FBS-RailML®-Interface (Export)

RailML-Versions 2.0 / 2.1 / 2.2
RailML-Profile-Versions 2.0.2 / 2.0.5 / 2.1.0 / 2.2.0

Manual

State: August 2022

Internet: www.irfp.de
E-Mail: info(at)irfp.de

Address: iRFP e. K.
Institut für Regional- und Fernverkehrsplanung
Hochschulstraße 45
01069 Dresden

Phone: +49 351 4706819

Inhaltsverzeichnis

1. General Hints Concerning the FBS-RailML-Interface

The current version of the FBS-RailML interface enables the export of selected **infrastructure data** and almost all **timetable data** from FBS into the RailML format in a structured, systematic form for technical evaluation in other programs. It is also possible to import timetable data from RailML, which is pictured in a separate manual. This document describes only the export.

Vehicle data from FBS is included in the exported RailML files to a basic extent, necessary to evaluate the timetable data. However, it is not included completely, as the main purpose of this RailML interface is seen in the exchange of timetable data. To import vehicle data to FBS, a special programme is available on the iRFP internet page at

http://www.en.irfp.de/entry-of-fbs-vehicle-data.html

Please note that the RailML standard does not automatically mean that two programmes can always exchange data without errors. This is due to the fact that the RailML standard does define the general structure of infrastructure and vehicle data - however, on the one hand, the **scope** of the data (the completeness) and on the other hand, the **details** of some special data differ depending on the application in different programmes. For this reason, there is a special interface description for the FBS implementation of the RailML format. It describes the specific data fields used by FBS and their contents. The current version of the interface description and the scheme documents (*.xsd) can be found at

http://www.en.irfp.de/technical-advice-to-railml.html

The exact amount of data to be exchanged is not determined by the RailML standard alone. If FBS outputs *more* data than the reading programme expects, the additional data merely has to be "skipped" during reading - which is usually unproblematic. However, it can also happen that the reading programme expects special data that is not available in this form in FBS. These must then usually be generated (calculated or derived from other data) during the import. In this respect, it may be necessary for the opposite party to implement a special interface for the import of FBS RailML files.

Therefore, please bear in mind that before using the interface, it is usually necessary to send the FBS interface description and, if necessary, some test data to the software supplier of the other programme involved in the data exchange. We are of course happy to support our customers in the pre-conversion of user-specific test data and the discussion of the contents of such interface.

In general, the following applies: It must be individually agreed for each case, which of the settings described below should be made for a specific export. Depending on the intended use, different settings may be necessary. Please coordinate this with the receiving side before use. The relevant settings can be saved as a template, as described in the *last chapter* of this document.

Please note that apart from the settings explained and visible here, a lot of other information from your FBS network is written into the RailML file. You should consider this before passing on the files and check it if necessary. For easier orientation, the **file names and parts of the paths** of the FBS files of the exported FBS network are also written into the RailML file. The associated paths are only exported if they are a relative path (relative to the network file or starting with a placeholder); absolute paths (starting with drive letter or network resource) are not exported.

2. Information About Using the Interface for FBS-Users

In order to be able to export data to RailML the following requirements have to be met in FBS:

- The entire FBS network must have a uniform timetable period.
- The engine, carriage/waggon and train category databases as well as the default railway company must be uniform in the network. All operating locations (e.g., stations) must have a unique abbreviation.
- Each change of line number must be placed on an operating location. If necessary, a "pseudo" operation location of the type "Change of line" at the location of the line number change, or the line number change must be moved to an adjacent operation location.
- All trains and train parts must be distinction the network, i.e., train numbers must not be used more than once, unless they can be distinguished via train number indexes or train part numbers. Should there be several trains with the same train number (supplementary timetables/alternate slots), multiple train number indexes must be used. In the case of several trains with the same train number containing a different planning status (e.g., ordered train slot <-> offered train slot), the filter by layer (see also chapter: *Scope of timetable data to be exported - Filter for trains and train parts*) can be used when exporting, to ensure uniqueness.

In addition to the technical minimum requirements, there are a number of other requirements that may result from the reading programme. Such requirements must be considered before the export - ideally already when creating the data in the FBS. Typically, these may include:

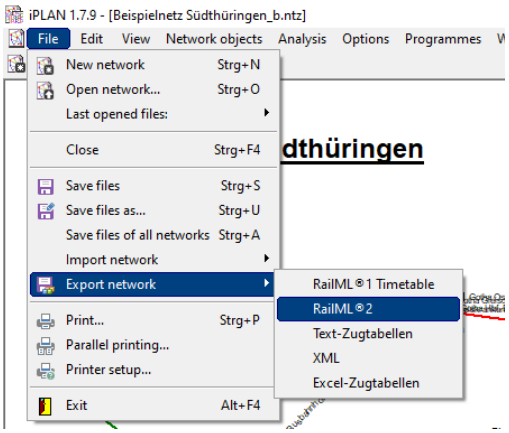
- All train parts (to be exported) must have a line description. Many programmes, especially in the field of passenger information, require a line description.
- All relevant operating locations (i.e., at least the stations in passenger traffic or rather train stops) must have a station number (IBNR). Many reading programmes do not identify the stations by their abbreviation - like FBS - but by a number. This number must then be entered in the FBS station database under "station no. (IBNR)". We are happy to support you if you need to import a list with such numbers for a larger amount of operation locations.
- It is possible that the reading programme requires train numbers, train part numbers and/or line descriptions to be purely numeric. In contrast, both in FBS and in RailML in general, it is possible that these elements may also contain letters and other characters (sometimes contrary to their caption "number").
- It should be considered, that the content or unit of measurement of some data fields that are optional for FBS is not fixed from the get-go. For example, concerning operation locations, no special format is assumed by FBS for their number, their station fee class or their coordinates. However, a reading programme might expect for example the coordinates to be in a certain format like the WGS84 system.

- It is not possible in RailML that an operating location positioned on several lines has different properties per line. It could be, for example, that an operation location (with the same abbreviation) is a station on one line and a junction on another (possibly parallel) line. In principle, such stations should also have different abbreviations in FBS; however, this is often not possible (in Germany, for example, due to compatibility with DB Netz). In those cases, you will therefore be asked during the export whether the operating point should be exported with the union of properties or whether you want to cancel the export.

Some of the points mentioned can optionally be checked by the FBS-RailML-interface when exporting from FBS to RailML (see also chapter *Checks of Content*).

3. Operating the Interface

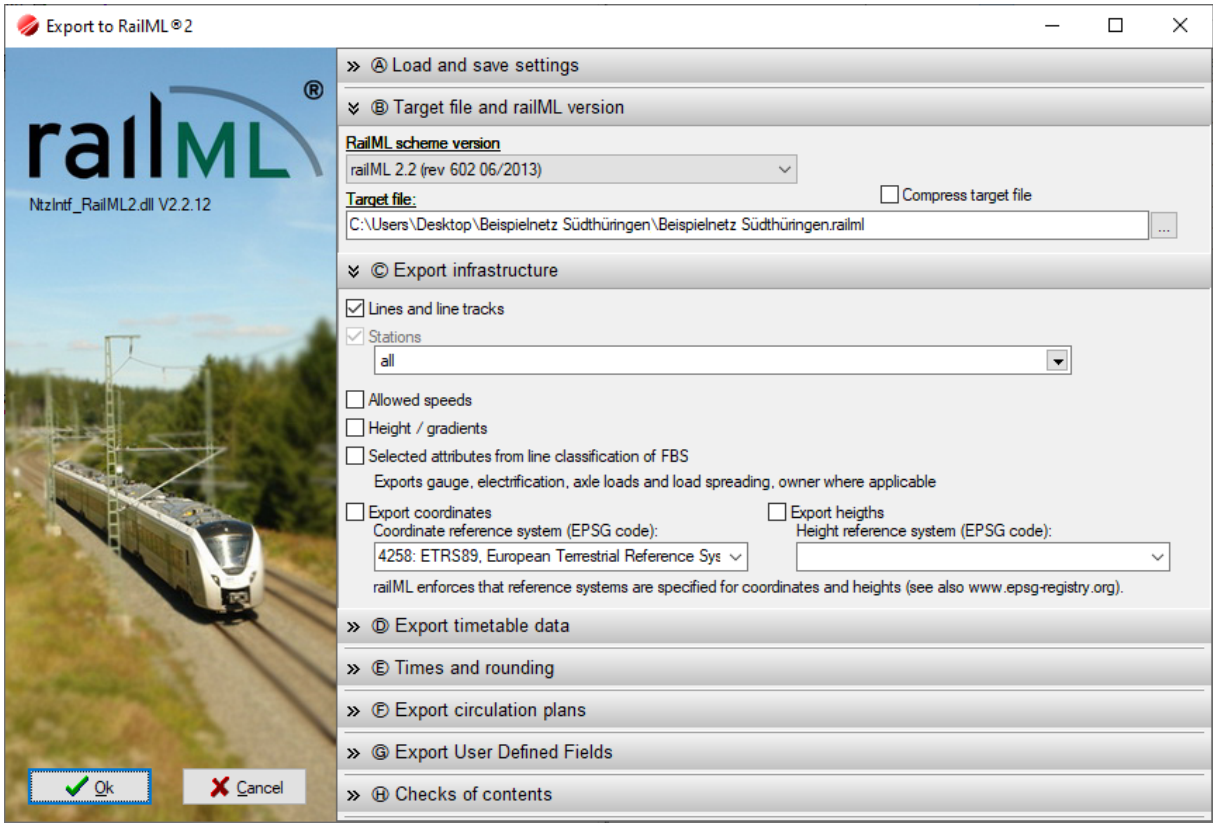
The export of data to RailML is triggered from FBS iPLAN via the menu item **File → Export Network → RailML®2**. The menu item is available if the FBS RailML interface has been installed and is enabled for the current FBS licence.



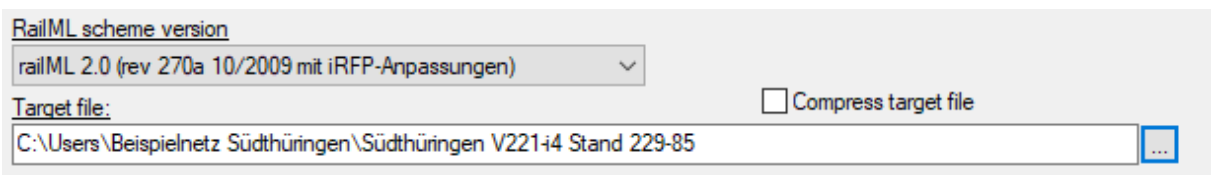
To install the interface, the file NtzIntf_RailML2.dll must be in the FBS programme directory (usually C:\Program Files\FBS). It can be copied there via online update or manually - a special installation programme is not needed.

More about installation and updates can be found here:

http://www.en.irfp.de/technical-advice-to-railml.html



3.1. Target File and RailML-Version



The RailML schemes are subject to constant development. At the same time, however, a scheme that can be exported once should be offered unchanged over a longer period of time

in order to maintain a certain "investment security" on the importing side. Therefore, the FBS-RailML interface offers the export selectable in several past **RailML scheme versions**. Older versions usually contain content restrictions compared to newer versions; these are described under

http://www.en.irfp.de/technical-advice-to-railml.html

(Section: Hints to usability and limitations of the RailML versions).

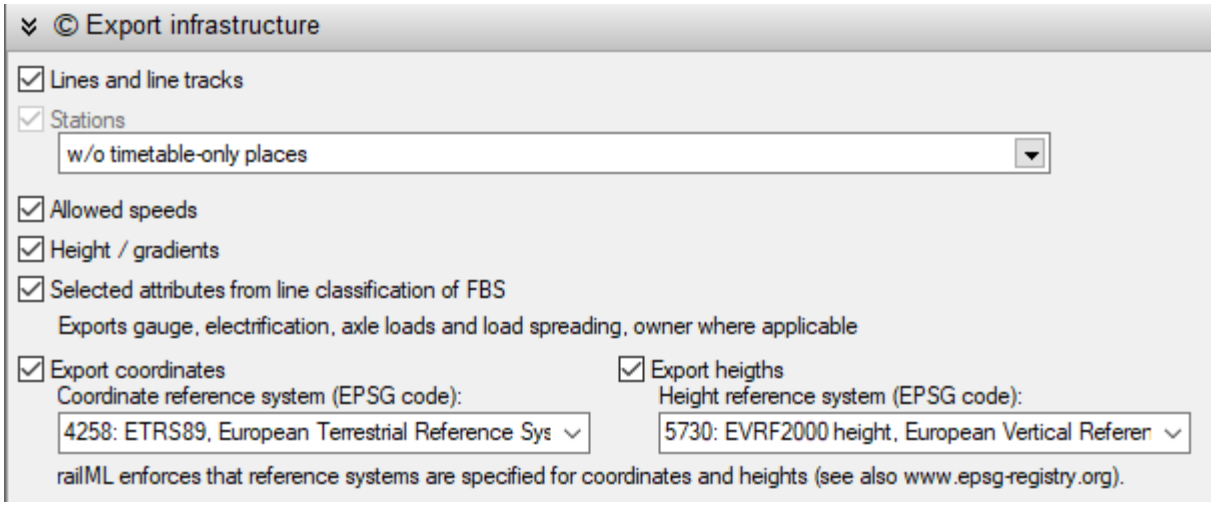
For the **target file** enter a file name with path and the extension .xml, .railML or .railmlx as the target file, whereby the following rule applies:

- ***.xml** is the standard for RailML files up to and including version 2.1 (backward compatibility),
- ***.railml** is the standard for newer, uncompressed RailML files,
- ***.railmlx** is the default for compressed RailML files (of any version).

The default is the path and name of the FBS-NET file, but with a customised extension. For default settings of other paths and/or a specific file name, see the notes under *Load and Save Settings* at the end.

Compression ("packing") of RailML files is only standardised from version 2.2, but can also be set on for older versions in FBS. Compression is done according to the zip file standard using the Deflate algorithm, meaning such files can also be unpacked manually with common "zip programs" if required.

3.2. Export Infrastructure



Export infrastructure, lets you select to a limited extent how much infrastructure information is to be output. If only the actual timetable is to be output (e.g. for passenger information or crew-rosters), you can switch off the option **Lines and line tracks**.

However, a minimum amount of infrastructure - specifically operating locations (e.g., stations) - is always necessary for RailML timetable data, which is why the option **operating locations** cannot be switched off completely. However, in order to save file size of potentially quite large RailML files, you can hide the operating points that are typically not relevant for most applications of timetable data such as passenger information or crew-rosters: *token only exchange places* are (staffed) block posts, block signals or ATC block posts that have no other property (i.e. are not at the same time e.g. halts). *Timetable-only places* are

"operating locations" in the FBS sense, which have neither operational nor traffic properties (e.g. infrastructural boundaries, changes of line). Operating locations where lines or trains start or end are exported in any case. Operations locations with public stops of trains must not be skipped (excluded) by the filter.

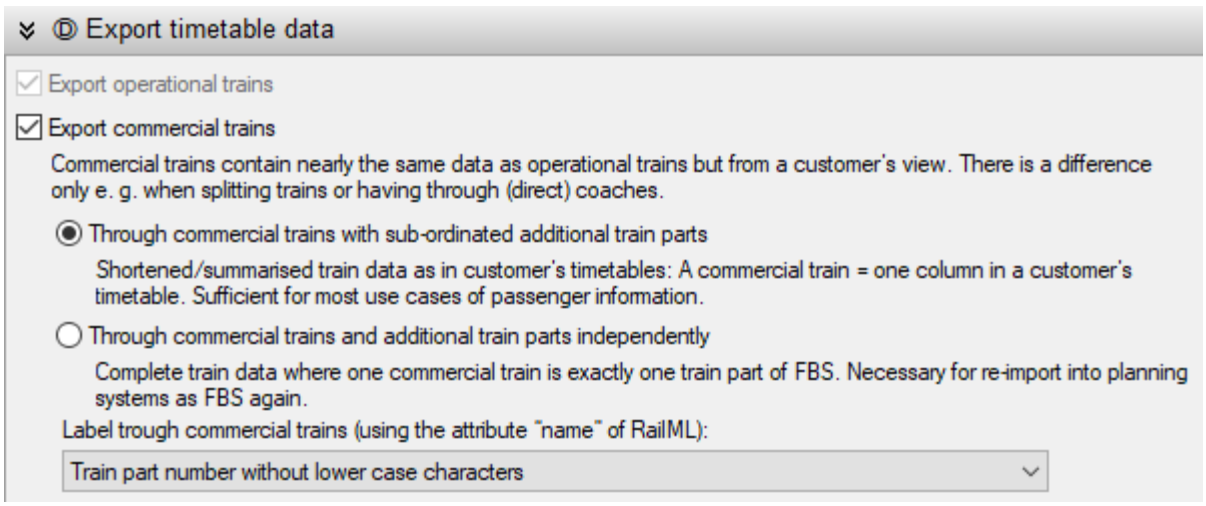
The options **Allowed speeds**, **Height / gradients** as well as **Selected attributes from line classification of FBS** are infrastructure data whose export can be considered in the following scenarios:

- Exchange of basic infrastructure data with timetable data, to be able to display book timetables or similar (electronic) documents.
- Exchange of a basic infrastructure data stock to avoid repeated manual data entry. However, as different programmes usually have different data models, a complete transfer will hardly be possible, but manual adjustment, at least in detail, will remain unavoidable.

For a general exchange of timetable data - for which the FBS-RailML interface was mainly developed - these options are rarely necessary.

Coordinates including heights are only permitted in RailML from version 2.2 onwards if the respective reference system is specified (up to RailML 2.1 you can still output coordinates without a reference system). For FBS the reference system is not important (coordinates are only used relative to each other in FBS), therefore FBS does not know the specifications; you may have to add them before export. Geo-coordinates are typically from WGS84; station elevations in Germany are from the German Main Elevation Grid 1912. However, since this does not have an EPSG number, you may have to specify the German Main Elevation Grid 1992 as a substitute.

3.3. Export Timetable Data – Operational and Commercial Trains



Export timetable data lets you adaptat the data defined in the RailML subscheme <timetable>.

RailML distinguishes between operational and commercial trains. The **operational trains** are fundamental and are always exported. They represent the view of the infrastructure manager, i.e., only one operational train can run on each line section at any given time.

In contrast, the **commercial trains** represent the passenger's view of the timetable. Under certain circumstances, several commercial trains can run simultaneously (in one operational train) (e.g., portion working, through coaches). The output of this data is in a sense redundant to the operational trains and optional. However, it is the basis for most subsequent applications (in particular e.g. for passenger information) and is mandatory in such cases.

a) For commercial trains, there are, depending on the application, these possibilities: They usually equal the **direct connections** from FBS, which are the basis for costumer's timetable and arrival & departure plan (see iPLAN: Analysis → Train summary → Direct connections). Commercial trains of this kind only differ from operational trains if portion working, through coaches or other vehicle changes are "at work" - i.e., if direct connections in terms of transfer-free connections across operational trains occur.

b) In rarer cases, in addition to direct connections, additional train parts also count as commercial trains - "additional train parts" here in the sense of train parts that run for purely commercial reinforcement (higher seating capacity) without representing additional transfer-free connections anywhere.

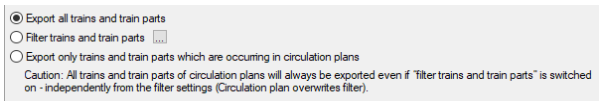
Case (a) is sufficient for most use cases of passenger information. Case (b) is mainly needed when a 1:1 recreation of the exact train parts from FBS is wanted, especially a re-import of the train data into FBS.

When exporting commercial trains, you have to always decide how the RailML attribute *name* of the direct connections should be set. Detailed information on the difference between the three options can be found in the document

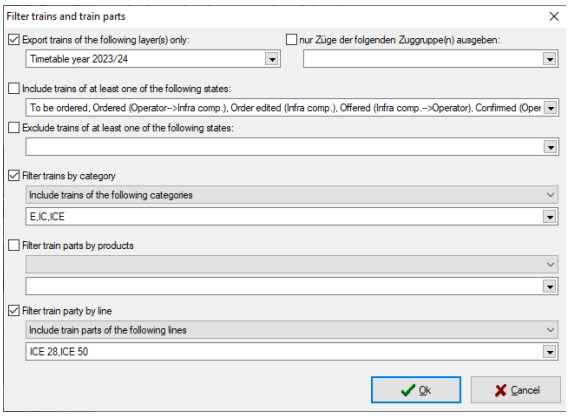
http://www.en.irfp.de/faq-english/61040e.html

If additional train parts are exported as independent *commercial trains* (case b), they do not receive the *name* attribute - the label always refers only to direct connections.

3.4. Export Timetable Data – Filter for Trains and Train Parts



Filter trains and train parts lets you determine the scope of the trains to be exported. For example, you can specify here that trains from other TOCs that you have only included for information in your graphic timetables should not be exported. These settings are referred to as **Filters** in the following.



The overall rule is that a train, train part or a direct connection must not be "torn apart" by a filter, i.e. a filter must always cover a complete train route and not only a part of it. If, for example, a train has train type **R** (the exact meaning of "R" is not relevant for this example) in its first route section, in the second route section has type **E** and in the third section **R** again and one would set a filter that excludes train type **E** (or only includes **R**), then the train would be "torn apart" by the filter into two independent trains - which would then, however, inevitably

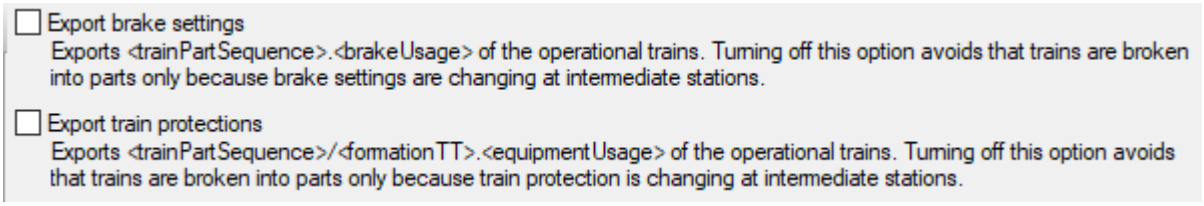
have the same train number. In the exported RailML file, this would violate the rule that each train must have a unique train number and would therefore look like a "primary key violation".

FBS cannot check all filter combinations for feasibility during export - you are responsible for the correct application of the filters yourself. Therefore, please proceed with caution, especially if you apply several filters at the same time. If you use several filters, they are logically linked with AND, i.e., a train or train part is only output if all set filters apply at the same time. A train can then be excluded, for example, on one route section based on the days of service and on another based on the train type.

Please also note the relations between the export of circulation plans and filters mentioned in the chapter Export Circulation Plans.

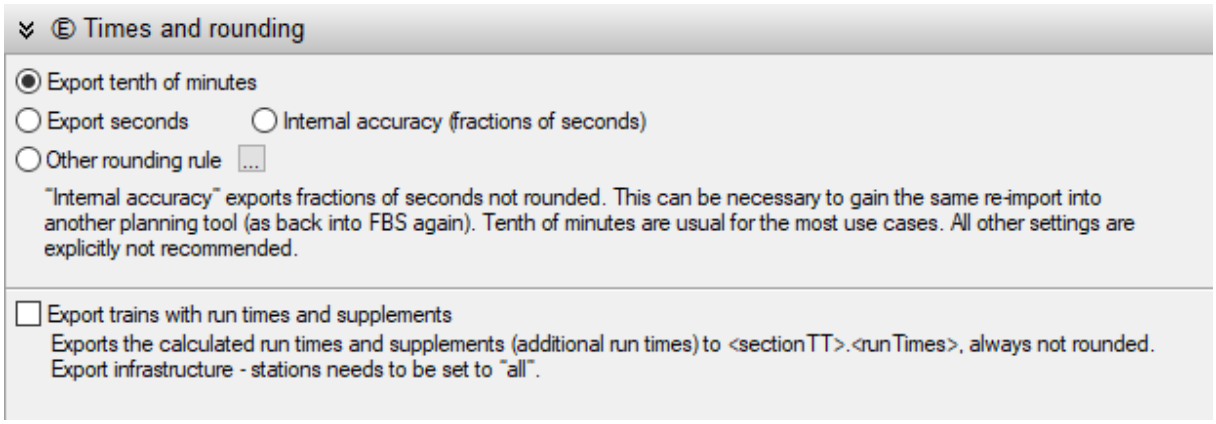
Overall, any more detailed filtering must be done when importing the data and not when exporting it. Since RailML is a universally valid interface format, the exported data must be independent of the "target programme". Only the reading interface knows the exact scope of the required data and what is considered a primary key in the target data model. Only the reading programme can therefore decide which data can be skipped and which must be analysed.

3.5. Export Timetable Data – Brakes and Train Protections



Brake settings and **train protections** are among the fundamental data defined in RailML and thus exportable, but they are not evaluated by the reading software in all cases (especially not for passenger information). If they are exported, a change of brake or train protection settings on the train route (at an en-route station) must also be properly described. This can cause a train route in RailML to be "torn apart" (divided into sections) simply because such settings change - making the RailML file more complex and larger than necessary. To avoid this, you should only switch these options on when the respective information is also evaluated.

3.6. Times and Rounding



In timetables, times are usually given in whole minutes (for commercial purposes) or tenths of minutes (for operational purposes). However, FBS internally needs a higher accuracy to avoid cumulative rounding errors at high speeds and closely spaced operating points (especially block posts such as block signals). RailML allows any conceivable accuracy between whole minutes to fractions of a second, which corresponds to the internal accuracy of FBS. Depending on the application, you should choose one of the following options:

- **Export tenths of minutes** should be the usual case for timetables when in doubt.

- **Internal accuracy** (fractions of seconds) is necessary if an exact re-import of the data into FBS or a comparable timetable construction software is intended. This setting avoids rounding errors due to data exchange.

- **Other rounding rule - round to integer minutes** can be considered if a data export for passenger information takes place and the importing software cannot round itself, i.e., expects whole minutes. To avoid contradictions, e.g., with driver's timetables or arrival & departure plans created with FBS, you should set the same rounding rule as in these documents, like *round everything down* or *traffic*.

However, settings other than *tenths of minutes* and *internal accuracy* are absolutely not recommended, especially not the *Export seconds* setting, which provokes unnecessary rounding errors and discrepancies with other timetable documents. (This setting is only present for backward compatibility with older RailML files and may be dropped altogether in a future version of the interface).

Export trains with run times and supplements lets you specify whether the RailML structure <timetable> should output <sectionTT>.<runTimes> . Depending on this, the following information about components of the train run time is output for each individual route section of each train:

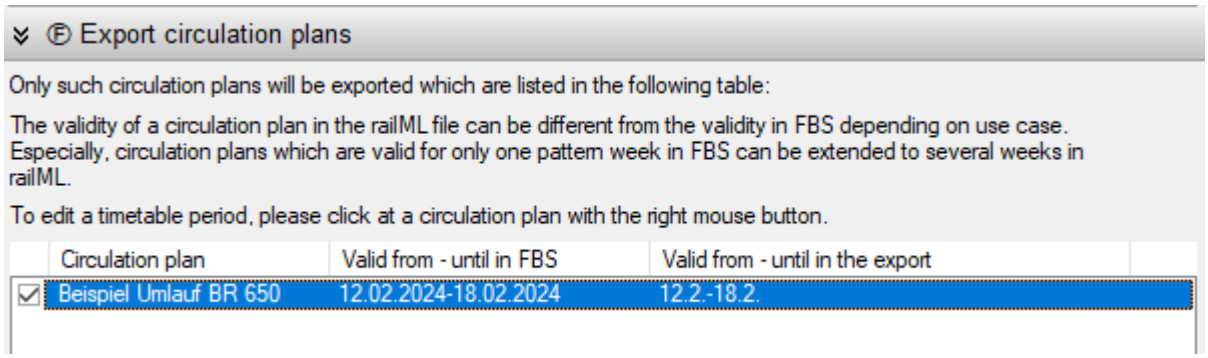
- the attribute *<runTimes>.minimalTime* with the actual ("physical") run time,
- the attribute *<runTimes>.operationalReserve* with the linear time supplement as time value,
- the attribute *<runTimes>.additionalReserve* with an optional non-linear time supplement as time value (if available),
- the attribute *<sectionTT>.percentageSupplement* with the linear time supplement as percentage value.

Switching off this information, which is not always needed, saves file size. You can only output this information if *Stations / all* has been selected under *Export infrastructure*.

Further information from <sectionTT> about the used line tracks (regular or opposite track) and the distance to be covered is always output if you have ticked the option *Lines and line*

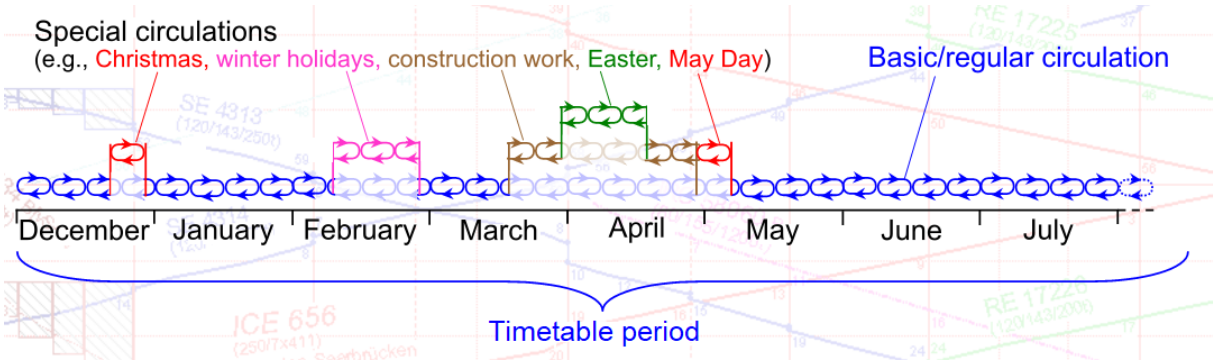
tracks under *Export infrastructure*.

3.7. Export Circulation Plans



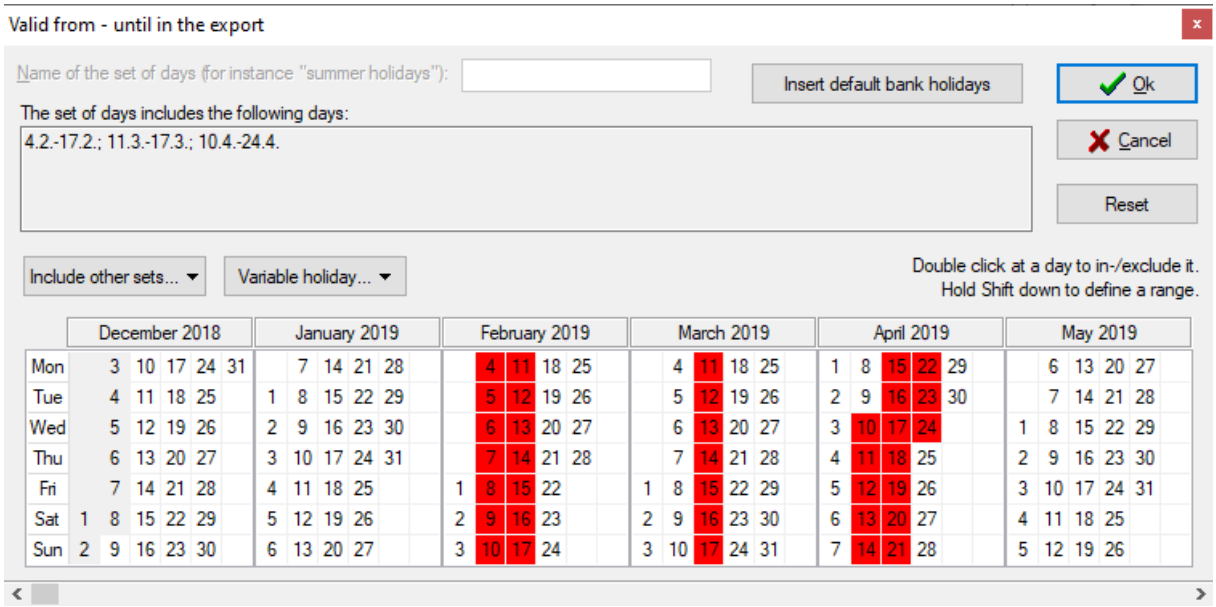
If you also want to **export circulation plans** with the RailML file, every one of them in the FBS network can be included or excluded individually.

If the timetable period of the FBS network is defined, each circulation plan has a validity period. In the simplest case, this is the entire timetable period of the network, but it can also be only a limited (partial) period of the timetable period. There can therefore be several circulation plans for the same vehicle series. Normally there is a **basic or regular circulation** for the timetable year, which is valid without interruption and without a specific date reference. This can be overwritten for certain periods (Christmas, Easter, holidays, construction work) by **special circulations**.



The partial periods of circulations for the same vehicles should be disjunctive, i.e. they should not overlap - otherwise the reading programme may not be able to determine which circulation should apply at which time.

Whether the date of the weekdays is displayed or not serves as a **default** setting for the validity of the circulation plans to be exported: Circulation plans of a specific week for which the display of the date is switched **off** are exported as a regular circulation for the whole year by default.



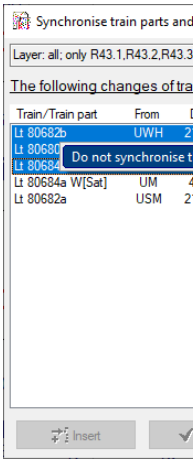
To change a timetable period of a circulation plan for export, right-click the circulation plan. In the calendar window ("Valid from - until in the export") you can select any number of independent periods in which the circulation is repeatedly valid. This does not affect the period of the circulation in FBS. If you give a circulation plan a longer period than it has in FBS, you have to make sure that the trains in the timetable run on all marked days. (The traffic periods of the trains are not automatically extended to the period of the circulation - which could result in slot conflicts).

Please pay attention to the following relation between circulation plans and filters:
Priority of circulation train parts over filter settings: All train parts that occur in a schedule to be exported are always exported - regardless of any train/train part filter settings. If, for example, a train part of line "21" is not to be exported and you have set the filter *Exclude train parts of the following lines* to "21", but this train part occurs in a circulation plan to be exported, it will still be exported!

If you do not export the whole timetable period of the network, by checking **Reduce timetable period** (under *Export timetable data*), the following rules apply:

- No circulation plan may be valid outside the selected export period. The export validity of the ciculation plans is therefore limited to the intersection with the selected export period.
- The selected export period may never extend beyond the validity of the network's timetable period.
- A circulation period may be exported beyond its validity defined in FBS, provided it lies within the selected export period. It then applies as mentioned above that you yourself are responsible for ensuring that the trains also run on the days in question.

Export of shunting movements in the timetable:
If a train part of a circulation plan to be exported cannot be found in the graphic timetables, the export is aborted with an error message, as it can be assumed that this is a previously undetected error in the circulation plan (forgotten to synchronise). In some cases, however, such runs (not planned in the graphic timetable) are desired, because they are, for example, shunting runs between station sections. They can be explicitly included with the property ***Do not synchronise this running***. This property can be conveniently set in the window *Synchronise train parts and graphic timetables* (Alt+R in a circulation plan) via right-click for several/all relevant train runs.



The following applies to the RailML export: The option *Do not synchronise this running*, only exports a "placeholder" for the train section in the circulation plan *if it cannot be found in the graphic timetables*. If, however, it can be found despite the option being set (for example, because only some properties should not be matched), then no placeholder is exported, but instead the regular graphic timetable train part.

With the option **Export inner weekday group**, the RailML export of the circulation plans is extended by additional information. This information is intended to enable a reading programme to display repeating run sequences, e.g., on Tue-Thu, on one sheet instead of using separate identical sheets for Tue, Wed and Thu. The prerequisite is, of course, that there are identical weekdays in the circulation plan. (It makes sense to activate the corresponding option in the FBS circulation module when creating the circulation plan, so that identical inner weekdays can be monitored).

Since the original RailML structures do not provide any attributes for repeating weekdays, this option adds another XML namespace to the resulting RailML file. This XML namespace defines the additional necessary attributes.

3.8. Export User Defined Fields

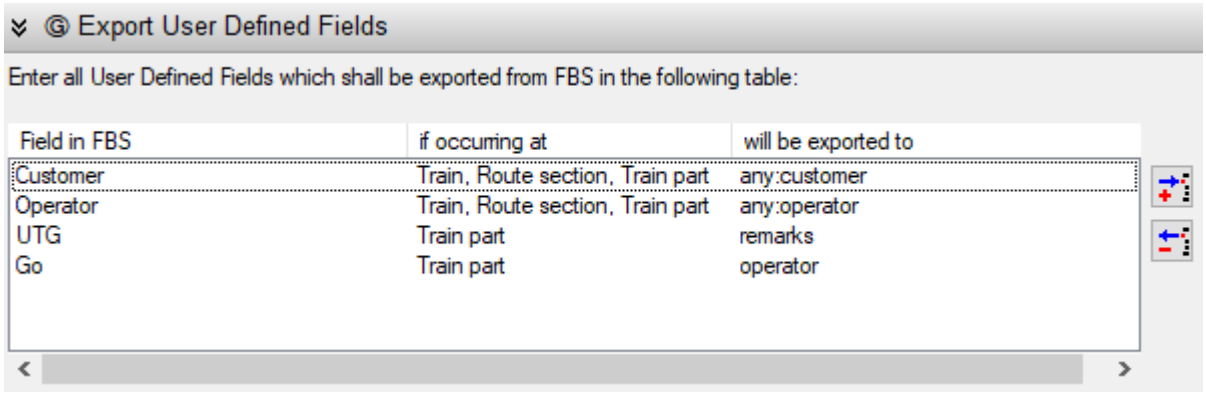
User defined fields (UDF) are used in the FBS to map data that is not important for the actual FBS function, but occurs in the environment of timetable data - for example, contractual information such as contract numbers, contract partners, transportation authority. As this information is not relevant in FBS itself, it is considered all the more (or even only) for data exchange between interfaces.

The FBS RailML interface can be individually extended in this respect with user-defined information. The RailML data format offers the following possibilities to map additional information:

a) in a RailML standard data field (attribute), which is not otherwise used by FBS. Possible attributes at train part level are *remarks, debitcode* and *operator* as well as the central structure <organizationalUnits> with possible entries for *infrastructureManager, vehicleManufacturer, vehicleOperator, customer, railwayUndertaking, operationalUndertaking, concessionaire* and *contractor*.
b) in an extension of the RailML standard with additional data fields (attributes). This can

be done if the user-defined field does not occur at train part level or if the standard fields are in any way inapplicable or unusable. There are almost no limits to the extension; however, such an extension requires the definition of a separate XML namespace in the RailML file. The XML namespace describes the additional attributes; it may also be necessary to fix the attributes in a so-called schema file (*.xsd file).

Once again: Such settings should be coordinated with the importing side; the export only makes sense if the other side can also import the user-defined fields.

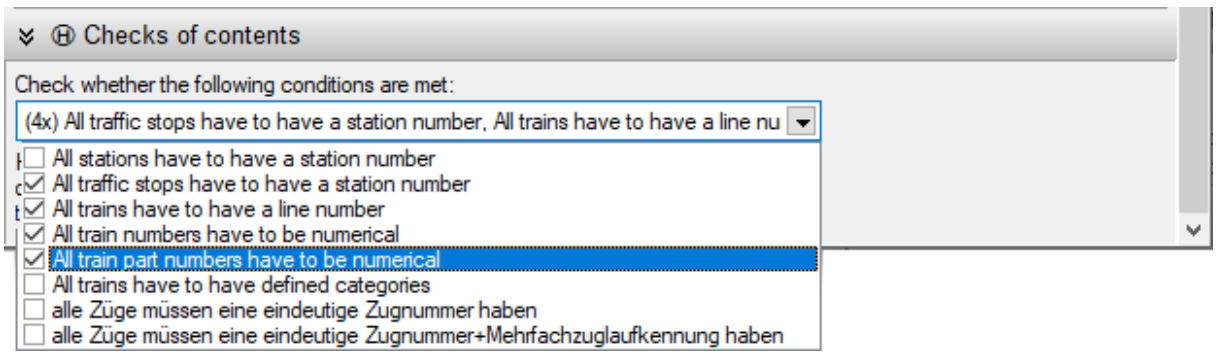


In the export settings of the FBS-RailML interface you can assign UDFs occurring in FBS with RailML attributes. UDFs in FBS can occur on trains, route sections or train parts. These three positions are permanently assigned to the following equivalents in RailML:

Train	= <train> (here only *operational trains*)
Route Section	= <trainPartSequence>
Train part	= <trainPart>

It should also be noted that the standard data field *debitcode* is set to a numerical value by RailML (only digits are permitted here). FBS does not check this condition. Apart from this exception and the somewhat cumbersome <organizationalUnits>, the two fields *remarks* and *operator* provide two relatively freely and straightforwardly usable data fields, at least at train part-level.

3.9. Checks of Content



Certain content requirements exceeding the technical minimum can be imposed on the RailML data by the receiving interface programme. These requirements must be respected when entering the data into FBS. To make it easier to ensure that the RailML data meets certain requirements, typical checks can be switched on during the export. If one of the conditions to be checked is violated, the export is aborted with a corresponding note.

Please note that these are only checks to avoid consequential errors or to reveal errors earlier - the interface does not add to or manipulate the data to be exported!

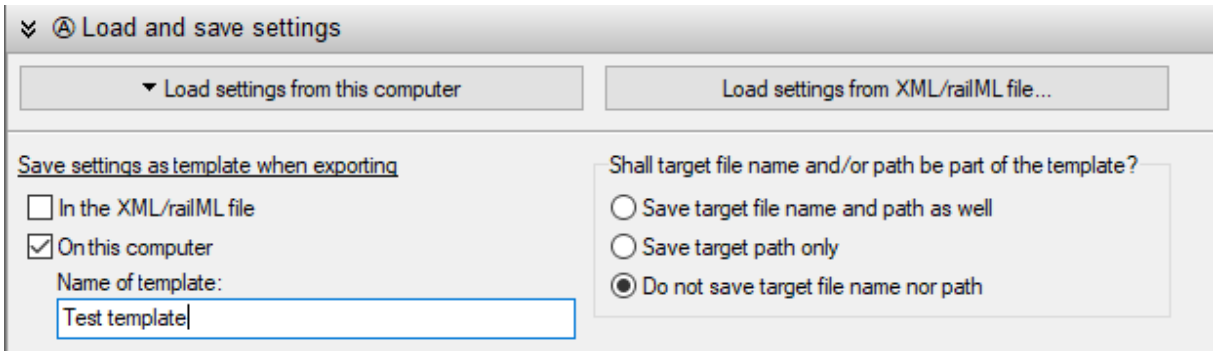
The settings shown are typical for many applications of the RailML2 interface. If you regularly export with the same settings, we recommend saving them as a template (see also *Load and Save Settings*).

3.10. Load and Save Settings

Usually, depending on the application, there are individual requirements for the contents of RailML files. You can set these as described in the previous chapters. Due to the wide range of options, it is possible to define specific default settings for certain applications (programme pairs such as "Export from FBS for passenger information with ...") so that content dependencies are fixed or are already checked by FBS during export. If you are developing an import interface or a company is doing so on your behalf, please contact us to coordinate the necessary contents and possible presetting of the options in FBS. We also ask you to fill out the following form with regard to the default settings and send it to us:

Form templates for FBS-railML-export (PDF, German)

Of course, you can also independently **save the settings as a template**, which you have made. In doing so, the network-dependent settings such as circulation plans and a class, product or line filter are also saved. However, when exporting a different network later, these settings are only used if they can actually be applied.

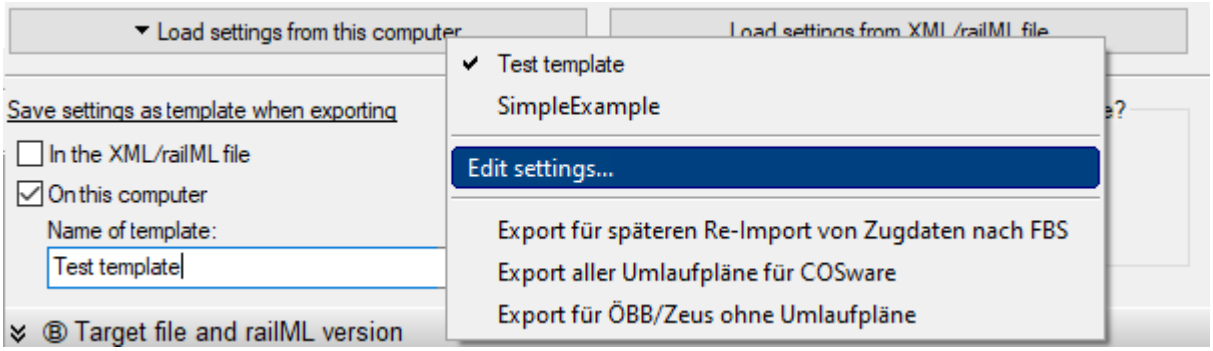


You can select whether the file name or path should also be saved as a template with in the right-hand area. When choosing *Do not save target file name nor path* the current file name is fixed for each future export; the input field *Target file* is therefore no longer connected to the current file name of the network file. With *Save target path only*, only the current path is saved - the file name itself remains dependent on the name of the network file. It would thus always be exported to the same target path (e.g., a RailML transfer directory), but with different RailML file names. *Do not save target file name nor path*,makes the default setting of the input field *Target file* the path and file name of the current network file (but with a different extension).

If you save the settings **in the XML/RailML file**, you can specify the last successfully exported file as the source for the next export (button *Load settings from XML/RailML file*). In order for the settings to be saved in the RailML file, an additional XML namespace (for the FBS export settings) must be added to the file for formal reasons. The reading programme must be able to "skip" this XML namespace correctly.

If you want to save the settings **on this computer**, you must give the template a unique

name. It can then be loaded again under this name (button *Load settings from this computer*). These templates are saved in the file *NtzIntf_RailML2.xmlini*, in the path that is set in the **configuration settings** of FBS for user-related configurations. To find the file, open the FBS start window and click *Installation and maintenance* → *Configuration settings...* → *Configuration files*. You can take this file with you when "moving" to another computer.



In addition to the self-created templates, there are a number of predefined templates to be found under *Load settings from this computer*. Under **Edit settings...** you can delete, rename or change the order of templates you have created yourself.