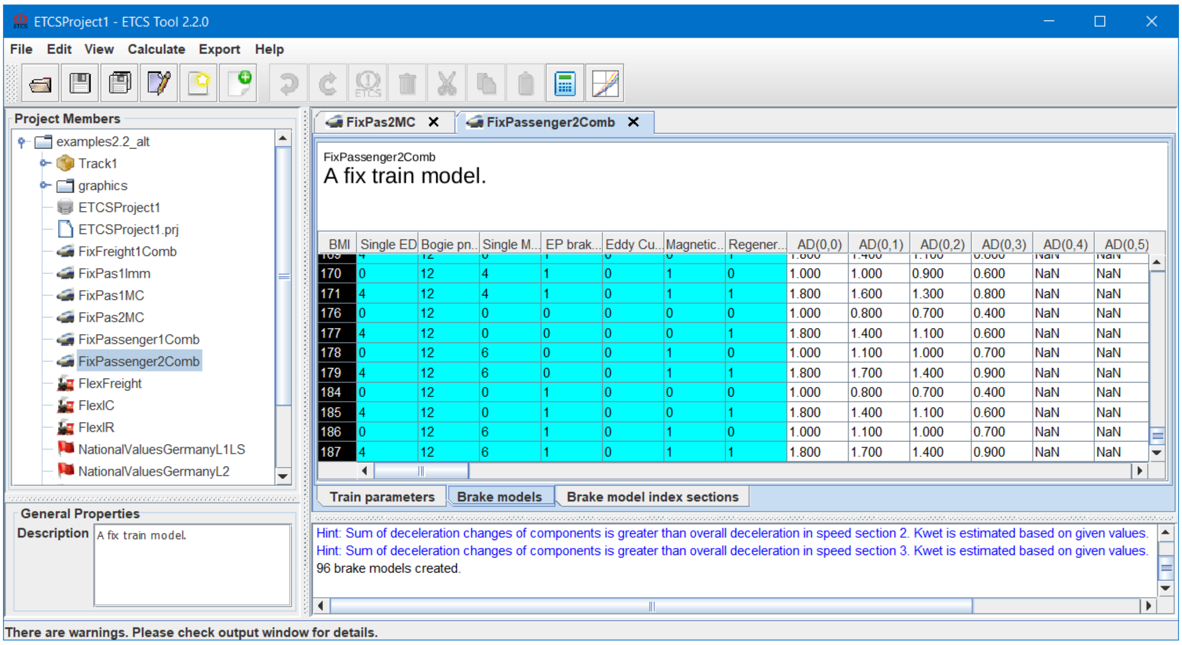


Figure 11: The brake models table

4.4 Calculation of Correction Factors for Fix Brake Models by Monte-Carlo Simulation

If all statistical parameters of a train's braking behaviour shall be modeled, it is not possible to calculate the correction factors by combinatorial logic anymore. In that case, a Monte-Carlo simulation is necessary. ETCS Brake Model Tool fully supports all requirements of [prEN 17997-1] plus some enhancements. A new model for Monte-Carlo simulation is created by **File – Create New Member** and selecting "Fix Train Model by Monte Carlo Simulation". All properties of the *fix train model by Monte Carlo simulation* are stored in the train model file with extension **.fixmc**. The simulation results and the final models will be stored in this file as well.

The input screen consists of three sections as shown in figure 12.

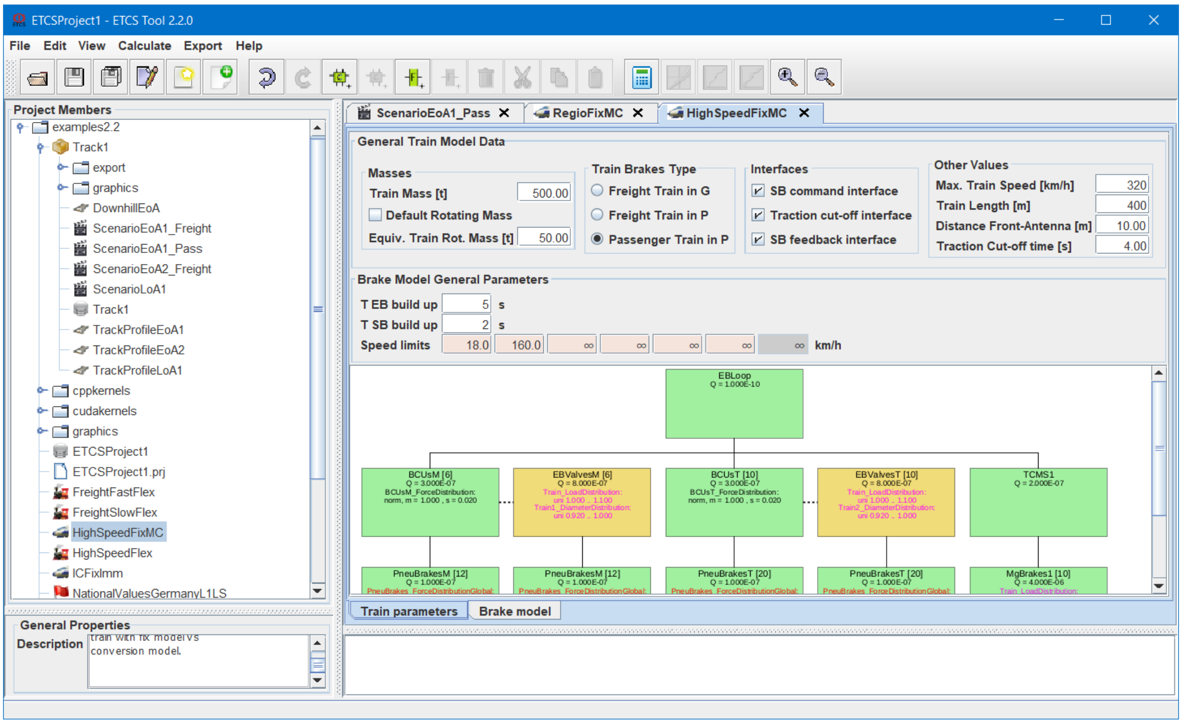


Figure 12: The editor to define the architecture of the brake system

Before starting to define any parameters, define a nominal train state with respect to

- load
- wheel diameter
- any other internal or external parameter that effects the brake force and that you want to consider in the model (such as temperature, age of brake pads)

This nominal state is not universally defined, you can (and have to) define it by yourself. Typically, the nominal train state assumes minimal mass (empty train) and maximum wheel diameter (new wheels). This definition has to be considered when you want to define a parameter later on in order to model the deviation from this nominal state.

4.4.1 General Train Model Data

This data is necessary for both fix and flexible train models. It is described in section 3.1.1.

4.4.2 Brake Model General Parameters

Here you have to enter the brake build up times and the upper speed limits of each section. The last section has no upper limit, i. e. if you want to define two speed sections, you only have to enter one value.

4.4.3 Architecture Editor

The third section is the architecture editor. Here the logical structure of the brake system is defined in terms of a block diagram.

There are two basic kinds of brake components: *controls* and *actors*. Both *controls* and *actors* may be characterized by an unavailabily (i. e. the probability of failing when the next emergency brake is required by the ETCS on-board) and statistical deviations, see section 4.4.5. An *actor* in addition is defined by its brake force for each speed section and some additional parameters, see section 4.4.8. A *control* can control other *controls* and/or *actors*, whereas an *actor* cannot control anything.

Components are added by **Edit – Create Control** or **Edit – Create Actor**. Both an *actor* or a *control* can be added on top level or below a selected *control*. Thus, the lowest level in the architecture contains only *actors*. The solid lines between components are to be read as "controls" (from top to bottom) or as "is controlled by" (from bottom to top).

4.4.4 Redundancies in Brake Systems

Typically, there are two kinds of redundancies in train brake systems:

- Redundancy "by nature": Typically each waggon or coach of a train has its own brake system. Thus if one component of this waggon or coach fails, the others will still brake – there is just a loss of brake force in the train.
- Redundancy "by intention": Typically for control components whose failure would have a high impact on the overall train brake force some kind of replacement exists. This replacement might either be a 1-by-1 replacement of the nominal component, or it is a more simple component. For example, the nominal control component might perform a load compensation, whereas the redundant control commands a fix brake force (which might be higher or lower than the nominal brake force depending on the particular loading condition).

With *fix train model by Monte Carlo simulation* of ETCS Brake Model Tool you can model "redundancies by intention" as well – even if the replacement has different parameters compared to the nominal component.

A replacement is visualized by a background in orange colour. The replacement is controlled by the higher level control. I. e. the dotted line between the replacement and the nominal component is to be read as "replaces" (from right to left) or as "is replaced by" (from left to right). The replacement is always of same type (*control* or *actor*) as the nominal component, and the number of replacements is always identic to the number of nominal components.

If a *control* is simulated as 'failed', the components of the branch below this *control* are ignored, i. e. the forces of all actors directly or indirectly controlled by this *control* will not be considered. If there is a replacement defined, the replacement branch will be considered instead. A replacing *control* may control the identic components as the nominal *control* (this is quite typical), but the replacement branch might as well look totally different.

4.4.5 Parameters of a Control

The parameters of a *control* are entered in the properties panel, see figure 13.

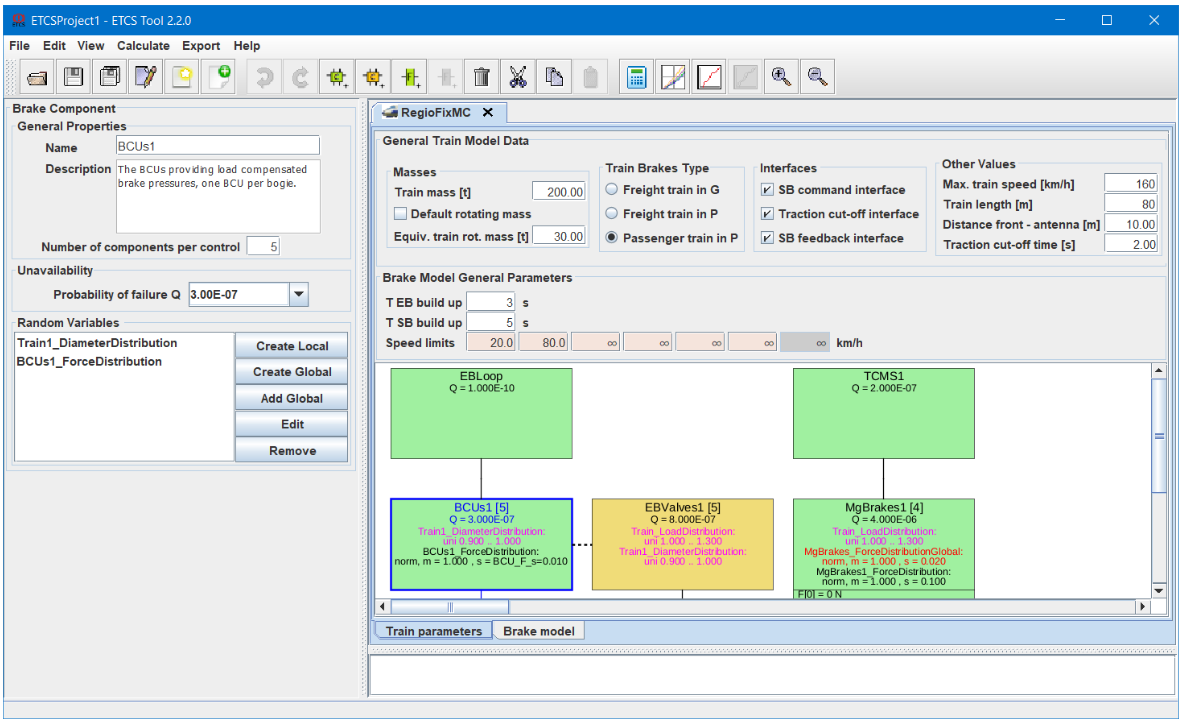


Figure 13: Parameters of a control block

General Properties

Name Each *control* needs a name. Blocks with identic names denote the identic component in the system, except of their manifold (see below).

Description An optional description of the component.

Number of components per control The brake system of a train typically consists of many similar components, e. g. two clamps per wheelset, two cylinders per bogie, two bogies per car, one WSP per car controlling 4 valves each. It would be annoying and error prone if someone would need to define each of these components separately. Thus you can define for each component, how many of these components are controlled by the same *control*. In the diagram this number is shown in square brackes beside the name of the component.

By **View – Expand Brake System Architecture** each component of the overall brake system is shown with a separate box, and its particular number (e. g. "[3/5]" means the third out of five identic components). The first component of each type is shown in the normal (dark) background, all similar components with lighter background.

Unavailability

The probability that the component will not perform its intended function when an emergency brake is commanded.

4.4.6 Random Variables

One of the benefits of *fix train model by Monte Carlo simulation* over *fix train model by combinatorial logic* is, that you can model the random characteristics of each component quite precisely. Each component may be characterised by multiple statistic parameters. By **Create Local** you define a random variable for each particular component of this kind. A dialog will open where you can select the distribution function and enter the relevant parameters, see figure 14.

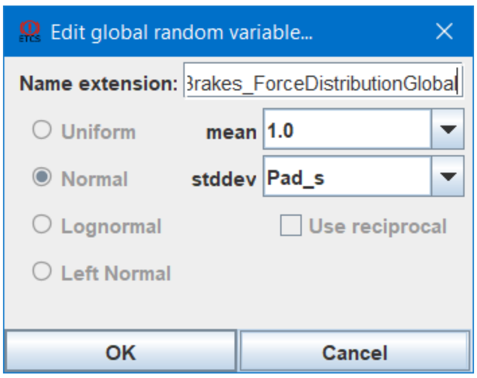


Figure 14: Statistical deviations parameters

Library values: Often the same value is used for many components. In order to simplify changing this value in all components, you can define a parameter holding the particular value. E. g. the value of the parameter "Pad_s" in figure 14 is defined in the *library* for easy adaptation.

Random parameters independent of brake components: In practise, there are random parameters that aren't induced by a technical characteristic of a brake system component. For example the wheel diameter is not related to any of the components of the brake system, but it affects the brake effort. Given that the two wheelsets of a bogie are supposed to always have the same diameter, but there is no relationship to the diameters of other wheelsets (re-profiling is performed bogie-wise), you can define a random variable on bogie level. If there is no *control* on bogie level, you can introduce a "virtual" *control* on this level, i. e. a "component" with no unavailability, just describing the wheel diameter with a random variable. Another (more difficult) option would be the use of global variables, see section 4.4.7.

Reciprocal of a random value: In case of a uniform distribution, you can select to use the reciprocal (see figure 15, i. e. to divide the brake force by the random value instead of multiplying.

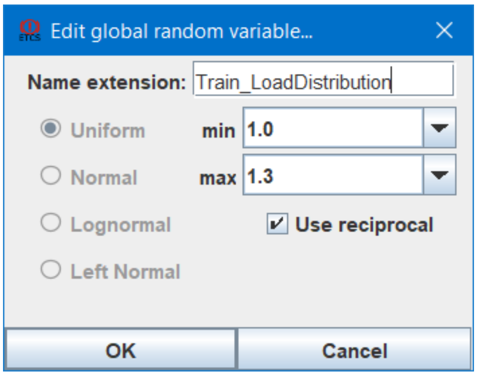


Figure 15: Parameters of a uniform distribution

This is in particular useful if you defined the nominal train status as train without load and maximum wheel diameter: In that case it is more intuitive to define the actual mass as a uniform distribution with $min = 1.0$ and $max = m_{max}/m_{min}$ and using the reciprocal instead of setting $max = 1.0$ and $min = m_{min}/m_{max}$. Similar for the wheel diameter: It is more catchy to define a uniform distribution with $min = d_{min}/d_{max}$ and $max = 1.0$ and using the reciprocal instead of setting $min = 1.0$ and $max = d_{max}/d_{min}$. [2]

4.4.7 Global Random Variables

It is possible to define random variables outside components. These random variables are called global random variable. A global random variable can be created in any component by **Create Global**. A global random variable can be used by any component by selecting **Add Global**. A dialog will occur showing all existing global random variables, see figure 16.

Assume for example a train consisting of three EMUs (of same or different type) coupled. Assume that all wheelsets of one EMU are typically re-profiled at the same time, thus the

[2] For the normal distribution (and variants of the normal distribution) the reciprocal doesn't make sense, both physically and mathematically (the maximum brake force could in principle increase to infinity).

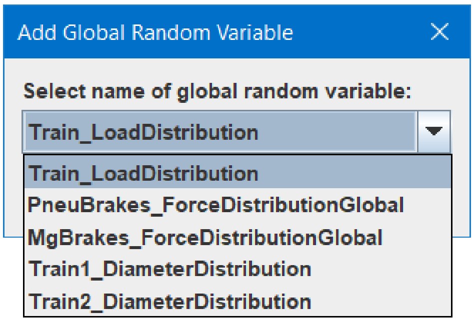


Figure 16: Global distribution selection

diameter of the wheelsets is a random variable on EMU level. The three EMUs within the train can be assumed to have different wheel diameters. Thus you can define a global variable "dw_1" for EMU 1, "dw_2" for EMU 2 and "dw_3" for EMU 3 and use these variables in the relevant *actors* of the three EMUs.

4.4.8 Parameters of an Actor

The parameters of an *actor* are entered in the properties panel, see figure 17.

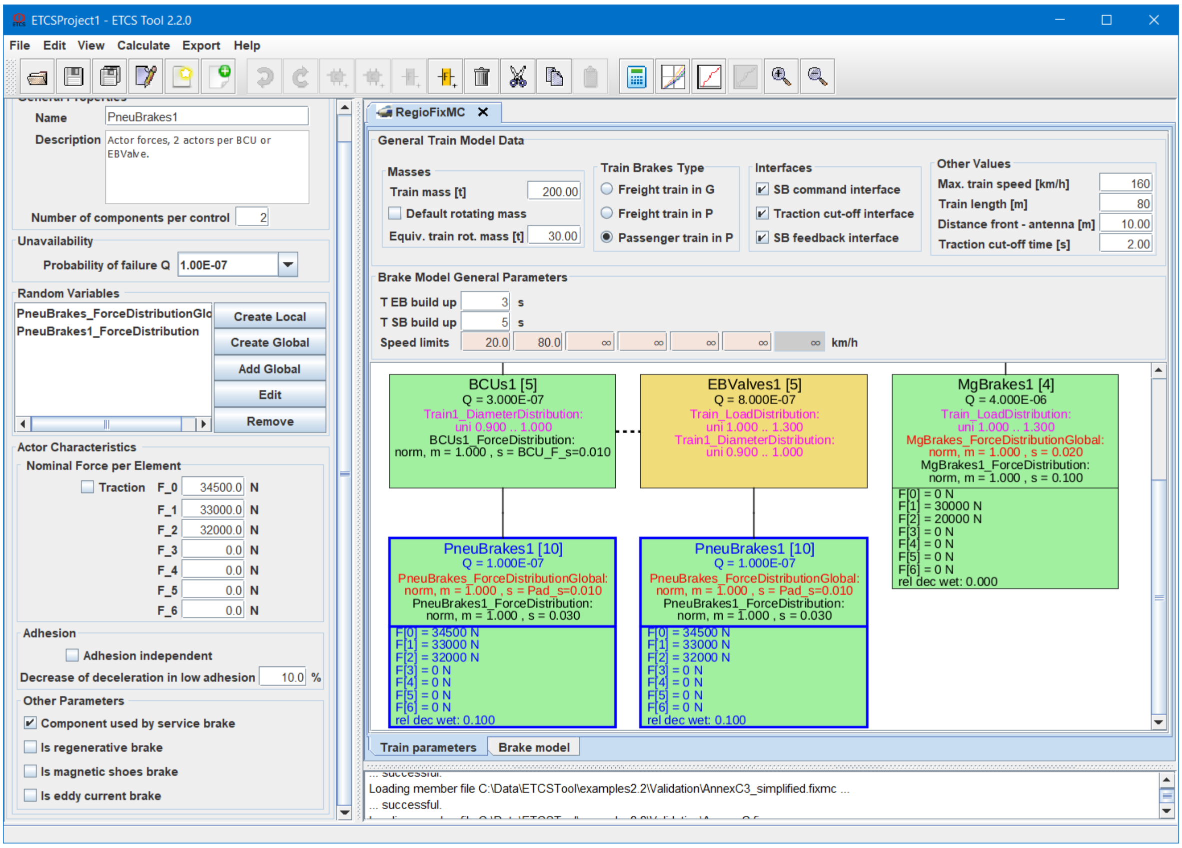


Figure 17: Parameters of an actor

An *actor* has all parameters of a *control* (see section 4.4.5), plus the following in addition:

Nominal Forces per Element: The deceleration forces for each speed section for the nominal train state (positive values).

In the rare case that you want to consider a failure of the traction system (applying positive traction effort by fault), you can select the "Traction" checkbox and enter the traction forces (positive values as well).

Adhesion: Select if the component is adhesion independent (e. g. magnetic track brakes or Eddy current brakes) or the decrease of deceleration in low adhesion conditions. See [prEN 17997-1] for more information.

Other parameters: Select the type of the actor and whether the actor shall be considered in the service brake model or not.

4.4.9 Performing the Simulation

In order to actually create ETCS brake models for commercial purpose, you'll need to buy a license for ETCS Brake Model Tool.

Simulations are performed on a GPU due to computational effort. In ETCS Brake Model Tool version 2.2 only GPUs by NVIDIA are supported.

When you select **Calculate – Calculate Brake Models** and any previously determined simulation results are still valid (e. g. if you only change some general train model values), you'll be asked whether you want to reuse the existing simulation results.

If there are no valid results or if you want to perform a simulation with more runs, the dialog shown in figure 18 will open.

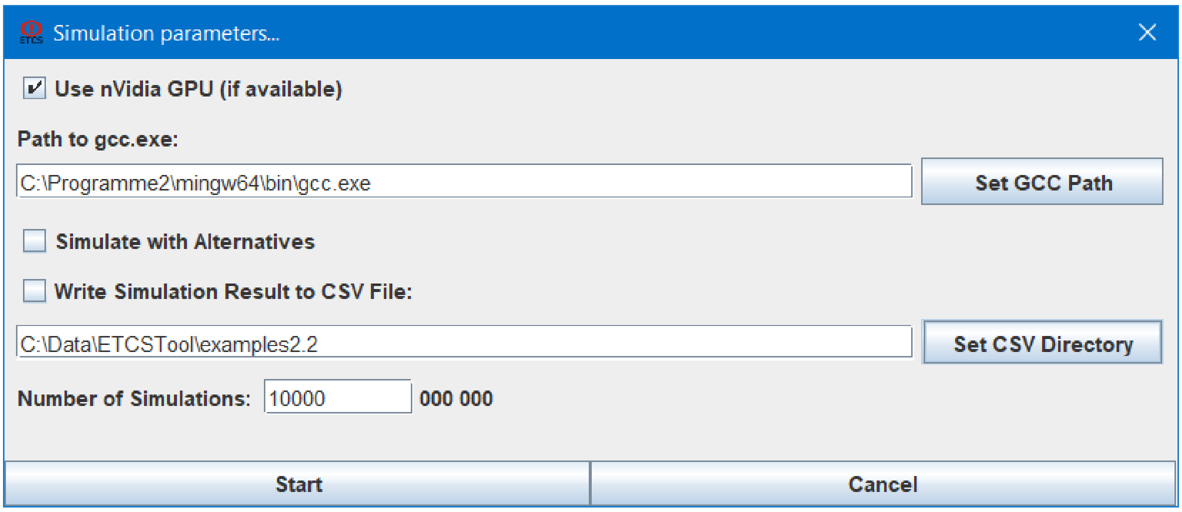


Figure 18: Simulation parameters

Simulation without GPU: If you have no NVIDIA GPU or no license for ETCS Brake Model Tool, you can nevertheless define a *fix train model by Monte Carlo simulation* and perform a Monte-Carlo simulation with 1 million simulation runs on the main CPU. [3]

In that case a variant of the (free) GNU Compiler Collection (gcc) must exist on your computer (no installation required). For Microsoft Windows see e. g. `www.mingw-w64.org`. The path to the C-Compiler has to be stated here.

Simulation with Alternatives: If you select **Simulate with Alternative** ETCS Brake Model Tool will perform the standard simulation plus an additional simulation for each alternative parameter stated in the *library*, see section 2.4. The overall simulation time will extend accordingly. Thus it is typically useful and sufficient to reduce the number of simulations to e. g. 1 billion instead of 10 billion. The results are presented in the k_dry or A_dry chart, see section 4.4.10.

Progress Monitor: The progress of the simulation will be shown by a progress monitor, see figure 19. You can abort the simulation at any time. In that case any previous or intermediate results are lost.

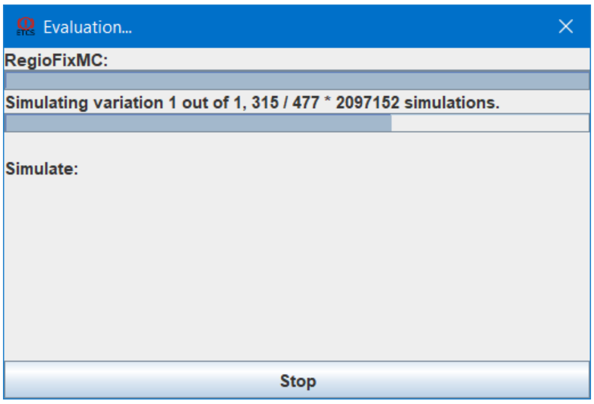


Figure 19: The progress monitor

Simulation results: In general, maximum eight brake models are required for a given trainset and brake system status – one for each combination of the three special brakes that can be de-activated by trackside. Since the simulation for de-activated special brakes only slightly increases the overall computational effort, these simulations will be performed as well.

Based on the simulation results the correction factors k_dry are determined considered a statistical confidence of 95% in accordance to [prEN 17997-1]. The ETCS brake models are presented in the *Brake Models* tab, see figure 20. Depending on the actually existing special brakes, two, four or even all eight brake models will be identic.

[3]This is typically sufficient for test purposes and most non-commercial applications but not for determination of correction factors for EBCL>5.

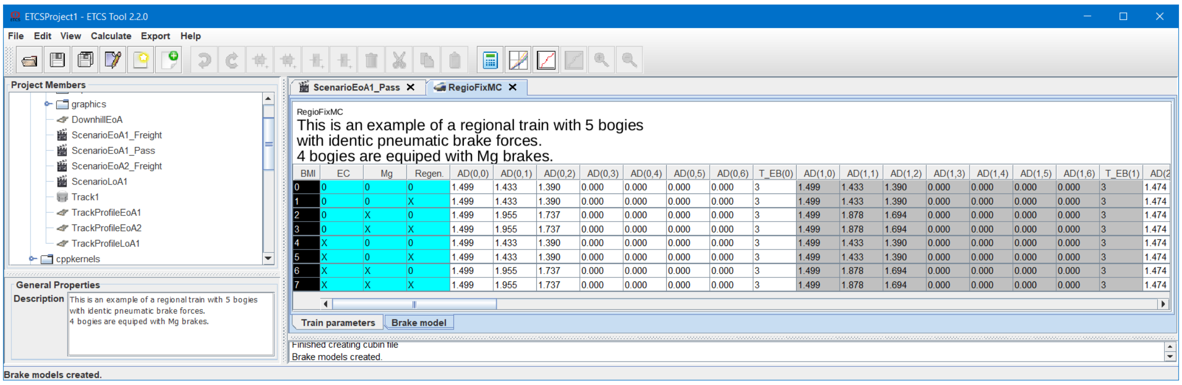


Figure 20: The brake models table

4.4.10 k_dry and A_dry Chart

The simulation results can be presented graphically, see figure 21. The statistical confidence of 95% defined in [prEN 17997-1] is applied herefore as well.

You can select which of the eight brake models (i. e. combination of special brakes) shall be shown, and whether the correction factors k_dry or the resulting safe deceleration a_dry shall be shown.

If the latest simulation has included alternative parameters (see section 4.4.9), you can select the curves for alternative parameters as well.

The graphics can be exported as `.png` or `.svg` file.

In the text field below the graphics, you can specify a probability and you'll get the exact correction factors and decelerations for this probability.

4.5 The Emergency Brake Decelerations Chart Window

After having calculated the brake model parameters, you can calculate brake distances as a function of the initial speed by **Calculate – Show Distances by EBCL Chart**. A separate window will open, see figure 22.

The brake distances for each *brake model* (BMI) and each EBCL can be visualized. The *brake model* for which the distances shall be shown is selected by its *brake model index*. The distances for different EBCL will be marked by different color.

Here, no track profiles or other data are considered. For calculating the brake curves according to [Subset 026-3] see section 8.

All axis can be scaled and zoomed.

The presented graphics can be exported to a vector graphic (**.svg**) or a bitmap (**.png**) file, select **File – Export ...** in the menu of the chart. Note that in vector graphics format,

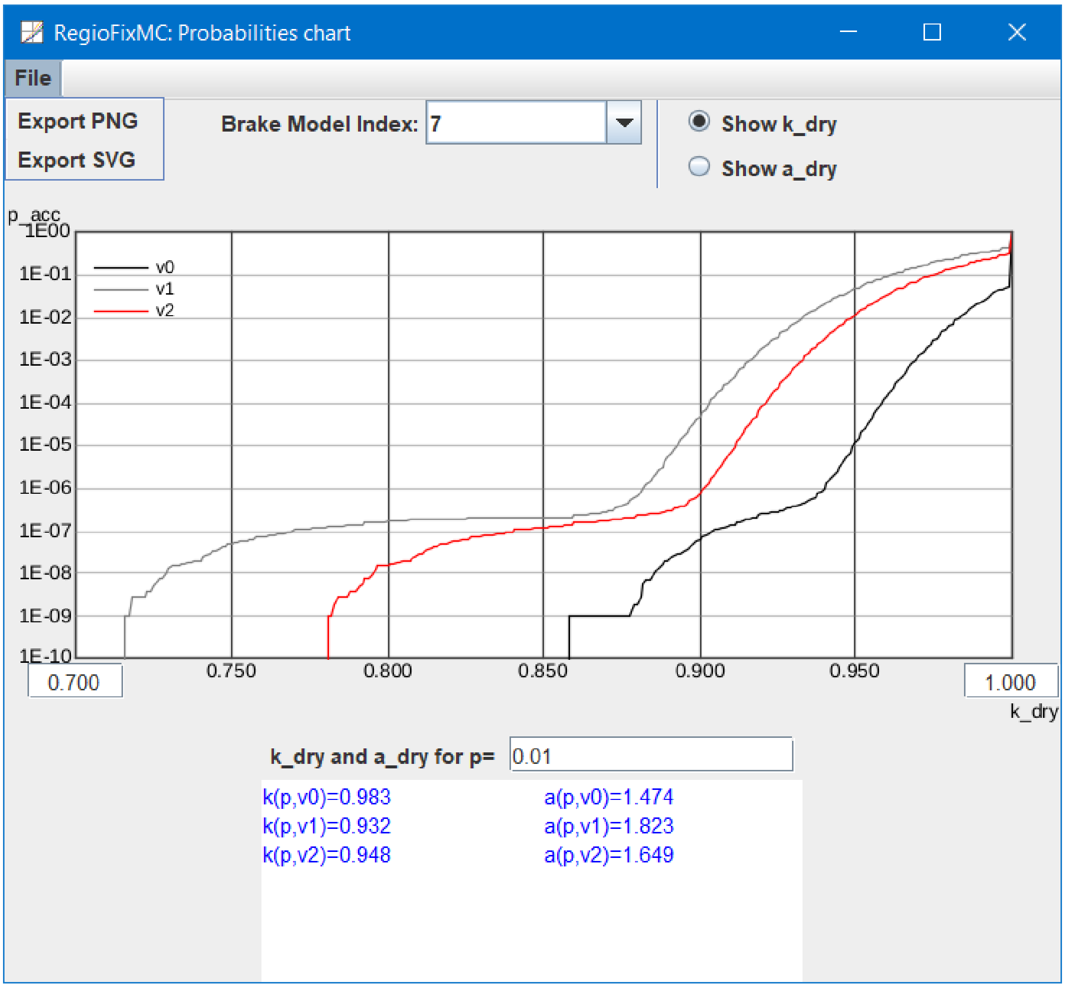


Figure 21: The chart presenting the dependency of the probability not to achieve the modeled deceleration as function of k_dry (or A_dry)

the graph data is exported with original resolution, so a later printout will have a very high quality (if not reduced by the later processing).

4.6 Exports

By **Export – Export Brake Models** all *brake models* of the fix train model will be saved in an CSV file with extension **.csv**. Depending on the *result mode* either absolute values are written for all $EBCL$'s, or for $EBCL > 0$ the correction factors will be written.

The file has the same name as the train model file, it is located in a subdirectory named 'export'.

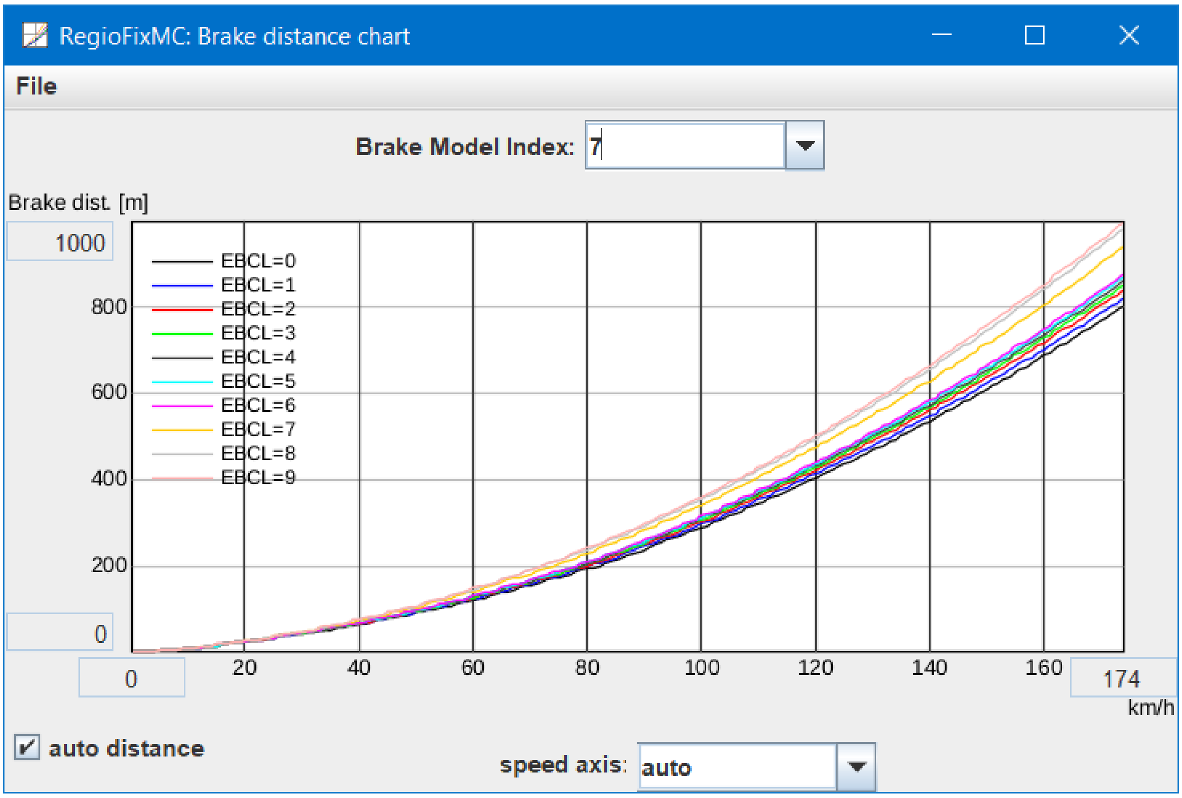


Figure 22: The emergency brake decelerations chart window

5 Normal Service Brake Model

The normal service brake model describes the "guidance curve".

5.1 The Normal Service Brake Model Panel

The data that is necessary for the calculation of the "guidance curve" is entered in the *normal service brake model panel*, see figure 23.

All properties of the *normal service brake model* are stored in the normal service brake model file (extension `.nsbm`). A *normal service brake model* that has not been saved after the latest modification is marked with an asterisk '*' in its title in the tab pane.

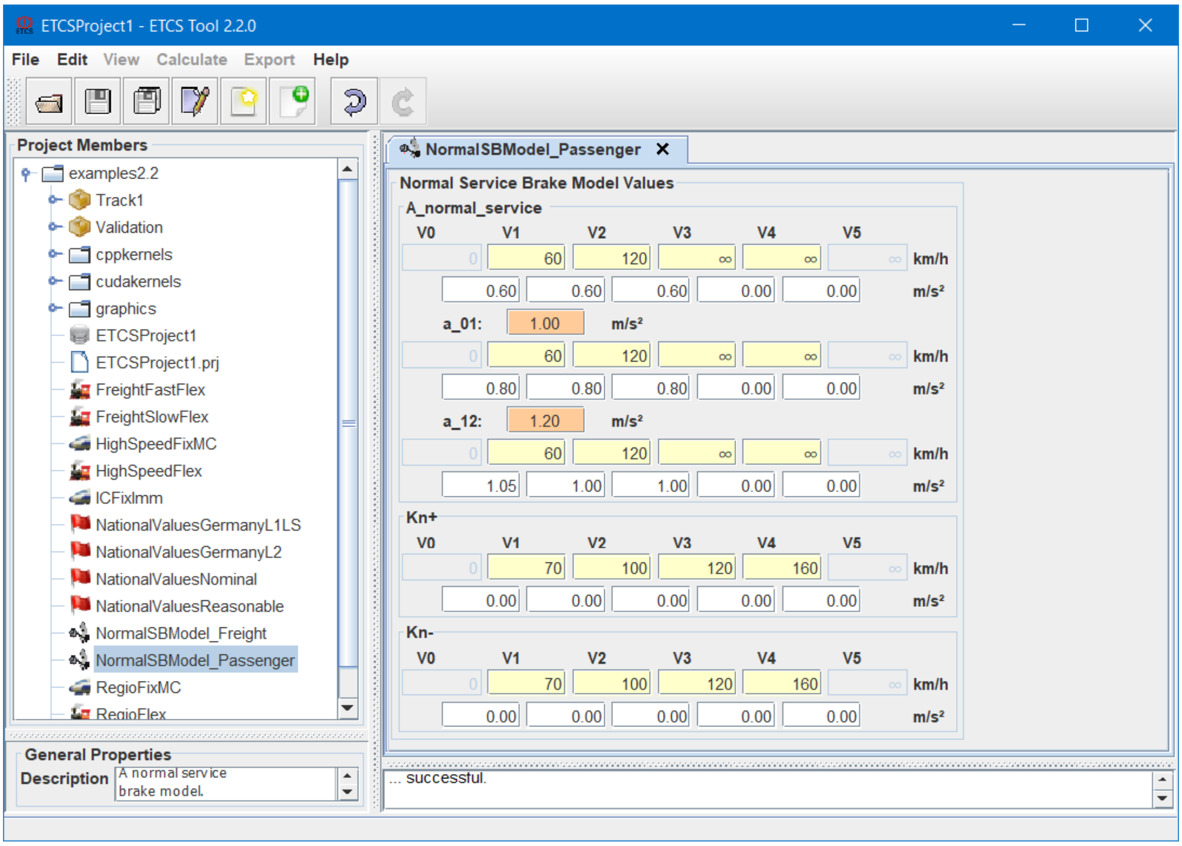


Figure 23: The normal service brake model panel

A_normal_service

The upper speed limits of up to five speed sections in km/h and the related deceleration values in m/s^2 (positive values). The last speed section has no upper speed limit. If you need less than five sections, type 'i' into the speed limit text field(s) that you want to set to "infinity" (see figure 23).

The values a_01 and a_12 in combination with the service brake deceleration used for the calculation of SBI_1 will determine which of the three sets of A_normal_service data will be used, see [Subset 026-3].

Kn+ and Kn-

Typically you won't use the actual service brake performance in the normal service brake model, but only some part of it, let's say 50%. Thus in case of slopes, the resulting "normal service brake model" will not reflect real operation anymore, because in real operation, the driver will apply more than 50% of the service brake effort when running downhill and less than 50% if running uphill. This can be adjusted by setting Kn+ and Kn- to values greater than $0\,m/s^2$. In particular, setting Kn+ to $9.81 m/s^2$ will result in no effect of uphill gradients, setting Kn- to $9.81 m/s^2$ will result in no effect of downhill gradients.

6 National Values

The *set of national values* to be used for calculation of a scenario are entered via the *national values panel*. Please see [Subset 026-3] for the exact meaning of each value.

The *set of national values* is stored in the national values file (extension `.nv`). A *set of national values* that has not been saved after the latest modification is marked with an asterisk '*' in its title in the tab pane.

6.1 The National Values Panel

All national values are entered via the *national values panel*, shown in figure 24.

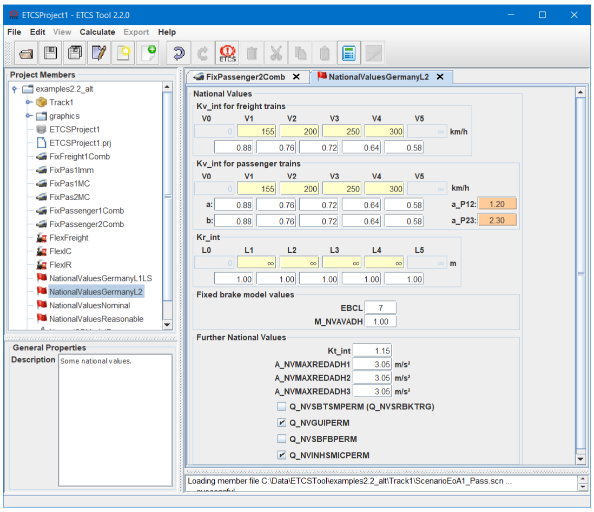


Figure 24: The national values panel

The sections "Kv_int for freight trains", "Kv_int for passenger trains" and "Kr_int" contain the national values necessary for scenarios using the *flexible train model* (conversion model), see section 3. If you need less than five speed or length sections, type 'i' into the speed or length limit text field(s) that you want to set to "infinity" (see figure 24).

The section "Fix brake model values" contains the values necessary for scenarios using the fix train model, see section 4.

The "Further National Values" are used by both types of train models (fix and flexible), except of Kt_int, which is only used by the *flexible train model*.

7 Track Profiles

The track profile describes all parameters of the track related to the braking distance. It also contains all relevant parameters of the movement authority (MA), such as EoA/LoA, SvL, target speed etc.

All locations stated in the track profile are positive values refering to a virtual zero location and increasing with the train running in nominal direction.

All properties of the *track profile* are stored in the track profile file (extension `.tp`). A *track profile* that has not been saved after the latest modification is marked with an asterisk '*' in its title in the tab pane.

7.1 The Track Profile Panel

All data is entered via the *track profile panel*, shown in figure 25.

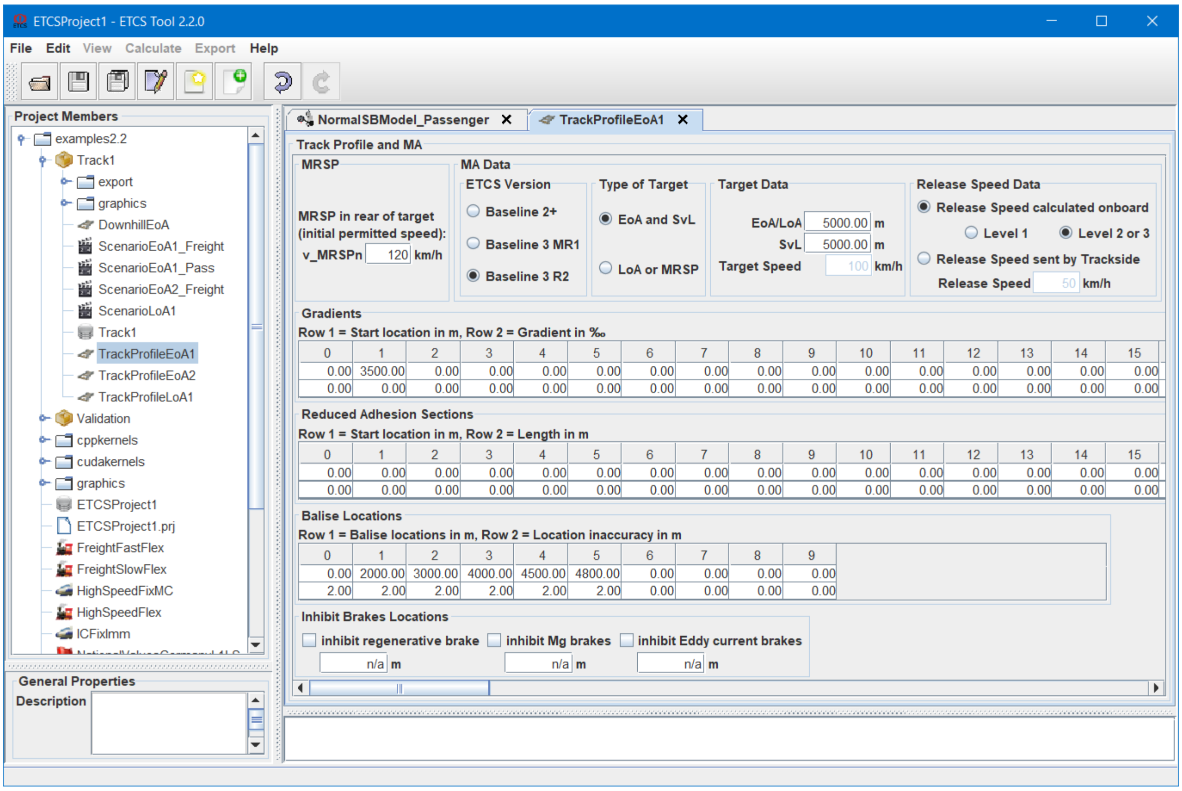


Figure 25: A track profile panel

7.1.1 MRSP

The only parameter in this section is the trackside speed restriction relevant in ceiling speed monitoring.

Note: The value is named "v_MRSPn" here in accordance to [Subset 026-3], even though the value does not need to consider the train related maximum speed. If the train related maximum speed stated in the *train model* used in a *scenario* is less than the value stated here, the train related maximum speed will be used.

7.1.2 MA Data

ETCS Version

The requirements regarding calculation of TSM related data has been changed between diffent ETCS versions (compare the different versions of Subset-026 chapter 3, i.e. versions 2.3.0d, 3.4.0, 3.6.0), therefore it is necessary to state the version of the trackside ETCS.

For "Baseline 2" it is presumed, that CR595 or [Subset 026-6] is implemented by the ETCS on-board, and that either packet 203 is sent by trackside and considered by the ETCS on-board or the "correction factors" are stored on-board. If this cannot be ensured, the results can only be a rough estimation of the behaviour of the train, since in "pure" Baseline 2 (version 2.0) no harmonized brake models exist.

Type of Target

Select whether the target is an "End of Authority" (EoA) or a "Limit of Authority" (LoA).

Target Data

In case of an EoA, the EoA location and the superised location (SvL) has to be entered.

In case of an LoA, the LoA location and the target speed has to be entered.

The locations must be positive values, with SvL greater or equal to EoA.

The distance from zero to the stated EoA location must be at least as large as the indication curve distance or guidance curve distance for the stated v_MRSPn or the maximum train speed. If this is not fulfilled, calculation of the *scenario* will fail.

Release Speed Data

If the MA requires to calculate the release speed (RS) on-board, select the ETCS level related to the MA.

If the MA requires to use a fix release speed (either "national value" or value given in MA), set the fix release speed value here.

Note: Typically the release speed should be calculated on-board. Using a fix release speed is typically either not safe or not beneficial from operational point of view (or even both).

7.1.3 Gradients

Enter the track gradients in the table. The start locations shall be strictly increasing. The first start location that is less then the previous determines the end of the profile, i. e. this value and all following values won't be considered.

7.1.4 Reduced Adhesion Sections

Enter the sections with reduced adhesion in the table. The start locations shall be strictly increasing. The second line states the length of each section. The sections should not overlap.

7.1.5 Balise Locations

Enter the locations of balises that can be used for re-location up to the EoA. The second line states the location inaccuracy for each balise.

7.1.6 Inhibit Brakes Locations

If some special brake shall be inhibited, select the checkbox and enter the correspondent location.

Note: It is assumed that it does't make sense to re-enable a special brake while braking to a target, therefore the behaviour of the train with respect to (re-)enabling a special brake is not modelled in the *train model*, and thus you cannot enter a "special brake profile" here.

8 Scenarios

A *scenario* combines a *train model*, a *set of national values*, a *track profile* and optionally a *normal service brake model* in order to determine braking distances and supervision limits. It also includes some additional parameters, see below.

8.1 The Scenario Panel

It is possible to compare two *scenarios* A and B, that may differ in any kind and number of parameters. All data of the *scenarios* A and B are stored in the scenario file (extension `.scn`). A scenario file that has not been saved after the latest modification is marked with an asterisk '*' in the title in the tab pane.

All data related to the *scenarios* A and B is entered in the upper part of the *scenario panel*, shown in figure 26.

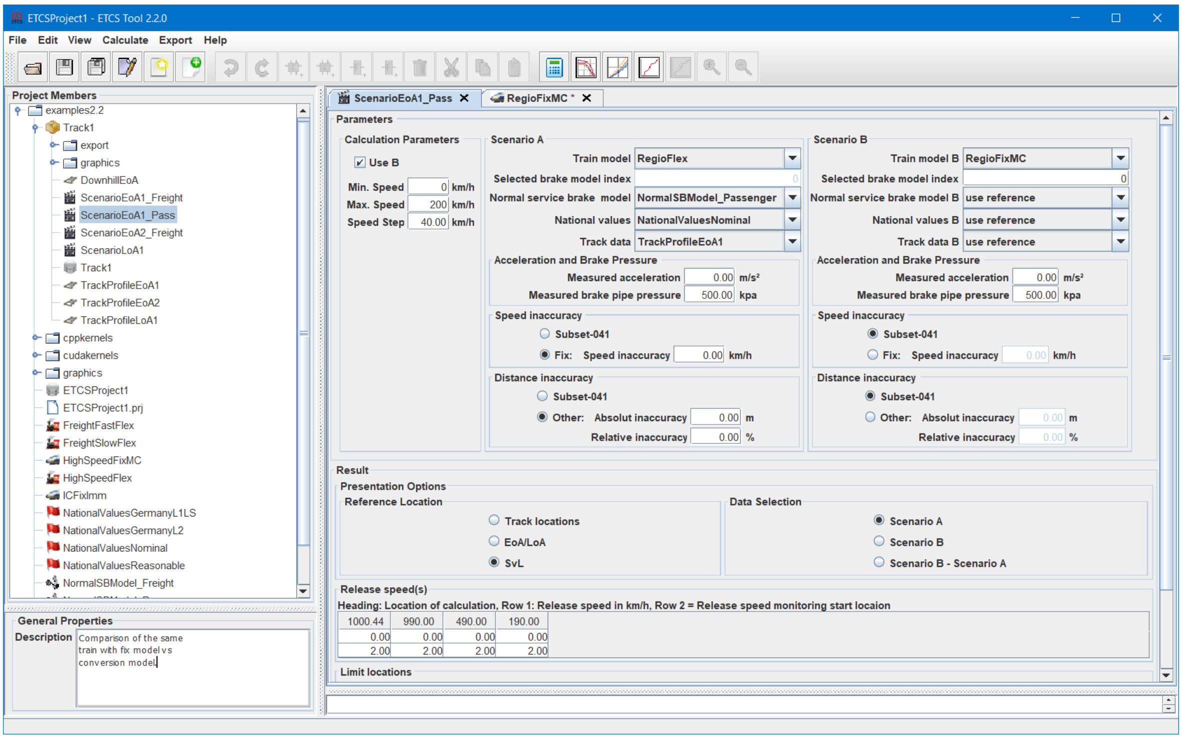


Figure 26: A scenario panel

The lower part of the *scenario panel* shows calculated results in two tables.

8.1.1 Parameters

This section shows all parameters that are necessary to evaluate a scenario or to compare two scenarios, which do not belong to a *train model*, a *normal service brake model*, a *set of national values* or a *track profile*.

Scenario A

Select a *train model*, a *set of national values* and a *track profile* out of the available *project members*. If you want to calculate a guidance curve, select a *normal service brake model* as well. Remember that you can select only *project members* in the same *package* as the scenario or in the global package.

If you select a fix train model, you'll have to select one of the *brake models* by its BMI. If you don't state a valid BMI, the *brake model* with the highest BMI (i. e. the master brake model) will be selected automatically.

Acceleration and brake pressure

According to [Subset 026-3] the measured acceleration will be used for determination of the EBI. You can enter a value greater than zero if this seems probable in reality.

The brake pressure will be used to adapt the service brake build-up time, but only if the brake feedback interface is available according to the *train model* and allowed according to the national value Q_NVSBFBPERM.

Speed inaccuracy

Define which speed under-reading amount shall be assumed in calculations. If "Subset-041" is selected, the under-reading amount is assumed to be the allowed value according to [Subset 041], 5.3.1.2. If "Fix" is selected, the value stated here is used for any speed.

Distance inaccuracy

Define which distance under-reading and over-reading amount shall be assumed in calculations. If "Subset-041" is selected, the absolute and relative under-reading or over-reading amount are assumed to be the allowed values according to [Subset 041], 5.3.1.1. If "Other" is selected, you can state another absolute and relative inaccuracy.

The relative inaccuracy always relates to the distance from the LRBG. Therefore the balise group locations given in the *track profile* have significant impact on the release speed(s) (if calculated on-board) and the limits typically.

Scenario B

If you want to compare two scenarios, select the related parameters as for scenario A.

Calculation parameters

These parameters only affect the "Limit locations" table in the lower part of the scenario panel. They have no impact on the limits chart described in section 8.2.

Select the minimum and maximum train running speed for whose to calculate limits, and the speed step size.

8.1.2 Result

This section will show the results calculated after pressing **Calculate – Calculate Scenario Limits** or the corresponding button.

Reference Location

The reference location of the results can be selected to be

- the absolute track locations (i. e. same coordinate system as the *track profile*),
- the distances to the EoA or LoA (positive values in rear of the EoA/LoA, negative values in advance of the EoA/LoA,
- the distances to the SvL (always positive values).

Data Selection

The release speed data and the locations can either be shown for scenario A, scenario B or the difference between B and A.

Release speed(s)

If the release speed is determined by trackside, the release speed monitoring (RSM) start location will be calculated and indicated here.

If the release speed is calculated on-board, a release speed will be calculated at the perturbation location and at each further balise group. All release speeds and the corresponding start locations will be shown here.

Limit locations

The calculated limits according to the reference scenario and the calculation parameters (see above).

8.2 Limits Chart Window

If you press **Calculate – Show limits chart** or the corresponding button, a chart window will open, showing the limits graphically.

All axis can be scaled and zoomed.

The presented graphics can be exported to a vector graphic (`.svg`) or a bitmap (`.png`) file, select **File – Export ...** in the menu of the chart. Note that in vector graphics format, the graph data is exported with original resolution, so a later printout will have a very high quality (if not reduced by the later processing).

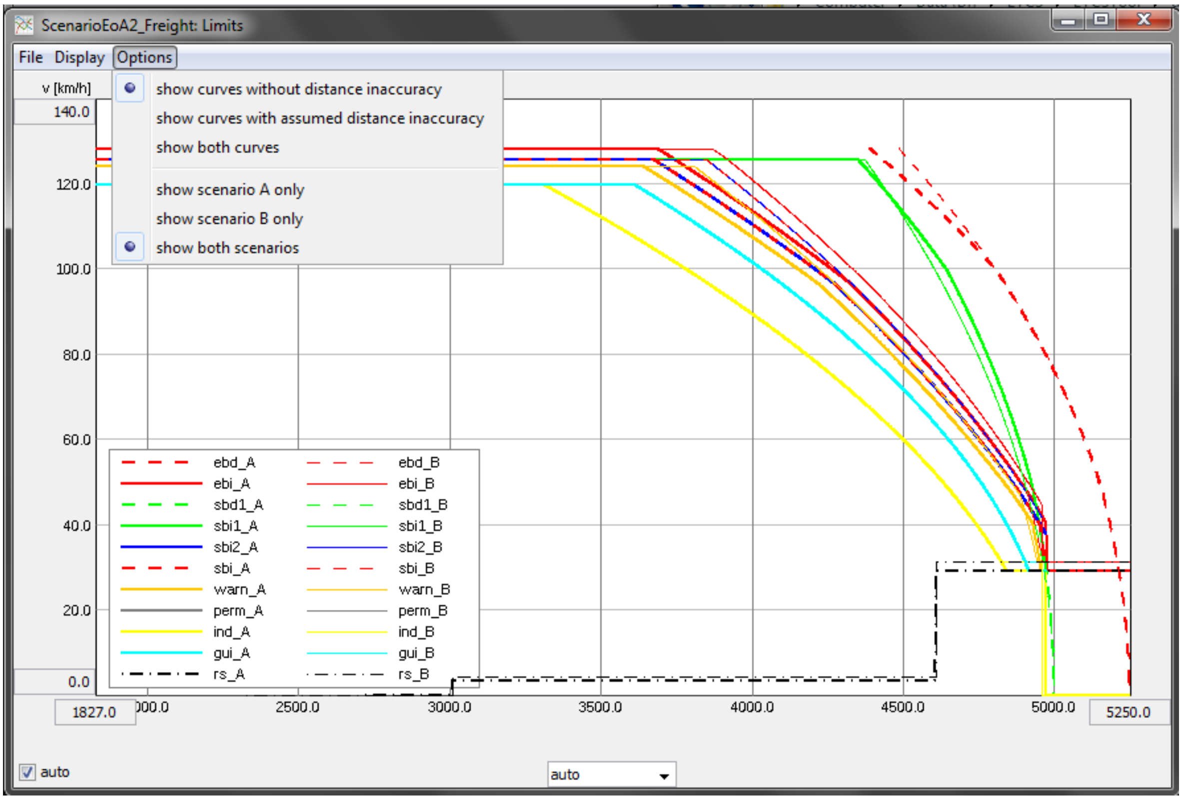


Figure 27: A limits chart window

9 Menus and Commands

9.1 Generals

The descriptions of models or events are immediately changed whenever you type a key in the description field. All other numerical or text values are changed after pressing 'Enter' only.

Note that not all commands or properties will be available in a specific situation. Typically only those possibilities, that make sense and result in a valid model are offered. In the case that a error occurs when executing a command, an error message is displayed in the status bar or in the message window.

9.2 The File Menu

This menu contains all commands related to the *project*, its models and libraries. Most commands are also available in pop-up menus that open when pressing the right mouse button in the *project members tree*.

9.2.1 New Project

If there is an open *project* this will be closed. If necessary you are asked to save data. After that a dialog will appear where you are asked for a name of the new *project*. Finally an empty *project* will be created.

9.2.2 Open Project

If there is an open *project* this will be closed. If necessary you are asked to save data. After that a dialog will appear where you can select the *project* to be opened. All libraries and models referred in the project file will be opened and indicated in the 'Project Members' tree.

9.2.3 Close Project

The current *project* is closed. If necessary you are asked to save changes in the *project*, library or models.

9.2.4 Project Properties

A *project properties dialog* window will open where you can set the project properties. Refer to section 2.5 for details.

9.2.5 Create new Package

A new package is created by **File – Create new Package**. You will be asked for the name of the new package. A sub-directory with the given name will be created in the project directory, and the *local library* file will be created.

9.2.6 Import Package

Not yet implemented.

9.2.7 Create new Member

A new *project member* is created by **File – Create new Member**. The *Create New Member Dialog* will open, where you can select the *package* the new member shall belong to, and the name and type of the new member, see figure 28.

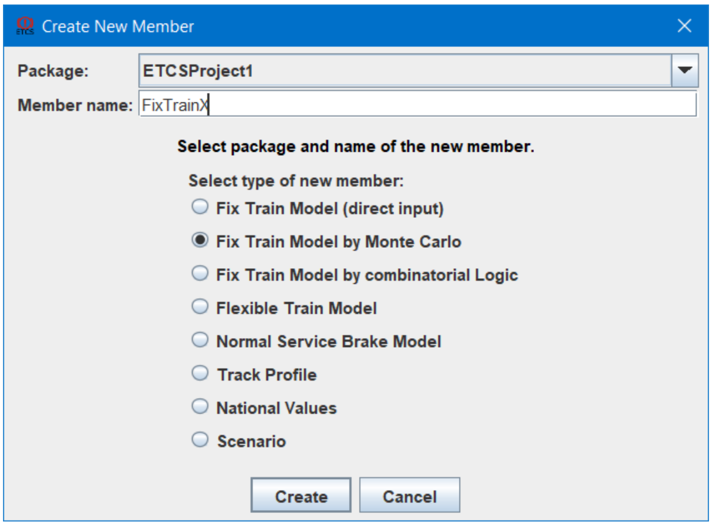


Figure 28: The create new member dialog

9.2.8 Add existing Member File

When selecting **File – Add existing Member File**, a dialog will appear where you can select the file to be added to the *project*, see figure 29.

Per default, only files with extension `.ignore` will be displayed. Anyhow you can also select other files, including files already belonging to another *package* of the *project*.

You can select the *package* to which the member shall be added, and enter a name for it.

9.2.9 Remove active Member

The reference to the member presented in the active tab will be removed from the *project*, the tab will be closed. If necessary you are asked to save data of this member.

9.2.10 Rename active Member

The *project member* presented in the active tab can be renamed by **File – Rename active Member**. A dialog will appear asking you for a new name.

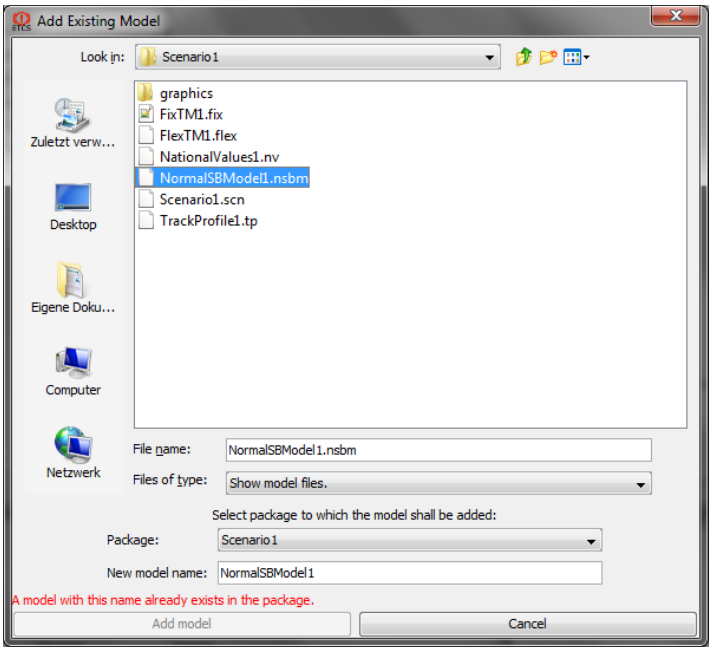


Figure 29: Add existing member dialog

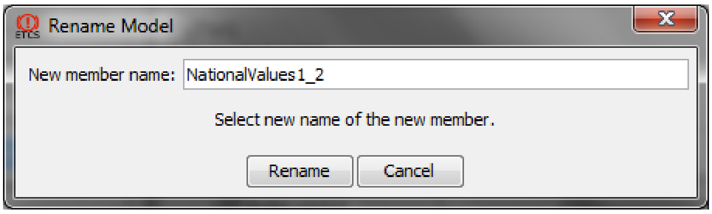


Figure 30: Rename member dialog

9.2.11 Move active Member

The *project member* presented in the active tab can be moved to another *package* by **File – Move active Member**. A dialog will appear asking you for a new package.

9.2.12 Duplicate active Member

The *project member* presented in the active tab can be duplicated by **File – Duplicate active Member**. A dialog will appear asking you for the *package* and the name of the duplicate.

9.2.13 Save active Member

Saves the *project member* currently displayed in the graphics tab. The file extension is automatically appended.

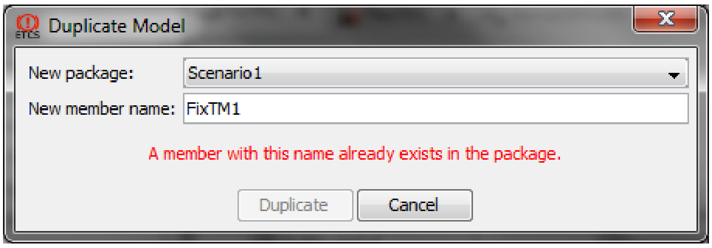


Figure 31: Duplicate member dialog

9.2.14 Save All

All *project members* and the *project* are saved if changed. Note that a project, that has not been saved after the latest modification, is marked with an asterisk '*' in the window title. An unsaved *project member* is marked with an asterisk '*' in the title of its graphics frame.

9.2.15 List of recently used Projects

A list of recently used projects is presented. Selecting one is similar to **File – Open Project**, only that no dialog will appear.

9.2.16 Exit

If necessary you are asked to save changes in the *project* or in *project members*. After that the application is terminated.

9.3 The Edit Menu

The **Edit** menu contains all commands related to changing the structure of a fix train model.

Most actions are also available as button in the menu bar. For some actions keyboard commands (short-cuts) exist, see the entries in the **Edit** menu.

9.3.1 Undo last Change

The last 10 actions can be withdrawn. Here an action can be either an *edit*-action as stated above or a change of some value. The tool-tip text always informs about the next action of the *undo*-action.

9.3.2 Redo last Undo

All undo actions can be withdrawn.

9.3.3 Edit Actions for Libraries

Create Parameter

Create a new alias for a value to be used in *fix train model by Monte Carlo simulation*. Each alias may be assigned an alternative value in addition, which will be used for Monte-Carlo simulation with alternative values, see section 4.4.9.

Delete

Delete a parameter. Only parameters currently not used by any *project member* can be deleted.

Cut

Remember a parameter in the background and delete it from this *library*. Only parameters currently not used by any *project member* can be cut.

Copy

Copy a parameter in order to add it to another *library*.

Paste

Paste a parameter from previously cut or copied from another *library*.

9.3.4 Edit Actions for Fix Train Models by direct Input

Add Brake Model

Create another brake model for the active fix train model. A dialog will open asking you for the brake model index (BMI) of the new brake model, see section 4.2.

Delete

The selected brake model is deleted.

Cut

The selected brake model is stored in the background so that it can be pasted with another *brake model index*. After that, it is deleted.

Copy

The selected brake model is stored in the background so that it can be pasted with another *brake model index*.

Paste

If a brake model has been cut or copied before, you'll be asked for the new *brake model index* of the stored brake model. If the *brake model index* is not already in use in this *fix train model (direct input)*, the stored brake model will be added to this train model.

9.3.5 Edit Actions for Fix Train Models by combinatorial Logic

Add Brake Failure Model

A *brake failure model* will be added to the *fix train model by combinatorial logic* and displayed in the brake components table. A unique name will be automatically assigned to the new *brake failure model*.

Delete

In case a *brake failure model* is selected, this *brake failure model* will be deleted.

Cut

The selected deleted *brake failure model* will be stored in the background so that it can be pasted at another position. After that, it will be deleted.

Copy

The selected *brake failure model* will be copied to a background memory. It can be pasted somewhere, see below.

Paste

If a *brake failure model* has been cut or copied before, it will be added to the active *fix train model by combinatorial logic* below the last *brake failure model*.

9.3.6 Edit actions for Fix Train Models by Monte Carlo Simulation

Create Actor

If a *brake component* of type **control** is selected, a *brake component* of type **actor** will be created below. If no *brake component* is selected, the new **actor** will be created on top level.

Create Control

If a *brake component* of type **control** is selected, a *brake component* of type **control** will be created below. If no *brake component* is selected, the new **control** will be created on top level.

Create replacing Brake Component

If a *brake component* is selected, another *brake component* of the same type will be created. It will be shown in orange color right beside the selected one.

Delete

In case a *brake component* is selected, this *brake component* will be deleted.

Delete Branch

In case a *brake component* is selected, this *brake component* and all *brake component* below will be deleted. If there is a redundant *brake component* defined for this *brake component*, this branch below will be deleted as well.

Cut

The selected *brake component* and the branch topped by this *brake component* will be stored in the background so that it can be pasted at another position. Then the selected *brake component* will be deleted. If a replacement *brake component* is defined, the replacement branch will be deleted as well.

Copy

The selected *brake component* and the branch topped by this *brake component* will be copied to a background memory. It can be pasted somewhere, see below.

Paste

If a *brake component* has been cut or copied before, it will be added to the active *fix train model by Monte Carlo simulation* below the selected *brake component* or – if no *brake component* is selected – as a top level component. Its name won't change, i. e. it is the same component as the original one.

Paste duplicate Component

If a *brake component* has been cut or copied before, a duplicate of the *brake component* will be added to the active *fix train model by Monte Carlo simulation* below the selected *brake component* or – if no *brake component* is selected – as a top level component. The name of the duplicate is extended to a unique name.

Paste duplicate Branch

If a *brake component* has been cut or copied before, a duplicate of the *brake component*, all lower *brake component* and any replacing *brake component* will be added to the active *fix train model by Monte Carlo simulation* below the selected *brake component* or – if no *brake* *component* is selected – as a top level component. You will be asked by which extension the names of the duplicated *brake component* shall be extended.

Paste as replacing Component

If a *brake component* has been cut or copied before, it will be added to the active *fix train model by Monte Carlo simulation* as a replacement of the selected *brake component*. Its name won't change, i. e. it is the same component as the original one. Note that the types of the selected *brake component* and the one to be pasted must be identic.

Paste Duplicate as replacing Component

If a *brake component* has been cut or copied before, a duplicate of the *brake component* will be added to the active *fix train model by Monte Carlo simulation* as a replacement of the selected *brake component*. The name of the duplicate is extended to a unique name. Note that the types of the selected *brake component* and the one to be pasted must be identic.

Move left / move right

Move the selected branch to the left or the right. Note that this operation affects the visualisation only, it has no effect on the model.

9.4 The View Menu

9.4.1 Expand Brake System Architecture

By default, the number of similar *brake component* is indicated by a number in rectangular brackets beside the name of the *brake component*, e. g. [5]. If you click the *Expand Brake System Architecture* option, each *brake component* of the active *fix train model by Monte Carlo simulation* will be shown as a separate box. The first entity of each type of *brake component* is presented with the standard background color, any other entity with a brighter background color.

9.4.2 Show Correction Factors

By default, for fix train models the safe decelerations will be shown for each EBCL instead of the correction factors. If you want to see the correction factors instead, select this option.

9.4.3 Show Absolute Values

By default, for fix train models the safe decelerations will be shown for each EBCL instead of the correction factors. If you have switched to show the correction factors, you can switch back to absolute values by this option.

9.4.4 Zoom

You can change the zoom for the architecture of the *fix train model by Monte Carlo simulation* brake architecture view.

9.5 The Calculate Menu

9.5.1 Validate Brake Models

The manually entered brake model parameters of a *fix train model (direct input)* are validated. Basically, some limits are checked.

9.5.2 Calculate Brake Models

In case of a *fix train model by combinatorial logic*, all nominal brake models and all corresponding safe brake models (i. e. correction factors) are calculated.

In case of a *fix train model by Monte Carlo simulation*, the Monte-Carlo simulation is performed and the eight ETCS brake models are created, see section 4.4.9.

9.5.3 Calculate Scenario Limits

The limits related to a *scenario* are calculated and shown in the result's table in the lower part of the panel.

This operation is only available, if a *scenario* is the active member.

9.5.4 Show Limits Chart

A chart window will open, showing the limits for the *scenario* as a function of location, see section 8.2.

This operation is only available, if a *scenario* is the active member.

9.5.5 Show Brake Distances by EBCL Chart

A chart window will open, showing the brake distances as a function of speed and EBCL, see section 4.5.

This operation is only available, if a fix train model is the active member and the brake models have already been calculated.

9.5.6 Show k_dry or a_dry Chart / with alternatives

A chart showing the probability of exceeding the calculated braking distance as a function of k_dry or a_dry is opened, see section 4.4.10.

This operation is only available, if a *fix train model by Monte Carlo simulation* is the active member and there are valid simulation results.

9.6 The Export Menu

9.6.1 Export Brake Model(s)

Saves all *brake models* of the active fix train model in a text file with extension `.csv` in the `export` sub-directory of the directory of the *train model*.

This operation is only available, if a fix train model is the active member and the brake models have been calculated or validated already.

9.6.2 Export Limits

The values shown in the limits table (and some more) will be written to a `.csv` file. The file will be written to the `export` sub-directory of the directory of the *scenario*.

This operation is only available, if a *scenario* is the active member and the limits have been calculated already.

9.6.3 Export Graphics

The brake system architecture or the *fix train model by Monte Carlo simulation* can be exported as a .png file.

9.7 The Help Menu

9.7.1 Help

The content of this document is presented in HTML format.

9.7.2 Set License File

Specify the path to the license file here.

Note: You must restart ETCS Brake Model Tool to load the new license file.

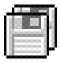
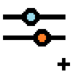
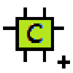
9.7.3 About

A window opens, indicating the version of ETCS Brake Model Tool and some parameters of the license.

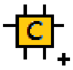
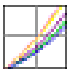
9.8 The Tool Bar

All frequently used commands are also available as buttons in the tool bar. The tool bar is context sensitive.

Table 4: Toolbar buttons

Icon	Command
	Close project and open another project
	Save active model
	Save all
	Project properties
	Create new member
	Add existing file to project
	Undo last change
	Redo last undo
	*fix train model (direct input)*: Add another brake model *fix train model by combinatorial logic*: Add brake failure model
	*fix train model (direct input)*: Delete brake model *fix train model by combinatorial logic*: Delete *brake failure model* *fix train model by Monte Carlo simulation*: Delete *brake component*
	*fix train model (direct input)*: Cut brake model *fix train model by combinatorial logic*: Cut *brake failure model* *fix train model by Monte Carlo simulation*: Cut *brake component*
	*fix train model (direct input)*: Copy brake model *fix train model by combinatorial logic*: Copy *brake failure model* *fix train model by Monte Carlo simulation*: Copy *brake component*
	*fix train model (direct input)*: Paste brake model *fix train model by combinatorial logic*: Paste *brake failure model* *fix train model by Monte Carlo simulation*: Paste *brake component*
	Create a parameter to be used for Monte Carlo simulation
	*fix train model by Monte Carlo simulation*: Add a control component

Continued on next page

Icon	Command
	*fix train model by Monte Carlo simulation*: Add a control component replacing another control
	*fix train model by Monte Carlo simulation*: Add an actor replacing another actor
	*fix train model by Monte Carlo simulation*: Add an actor
	Calculate scenario or a fix train model
	fix train model: Show brake distances by EBCL chart
	*scenario*: Show scenario limits chart
	*fix train model by Monte Carlo simulation*: Show k_dry or A_dry chart with alternatives
	*fix train model by Monte Carlo simulation*: Show k_dry or A_dry chart
	*fix train model by Monte Carlo simulation*: Increase the size of the architecture diagram
	*fix train model by Monte Carlo simulation*: Reduce the size of the architecture diagram

References

[Subset 026] : *ERTMS/ETCS System Requirements Specification*, ERA, UNISIG, EEIG ERTMS USERS GROUP (Issue 3.6.0)

[Subset 026-3] : *ERTMS/ETCS System Requirements Specification, Chapter 3: Principles*, ERA, UNISIG, EEIG ERTMS USERS GROUP (Issue 3.6.0)

[Subset 026-6] : *ERTMS/ETCS System Requirements Specification, Chapter 6: Management of older System Versions*, ERA, UNISIG, EEIG ERTMS USERS GROUP (Issue 3.6.0)

[Subset 026-7] : *ERTMS/ETCS System Requirements Specification, Chapter 7: ERTMS/ETCS language*, ERA, UNISIG, EEIG ERTMS USERS GROUP (Issue 3.6.0)

[Subset 034] : *ERTMS/ETCS Train Interface Functional Interface Specification*, ERA, UNISIG, EEIG ERTMS USERS GROUP (Issue 3.3.0)

[Subset 040] : *ERTMS/ETCS Dimensioning and Engineering Rules*, UNISIG (Issue 3.2.0)

[Subset 041] : *ERTMS/ETCS Performance Requirements for Interoperability*, UNISIG (Issue 3.2.0)

[Subset 119] : *ERTMS/ETCS Train Interface, Functional Specification*, ERA, UNISIG, UNIFE (inofficial draft)

[prEN 17997-1] : *Railway applications – Braking – Definition of ETCS brake curve parameters for Gamma trains – Part 1: Emergency brake curve parameters*, CEN/TC 256, 2023-05, WI 00256966:2023