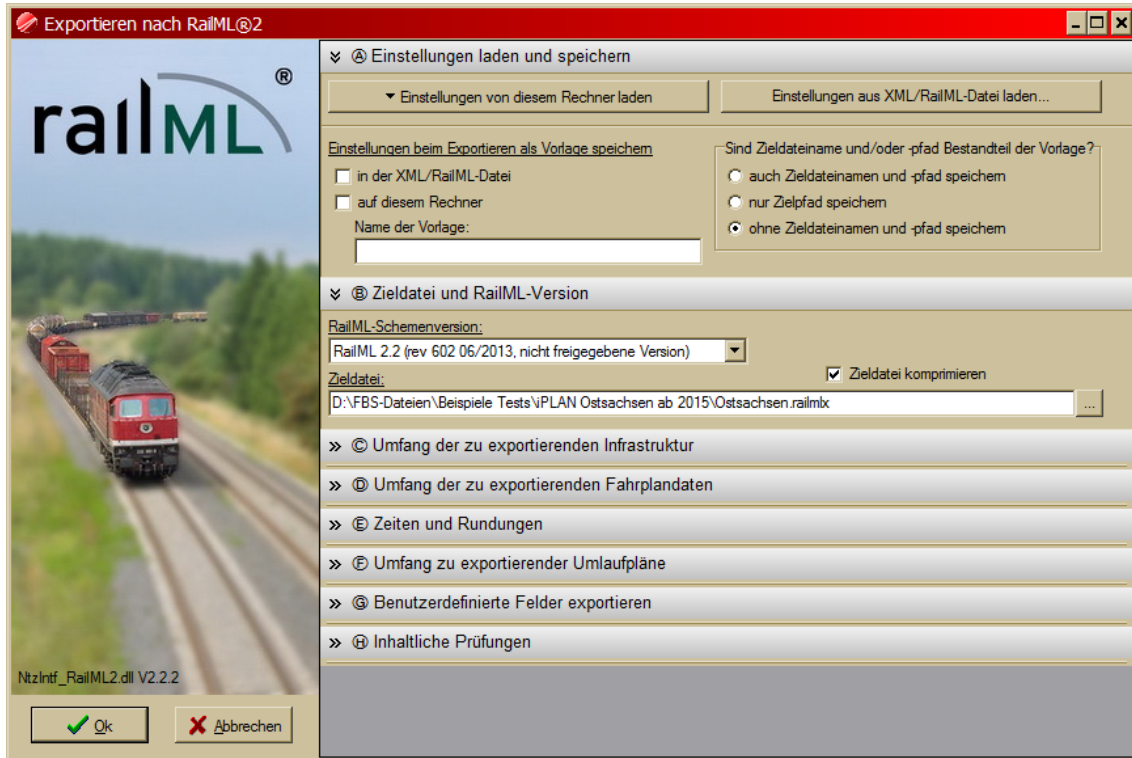


Hinweise zur FBS-RailML®-Schnittstelle

Stand: 24.07.2015

RailML-Versionen: 2.0 / 2.1 / 2.2

RailML-Profil-Versionen: 2.0.0 / 2.0.5 / 2.1.0 / 2.2.0



Inhalt

Allgemeine Hinweise zur FBS-RailML-Schnittstelle	2
Hinweise zur Nutzung der Schnittstelle für FBS-Anwender	3
Bedienung der Schnittstelle	4
Zieldatei und RailML-Version	4
Umfang der zu exportierenden Infrastruktur	5
Umfang der zu exportierenden Fahrplandaten – verkehrliche und betriebliche Züge	6
Umfang der zu exportierenden Fahrplandaten – Filter für Züge und Zugteile	7
Umfang der zu exportierenden Fahrplandaten – Bremsen und Zugbeeinflussung	7
Zeiten und Rundungen	8
Umlaufpläne und Gültigkeitszeiträume	9
Benutzerdefinierte Felder exportieren	11
Inhaltliche Prüfungen	12
Speichern der Einstellungen als Vorlage	12

Allgemeine Hinweise zur FBS-RailML-Schnittstelle

Die FBS-RailML-Schnittstelle ermöglicht in der aktuellen Version einen Export von ausgewählten **Infrastruktur-** sowie nahezu allen **Fahrplandaten** aus FBS in das RailML-Format in strukturierter, systematischer Form zur technischen Auswertung in anderen Programmen. Ebenfalls ist ein Import von Fahrplandaten aus RailML möglich, wofür eine separate Beschreibung existiert. Dieses Dokument hier beschreibt den Export.

Fahrzeugdaten aus FBS sind in den exportierten RailML-Dateien in einem grundlegenden Umfang enthalten, der zum Auswerten der Fahrplandaten erforderlich ist. Sie sind jedoch nicht vollständig enthalten, da der Hauptzweck dieser RailML-Schnittstelle im Austausch von Fahrplandaten gesehen wird. Zum Einlesen von Fahrzeugdaten nach FBS steht Ihnen auf den iRFP-Internetseiten ein spezielles Programm zur Verfügung unter www.irfp.de/deutsch/fbs/ilok.html.

Bitte beachten Sie, dass der RailML-Standard nicht automatisch bedeutet, dass zwei Programme mit Sicherheit fehlerfrei Daten austauschen können. Das liegt einerseits daran, dass der RailML-Standard zwar die allgemeine Struktur von Infrastruktur- und Fahrzeugdaten definiert – jedoch sind zum einen der **Umfang** der Daten (die Vollständigkeit) und zum anderen die **Details** einiger spezieller Daten je nach Anwendung in verschiedenen Programmen unterschiedlich. Aus diesem Grunde gibt es eine spezielle Schnittstellenbeschreibung für die FBS-Implementierung des RailML-Formats. Hierin werden die konkret von FBS verwendeten Datenfelder und deren Inhalt beschrieben. Die jeweils aktuelle Version der Schnittstellenbeschreibung und der Schemendateien finden Sie unter www.irfp.de/deutsch/fbs/schnittstelle_railml.html.

Insbesondere der genaue Umfang der auszutauschenden Daten ist allein durch den RailML-Standard nicht bestimmt. Sofern FBS *mehr* Daten ausgibt, als das lesende Programm erwartet, müssen die zusätzlichen Daten lediglich beim Lesen „übersprungen“ werden – was i. d. R. unproblematisch ist. Es kann jedoch ebenso gut vorkommen, dass das lesende Programm spezielle Daten erwartet, die in FBS in dieser Form nicht vorhanden sind. Diese müssen dann meist beim Import erzeugt (berechnet, aus anderen Daten abgeleitet) werden. Insofern kann es notwendig werden, dass zum Import von FBS-RailML-Dateien eine spezielle Schnittstellenimplementierung auf der Gegenseite erstellt werden muss.

Bitte bedenken Sie daher, dass es vor einem Einsatz der Schnittstelle i. d. R. notwendig ist, die FBS-Schnittstellenbeschreibung und ggf. einige Testdaten dem Software-Lieferanten des anderen am Datenaustausch beteiligten Programms zukommen zu lassen. Wir unterstützen unsere Kunden selbstverständlich gern bei der Vorab-Konvertierung von nutzerspezifischen Testdaten und der Besprechung der Schnittstelleninhalte.

Im Allgemeinen gilt: Welche der im Folgenden beschriebenen Einstellungen für einen konkreten Export vorgenommen werden sollen, muss im Einzelfall mit dem Empfänger der Daten abgestimmt werden. Je nach Verwendungszweck können unterschiedliche Einstellungen notwendig sein. Bitte stimmen Sie dies vor der Anwendung mit der empfangenden Seite ab. Die jeweils zutreffenden Einstellungen können als Vorlage gespeichert werden, wie dies im letzten Kapitel dieses Dokuments beschrieben ist.

Beachten Sie bitte, dass außer den hier erläuterten und sichtbaren Einstellungen viele weitere Informationen aus Ihrem FBS-Netz in die RailML-Datei geschrieben werden. Dies sollten Sie vor Weitergabe der Dateien bedenken und ggf. prüfen. Zur leichteren Orientierung werden auch die **Dateinamen und teilweise Pfade** der FBS-Dateien des exportierten FBS-Netzes in die RailML-Datei geschrieben. Die dazugehörigen Pfade werden nur exportiert, wenn es ein relativer Pfad (relativ zur Netzdatei oder mit Platzhalter beginnend) sind; absolute Pfade (beginnend mit Laufwerksbuchstaben oder Netzwerk-Ressource) werden nicht ausgegeben.

Hinweise zur Nutzung der Schnittstelle für FBS-Anwender

Folgende Voraussetzungen sind FBS-intern notwendig, um Daten nach RailML exportieren zu können:

- Das ganze FBS-Netz muss eine einheitliche Fahrplanperiode haben.
- Triebfahrzeug-, Wagen- und Zuggattungsverzeichnis sowie die Standard-Bahnverwaltung müssen einheitlich im Netz sein.
- Alle Betriebsstellen müssen eine eindeutige Abkürzung haben.
- Jeder Wechsel einer Streckennummer muss auf einer Betriebsstelle liegen. Nötigenfalls muss an der Stelle des Streckennummernwechsels eine „Pseudo-Betriebsstelle“ vom Typ *Streckenwechsel* eingefügt oder der Streckennummernwechsel zu einer benachbarten Betriebsstelle verschoben werden.
- Die Züge und Zugteile müssen eindeutig im Netz sein, d. h. es dürfen keine Zugnummern ohne Mehrfachzuglaufkennung oder Zugteilnummern mehrfach vorkommen. Bei mehreren Zügen mit gleicher Zugnummer (Ergänzungsfahrpläne/Alternativtrassen) müssen Mehrfachzuglauf-Kennungen verwendet werden. Bei mehreren Zügen mit gleicher Zugnummer, die einen unterschiedlichen Planungsstand enthalten (z. B. bestellte Trasse <-> angebotene Trasse), verwenden Sie beim Export ggf. den Filter nach Ebene (s. u.), um die Eindeutigkeit sicherzustellen.

Neben den technisch bedingten Mindestanforderungen gibt es eine Reihe von weiteren Anforderungen, die eventuell aus dem weiterverarbeitenden (lesenden) Programm resultieren. Solche Anforderungen sind vor dem Export – am besten bereits beim Erstellen der Daten im FBS – zu bedenken. Typischer Weise können hier vorkommen:

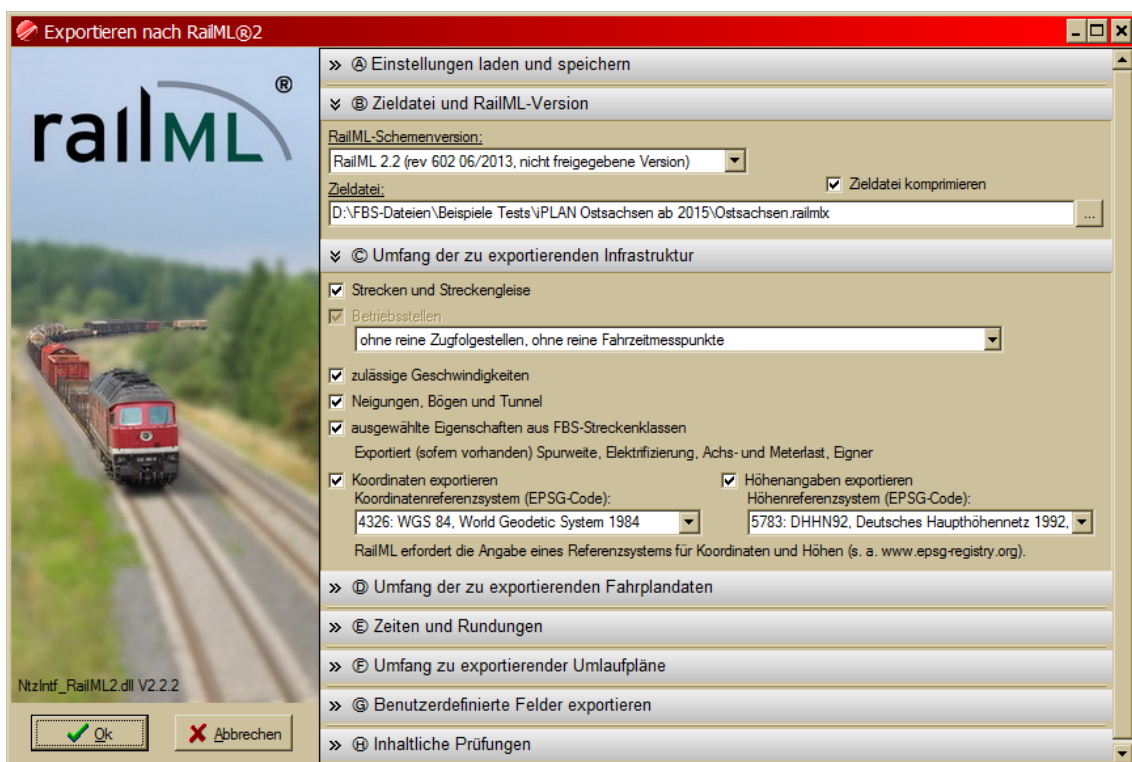
- Alle (zu exportierende) Zugteile müssen eine Linienbezeichnung haben. Viele Programme insbesondere aus dem Bereich der Fahrgastinformation benötigen eine Linienbezeichnung.
- Alle relevanten Betriebsstellen (d. h. zumindest die Zugangsstellen im Reiseverkehr bzw. die Halte Ihrer Züge) müssen eine Betriebsstellennummer haben. Viele lesende Programme identifizieren die Betriebsstellen nicht anhand deren Abkürzung – wie FBS – sondern anhand einer Nummer. Diese Nummer muss dann im FBS-Betriebsstellenverzeichnis unter „interne Bahnhofnummer“ eingegeben werden. Wir unterstützen Sie gern, falls Sie eine Liste mit solchen Nummern für eine größere Anzahl Betriebsstellen importieren müssen.
- Es ist möglich, dass das lesende Programm erfordert, dass Zugnummern, Zugteilnummern und/oder Linienbezeichnungen rein numerisch sein müssen. Sowohl in FBS als auch in RailML im Allgemeinen ist es hingegen möglich, dass diese Elemente (teilweise entgegen ihrem Titel „Nummer“) auch Buchstaben und andere Zeichen enthalten dürfen.
- Zu bedenken ist beispielsweise auch, dass der Inhalt bzw. die Einheit einiger für FBS optionaler Datenfelder nicht von vorn herein feststeht. So wird beispielsweise bei der Nummer, der Entgeltklasse oder den Koordinaten von Betriebsstellen von FBS kein spezielles Format vorausgesetzt. Ein lesendes Programm könnte jedoch erwarten, dass die Koordinaten z. B. im WGS84-System vorliegen.
- Es ist in RailML nicht möglich, dass eine Betriebsstelle, die an verschiedenen Strecken liegt, je Strecke verschiedene Eigenschaften hat. Es könnte z. B. sein, dass eine Betriebsstelle (mit der gleichen Abkürzung) an einer Strecke ein Bahnhof, an einer anderen (vmtl. parallelen) Strecke eine Abzweigstelle ist. Grundsätzlich sollten solche Betriebsstellen auch in FBS unterschiedliche Abkürzungen haben; das ist jedoch oft (in Deutschland z. B. aus Kompatibilität zu DB Netz) nicht möglich. Sie werden daher im Zweifelsfall beim Export gefragt, ob die Betriebsstelle mit der Vereinigungsmenge der Eigenschaften exportiert werden soll oder ob Sie den Export abbrechen wollen.

Einige der genannten Punkte können beim Export aus FBS nach RailML optional durch das Schnittstellenprogramm geprüft werden (s. a. Thema *Inhaltliche Prüfungen*).

Bedienung der Schnittstelle

Der Export von Daten nach RailML wird aus FBS-*iPLAN* oder FBS-*NETZ* heraus über den Menüpunkt **Datei** → **(Netz) Exportieren** → **RailML@2** ausgelöst. Der Menüpunkt ist verfügbar, wenn die FBS-RailML-Schnittstelle installiert wurde und für die aktuelle FBS-Lizenz freigeschaltet ist.

Zur Installation der Schnittstelle muss die Datei **NtzIntf_RailML2.dll** im FBS-Programmverzeichnis (normalerweise C:\Programme\FBS) liegen. Sie kann auch per Online-Update oder manuell dorthin kopiert werden – ein spezielles Installationsprogramm ist nicht zwingend erforderlich.

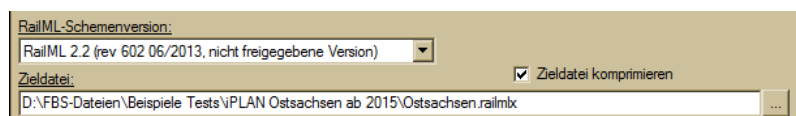


Zielfile und RailML-Version

Die RailML-Schemen unterliegen ständiger Weiterentwicklung. Gleichzeitig soll jedoch ein einmal exportierbarer Zustand möglichst über längere Zeit unverändert angeboten werden, etwa um eine gewisse „Investitionssicherheit“ auf der einlesenden Seite zu erhalten. Daher bietet die FBS-RailML-Schnittstelle den Export wählbar in mehrere rückliegende **RailML-Schemenversionen** an. Ältere Versionen enthalten i. d. R. inhaltliche Einschränkungen gegenüber neueren Versionen; diese sind unter www.irfp.de/deutsch/fbs/schnittstelle_railml_entwickler.html#HinwVerw zusammengestellt.

Als **Zielfile** geben Sie bitte einen Dateinamen mit Pfad und der Endung **.xml**, **.railML** bzw. **.railmlx** an, wobei die folgende Zuordnungsregel gilt:

- ***.xml** ist der Standard für RailML-Dateien bis einschl. Version 2.1 (Abwärtskompatibilität),
- ***.railml** ist der Standard für neuere, unkomprimierte RailML-Dateien,
- ***.railmlx** ist der Standard für komprimierte RailML-Dateien (gleich welcher Version).



Die Voreinstellung ist Pfad und Name der FBS-NETZ-Datei, jedoch mit angepasster Endung. Zur Voreinstellung anderer Pfade und/oder eines konkreten Dateinamens s. Anmerkungen unter *Einstellungen als Vorlage speichern* am Ende.

Das **Komprimieren** („packen“) von RailML-Dateien ist erst ab Version 2.2 genormt, kann FBS-seitig aber auch für ältere Versionen eingestellt werden. Komprimiert wird nach dem Zip-Datei-Standard mit Deflate-Algorithmus, d. h. solche Dateien können mit gängigen „Zip-Programmen“ bei Bedarf auch manuell wieder ausgepackt werden.

Umfang der zu exportierenden Infrastruktur

☑ © Umfang der zu exportierenden Infrastruktur

☑ Strecken und Streckengleise

☑ Betriebsstellen
ohne reine Zugfolgstellen, ohne reine Fahrzeitmesspunkte

☑ zulässige Geschwindigkeiten

☑ Neigungen, Bögen und Tunnel

☑ ausgewählte Eigenschaften aus FBS-Streckenklassen
Exportiert (sofern vorhanden) Spurweite, Elektrifizierung, Achs- und Meterlast, Eigner

☑ Koordinaten exportieren
Koordinatenreferenzsystem (EPSG-Code): 4326: WGS 84, World Geodetic System 1984

☑ Höhenangaben exportieren
Höhenreferenzsystem (EPSG-Code): 5783: DHHN92, Deutsches Haupthöhennetz 1992

RailML erfordert die Angabe eines Referenzsystems für Koordinaten und Höhen (s. a. www.epsg-registry.org).

Unter *Umfang der zu exportierenden Infrastruktur* können Sie in begrenztem Maße wählen, wieviel Infrastruktur-Informationen ausgegeben werden sollen. Sofern nur der eigentliche Fahrplan ausgegeben werden soll (z. B. zur Fahrgastinformation oder für Dienstpläne), können Sie die

Option **Strecken und Streckengleise** ausschalten.

Ein Mindestmaß an Infrastruktur – konkret Betriebsstellen – ist für RailML-Fahrplandaten jedoch immer notwendig, weshalb sich die Option **Betriebsstellen** nicht ganz ausschalten lässt. Um aber Dateigröße der tendenziell relativ großen RailML-Dateien einzusparen, können Sie die Betriebsstellen ausblenden, die für die meisten Anwendungen von Fahrplandaten wie Fahrgastinformation oder Dienstpläne typischer Weise nicht relevant sind: *Reine Zugfolgstellen* sind Blockstellen, Blocksignale oder Blockkennzeichen, die keine weitere Eigenschaft haben (also nicht gleichzeitig z. B. Haltepunkte sind). *Reine Fahrzeitmesspunkte* sind solche „Betriebsstellen“ im FBS-Sinne, die weder betriebliche noch verkehrliche Eigenschaften haben (z. B. Bahnverwaltungsgrenzen, Streckenwechsel). Betriebsstellen, an denen Strecken oder Züge beginnen oder enden, werden in jedem Falle exportiert. Betriebsstellen mit Verkehrshalten von Zügen dürfen durch den Filter nicht übersprungen (ausgeschlossen) werden.

Die Optionen **zulässige Geschwindigkeiten**, **Neigungen, Bögen und Tunnel** sowie **ausgewählte Eigenschaften aus FBS-Streckenklassen** sind Infrastruktur-Daten, deren Export etwa in folgenden Szenarien in Betracht kommt:

- Mitgabe grundlegender Infrastrukturdaten mit Fahrplandaten, etwa um Buchfahrpläne oder ähnliche (elektronische) Unterlagen darstellen zu können.
- Austausch eines Grunddatenbestands an Infrastruktur zur Vermeidung wiederholter manueller Dateneingabe. Da unterschiedliche Programme jedoch i. d. R. auch unterschiedliche Datenmodelle haben, wird kaum eine vollständige Übernahme möglich sein, sondern ein manuelles Ergänzen zumindest im Detail unvermeidbar bleiben.

Für einen allgemeinen Austausch von Fahrplandaten – für den die FBS-RailML-Schnittstelle hauptsächlich entwickelt wurde – sind diese Optionen nur selten notwendig.

Koordinaten einschl. Höhen sind in RailML ab Version 2.2 nur unter Angabe des jeweiligen Referenzsystems zulässig (bis RailML 2.1 können Sie Koordinaten noch ohne Referenzsystem ausgeben). Für FBS ist das Referenzsystem nicht bedeutsam (Koordinaten werden in FBS nur relativ zueinander verwendet), daher kennt FBS die Angaben nicht; Sie müssen sie ggf. vor dem Export ergänzen. Geo-Koordinaten sind typischer Weise von WGS84; Bahnhofshöhenangaben in Deutschland vom Deutschen Haupthöhennetz 1912. Da dieses jedoch keine EPSG-Nummer hat, müssen Sie u. U. ersatzweise das Deutsche Haupthöhennetz 1992 angeben.

Umfang der zu exportierenden Fahrplandaten – verkehrliche und betriebliche Züge

Unter *Umfang der zu exportierenden Fahrplandaten* ist eine Anpassung der Daten möglich, die im RailML-Teilschema <timetable> definiert sind.

RailML unterscheidet bezüglich der Züge zwischen betrieblichen (*operational*) und verkehrlichen Zügen (*commercial trains*). Die **betrieblichen Züge** sind grundlegend und werden immer exportiert. Sie stellen die Sicht des Infrastrukturbetreibers dar, d. h. zu jedem Zeitpunkt kann auf jedem Streckenabschnitt nur *ein* betrieblicher Zug verkehren.

Im Gegensatz dazu stellen die **verkehrlichen Züge** die Sicht des Reisenden auf den Fahrplan dar. Unter gewissen Umständen können mehrere verkehrliche Züge gleichzeitig (in einem betrieblichen Zug) verkehren (Flügeln, Kurswagen).

Die Ausgabe dieser Daten ist in gewissem Sinne redundant zu den betrieblichen Zügen und optional. Sie ist jedoch Basis für die meisten Folgeanwendungen (insbesondere z. B. für die Fahrgastinformation) und in solchen Fällen zwingend erforderlich.

☑ Umfang der zu exportierenden Fahrplandaten

Betriebliche Züge ("operational trains") exportieren

Verkehrliche Züge ("commercial trains") exportieren

Verkehrliche Züge enthalten nahezu die gleichen Daten wie betriebliche Züge, jedoch aus der Sicht von Reisenden. Ein Unterschied ergibt sich z. B. beim Flügeln und bei Kurswagenverbindungen.

Direktverbindungen mit untergeordneten Verstärkern
Reduzierte Darstellung, die sich an Tabellenfahrplänen orientiert: Ein verkehrlicher Zug = eine Spalte im Tabellenfahrplan. Für die meisten Anwendungsfälle der Fahrgastinformation ausreichend.

Direktverbindungen und Verstärker als unabhängige „commercial trains“
Vollständige Darstellung, bei der jeder verkehrliche Zug genau einem Zugteil aus FBS entspricht. Notwendig für Re-Import der Daten in Planungssysteme wie FBS.

Direktverbindungen beschriften mit (RailML-Attribut "name"):

Zugteilnummer ohne Kleinbuchstaben

Für die verkehrlichen Züge gibt es je nach Anwendungsfall folgende Möglichkeiten:

- Sie entsprechen üblicher Weise den **Direktverbindungen** aus FBS, die Basis für Tabellen- und Aushangfahrplan sind (s. a. iPLAN/NETZ: Auswertung → Übersicht Züge → Direktverbindungen). Verkehrliche Züge dieser Art unterscheiden sich nur dann von betrieblichen, wenn Flügeln, Kurswagen oder andere Fahrzeugwechsel „im Spiel“ sind – eben wenn sich *Direktverbindungen* im Sinne umsteigefreier Verbindungen über betriebliche Züge hinweg ergeben.
- In selteneren Fällen zählen neben den Direktverbindungen auch *Verstärker* als verkehrliche Züge – „Verstärker“ hier im Sinne von Zugteilen, die zur rein verkehrlichen Verstärkung (höheres Sitzplatzangebot) verkehren ohne dass sie irgendwo zusätzliche umsteigefreie Verbindungen darstellen, die ohne sie nur durch Umsteigen möglich wären.

Für die meisten Anwendungsfälle der Fahrgastinformation genügt Fall (a). Fall (b) wird vor allem dann benötigt, wenn es um eine 1:1-Wiederherstellung der genauen Zugteile aus FBS geht, vor allem um einen Re-Import der Zugdaten in FBS.

Beim Export verkehrlicher Züge müssen Sie in jedem Falle entscheiden, wie das RailML-Attribut *name* der Direktverbindungen gesetzt werden soll. Detailinformationen zum Unterschied der drei Optionen finden Sie im Dokument

www.irfp.de/deutsch/service/faq/beschriftung_von_direktverbindungen.pdf

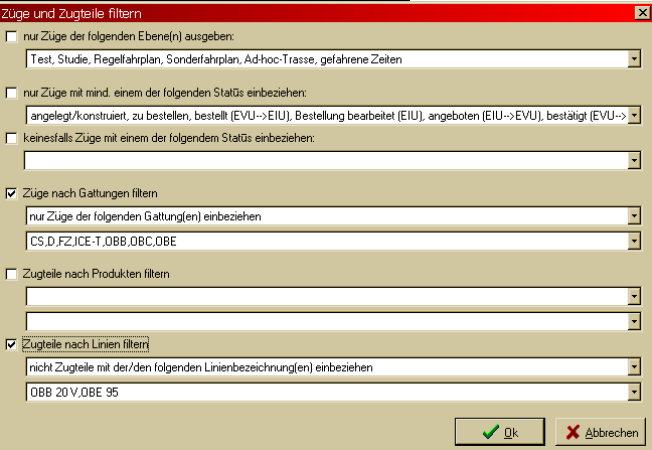
Falls Verstärker als unabhängige *commercial trains* exportiert werden (Fall b), erhalten diese kein Attribut *name* – die Beschriftung bezieht sich immer nur auf Direktverbindungen.

Umfang der zu exportierenden Fahrplandaten – Filter für Züge und Zugteile

alle Züge und Zugteile exportieren
 Züge und Zugteile filtern
 nur die Züge und Zugteile exportieren, die in zu exportierenden Umlaufplänen vorkommen

Mit Zügen und Zugteilen filtern können Sie den Umfang der zu exportierenden Züge bestimmen. Beispielsweise lässt sich hier festlegen, dass Züge anderer EVU, die Sie in Ihren Bildfahrplänen nur informativ aufgenommen haben, nicht exportiert werden sollen. Diese Einstellungen werden im Folgenden als **Filter** bezeichnet.

Insgesamt besteht die Regel, dass ein Zug, Zugteil oder eine Direktverbindung durch einen Filter nicht „auseinandergerissen“ werden darf, d. h. ein Filter muss immer einen vollständigen Zuglauf erfassen und nicht nur einen Teil davon. Wenn z. B. ein Zug in einem ersten Abschnitt mit der Zuggattung **R** verkehrt, im zweiten Abschnitt mit **E** und im dritten Abschnitt wieder mit **R** und man einen Filter einstellen würde, der die Zuggattung **E** ausschließt (oder *nur R* einschließt), dann würde der Zug durch den Filter in zwei unabhängige Züge „zerrissen“ werden – die dann aber zwangsläufig die gleiche Zugnummer hätten. In der exportierten RailML-Datei würde dies gegen die Regel verstoßen, dass jeder Zug eine eindeutige Zugnummer haben muss und daher wie eine „Primärschlüsselverletzung“ aussehen.



Insgesamt besteht die Regel, dass ein Zug, Zugteil oder eine Direktverbindung durch einen Filter nicht „auseinandergerissen“ werden darf, d. h. ein Filter muss immer einen vollständigen Zuglauf erfassen und nicht nur einen Teil davon. Wenn z. B. ein Zug in einem ersten Abschnitt mit der Zuggattung **R** verkehrt, im zweiten Abschnitt mit **E** und im dritten Abschnitt wieder mit **R** und man einen Filter einstellen würde, der die Zuggattung **E** ausschließt (oder *nur R* einschließt), dann würde der Zug durch den Filter in zwei unabhängige Züge „zerrissen“ werden – die dann aber zwangsläufig die gleiche Zugnummer hätten. In der exportierten RailML-Datei würde dies gegen die Regel verstoßen, dass jeder Zug eine eindeutige Zugnummer haben muss und daher wie eine „Primärschlüsselverletzung“ aussehen.

FBS kann beim Export nicht alle Filterkombinationen auf Zulässigkeit prüfen – Sie sind für die korrekte Anwendung der Filter selbst verantwortlich. Bitte gehen Sie daher mit Bedacht vor, insbesondere wenn Sie mehrere Filter gleichzeitig anwenden. Wenn Sie mehrere Filter verwenden, werden diese logisch-UND-verknüpft, d. h. ein Zug bzw. Zugteil wird nur dann ausgegeben, wenn alle gesetzten Filter gleichzeitig zutreffen. Ein Zug kann dann z. B. auf einem Abschnitt auf Grund der Verkehrstage und auf einem anderen auf Grund der Zuggattung ausgeschlossen werden.

Bitte beachten Sie auch die im Kapitel zu Umlaufplänen genannten Zusammenhänge zwischen dem Export von Umlaufplänen und Filtern.

Insgesamt muss ein eventuell detaillierteres Filtern beim Import der Daten erfolgen und nicht beim Export. Da RailML ein allgemeingültiges Schnittstellenformat ist, müssen die exportierten Daten unabhängig vom „Zielprogramm“ sein. Nur die lesende Schnittstelle kennt den genauen Umfang der benötigten Daten und weiß, was im Zieldatenmodell als Primärschlüssel gilt. Nur das lesende Programm kann daher entscheiden, welche Daten überlesen werden können und welche ausgewertet werden müssen.

Umfang der zu exportierenden Fahrplandaten – Bremsen und Zugbeeinflussung

Bremseinstellungen und **Zugbeeinflussung** gehören zu den grundsätzlich in RailML definierten und damit exportierbaren Daten, werden

jedoch nicht in allen Fällen von der lesenden Software ausgewertet (insbesondere nicht zur Fahrgastinformation). Wenn sie exportiert werden, muss auch ein Wechsel der Brems- oder Zugbeeinflussungseinstellungen während des Zuglaufs (an einem Unterwegsbahnhof) ordnungsgemäß beschrieben werden. Dies kann dazu führen, dass ein Zuglauf in RailML „gebrochen“ (in Abschnitte unterteilt) wird, allein weil solche Einstellungen wechseln – und damit

Bremseinstellungen exportieren
 Schreibt <trainPartSequence>.<brakeUsage> der betrieblichen Züge. Das Abschalten dieser Option vermeidet, dass Zugläufe allein wegen wechselnder Bremseneinstellungen unterbrochen werden.

Zugbeeinflussung exportieren
 Schreibt <trainPartSequence>.<formationTT>.<equipmentUsage> der betrieblichen Züge. Das Abschalten dieser Option vermeidet, dass Zugläufe allein wegen wechselnder Zugbeeinflussung unterbrochen werden.

die RailML-Datei komplexer und größer wird als nötig. Um dies zu vermeiden, sollten Sie die Optionen nur dann einschalten, wenn die jeweilige Information auch ausgewertet wird.

Zeiten und Rundungen

In Fahrplänen werden Zeiten üblicher Weise in ganzen Minuten (für verkehrliche Zwecke) oder Zehntelminuten (für betriebliche Zwecke) vorgehalten. FBS braucht intern jedoch eine höhere Genauigkeit, damit es bei

hohen Geschwindigkeiten und dicht liegenden Betriebsstellen (insbesondere Zugfolgestellen wie Blocksignalen) nicht zu kumulierenden Rundungsfehlern kommt. RailML erlaubt jede denkbare Genauigkeit zwischen ganzen Minuten bis hin zu Sekundenbruchteilen, die der internen Genauigkeit von FBS entsprechen. Je nach Anwendungsfall sollten Sie eine der folgenden Optionen wählen:

- **Zehntelminuten ausgeben** sollte im Zweifel der für Fahrpläne übliche Normalfall sein.
- **Interne Genauigkeit** (Sekundenbruchteile) ist notwendig, falls ein exakter Re-Import der Daten in FBS oder eine vergleichbare Fahrplankonstruktions-Software beabsichtigt ist. Diese Einstellung vermeidet Rundungsfehler durch den Datenaustausch.
- **andere Rundungsregel – auf ganze Minuten runden** kann in Frage kommen, falls ein Datenexport zur Fahrgastinformation stattfindet und die einlesende Software nicht selbst runden kann, d. h. ganze Minuten erwartet. Um Widersprüche etwa zu mit FBS erstellten Tabellen- oder Aushangfahrplänen zu vermeiden, sollten Sie die gleiche Rundungsregel einstellen wie in diesen Unterlagen – z. B. *alles abrunden* oder *verkehrlich*.

Allerdings sind andere Einstellungen als *Zehntelminuten* und *interne Genauigkeit* ausdrücklich nicht empfohlen, insbesondere nicht die Einstellung *Sekunden ausgeben*, die unnötige Rundungsfehler und Abweichungen zu anderen Fahrplanunterlagen provoziert. (Diese Einstellung ist nur zur Abwärtskompatibilität mit älteren RailML-Dateien vorhanden und kann in einer zukünftigen Version der Schnittstelle ganz entfallen.)

Unter **Züge mit Fahrzeitdifferenzen und Zuschlägen** exportieren können Sie festlegen, ob die RailML-`<timetable>`-Struktur `<sectionTT>.<runTimes>` ausgegeben werden soll. Davon abhängig werden für jeden einzelnen Streckenabschnitt jedes Zuges folgende Informationen über Bestandteile der Fahrzeit ausgegeben:

- das Attribut `<runTimes>.minimalTime` mit der eigentlichen („physikalischen“) Fahrzeit,
- das Attribut `<runTimes>.operationalReserve` mit dem linearen Fahrzeitzuschlag als Zeitwert,
- Attribut `<runTimes>.additionalReserve` mit einem eventuellen nicht-linearen Fahrzeitzuschlag als Zeitwert (sofern vorhanden),
- Attribut `<sectionTT>.percentageSupplement` mit dem linearen Fahrzeitzuschlag als Prozentwert.

Das Abschalten dieser nicht immer benötigten Informationen spart Dateigröße. Sie können diese Informationen nur dann ausgeben, wenn unter Infrastruktur *alle Betriebsstellen* ausgewählt wurden.

Weitere Informationen aus `<sectionTT>` über die benutzten Streckengleise (Regel- oder Gegengleis) und die zurückzulegende Entfernung werden immer ausgegeben, wenn Sie unter Infrastruktur die Option *Strecken und Streckengleise* eingeschaltet haben.

Umlaufpläne und Gültigkeitszeiträume

Wenn Sie mit der RailML-Datei auch **Umlaufpläne exportieren** wollen, können Sie jeden im FBS-Netz enthaltenen Umlaufplan ein- oder ausschließen.

☑ ☒ Umfang zu exportierender Umlaufpläne

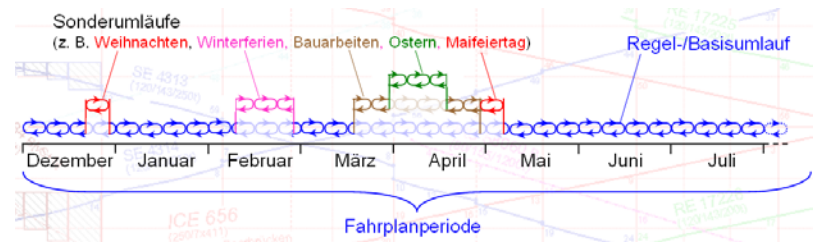
Es werden nur die Umlaufpläne exportiert, die in der folgenden Tabelle eingeschaltet sind.

Die Gültigkeit eines Umlaufplans in der RailML-Datei kann je nach Anwendungsfall gegenüber der Gültigkeit im FBS angepasst sein. Insbesondere können im FBS nur einmal vorhandene Musterwochen auf mehrere Wochen ausgedehnt werden.

Zum Bearbeiten eines Gültigkeitszeitraums klicken Sie mit der rechten Maustaste auf einen Umlaufplan.

Umlaufplan	gültig im FBS von - bis	gültig im Export von - bis
<input checked="" type="checkbox"/> Tfz 650	14.12.2020-20.12.2020	13.12.2020-11.12.2021

Sofern die Fahrplanperiode des FBS-Netzes definiert ist, trägt auch jeder Umlaufplan eine Gültigkeitsperiode. Im einfachsten Fall ist das die gesamte Fahrplanperiode des Netzes, es kann aber auch nur ein begrenzter (Teil-)Zeitraum der Fahrplanperiode sein. Es kann daher mehrere Umlaufpläne für die gleiche Fahrzeugbaureihe geben. I. d. R. ist ein **Grund- oder Regelumlauf** für das Fahrplanjahr vorhanden, welcher ununterbrochen ohne konkreten Datumsbezug gilt. Dieser kann für bestimmte Zeiträume (Weihnachten, Ostern, Ferien, Bauarbeiten) durch **Sonderumlaufpläne** überschrieben werden.



Die Teilzeiträume von Umläufen für die gleichen Fahrzeuge sollten disjunkt sein, d. h. sie sollten sich nicht überlappen - ansonsten kann das lesende Programm u. U. nicht feststellen, welcher Umlauf zu welchem Zeitpunkt gelten soll.

Als **Voreinstellung** der Gültigkeit von zu exportierenden Umlaufplänen dient die Anzeige des Datums der Wochentage als Indiz: Umlaufpläne für eine konkrete Woche, bei denen die Anzeige des Datums **ausgeschaltet** ist, werden als Regelumlauf per Voreinstellung ganzjährig exportiert.

Um einen Gültigkeitszeitraum eines Umlaufplans für den Export zu ändern, klicken Sie mit der rechten Maustaste auf den Umlaufplan. Sie können im Kalender-Fenster („gültig im Export von - bis“) beliebig viele voneinander unabhängige Zeiträume anwählen, in denen der Umlauf wiederholt gültig ist. Die Gültigkeit des Umlaufs in FBS wird dadurch nicht beeinflusst.

Wenn Sie einem Umlaufplan auf diese Art eine längere Gültigkeit geben, als er in FBS hat, müssen Sie jedoch selbst darauf achten, dass die im Umlaufplan vorkommenden Züge auch an allen markierten Tagen verkehren. (Die Verkehrszeiträume der Züge werden nicht etwa automatisch auf die Gültigkeit des Umlaufs erweitert - wodurch sich u. a. Trassenkonflikte ergeben könnten.)

gültig im Export von - bis

Name des Verkehrsgebietes (z.B. "Sommerferien")

Der Bereich umfasst folgende Tage:
15.2.-28.2.; 22.3.-28.3.; 14.4.-28.4.

Buttons: Ok, Abbrechen, Zurücksetzen

Bereiche einfügen | veränd. Feiertage

Doppelklicken Sie auf einen Tag, um diesen ein- bzw. auszuschließen. Halten Sie die Shift-Taste gedrückt zum Auswählen eines Bereiches.

	Dezember 2009	Januar 2010	Februar 2010	März 2010	April 2010	Mai 2010
Mi	7 14 21 28	4 11 18 25	1 8 15 22	1 8 15 22 29	5 12 19 26	3 10 17 24 31
Di	1 8 15 22 29	5 12 19 26	2 9 16 23	2 9 16 23 30	6 13 20 27	4 11 18 25
Mi	2 9 16 23 30	6 13 20 27	3 10 17 24	3 10 17 24 31	7 14 21 28	5 12 19 26
Do	3 10 17 24 31	7 14 21 28	4 11 18 25	4 11 18 25	1 8 15 22 29	6 13 20 27
Fr	4 11 18 25	1 8 15 22 29	5 12 19 26	5 12 19 26	2 9 16 23 30	7 14 21 28
Sa	5 12 19 26	2 9 16 23 30	6 13 20 27	6 13 20 27	3 10 17 24	1 8 15 22 29
So	6 13 20 27	3 10 17 24 31	7 14 21 28	7 14 21 28	4 11 18 25	2 9 16 23 30

Bitte achten Sie auf folgenden Zusammenhang zwischen Umlaufplänen und Filtern:

Vorrang von Umlauf-Zugteilen vor Filtereinstellungen: Es werden immer alle Zugteile exportiert, die in einem zu exportierenden Umlaufplan vorkommen – unabhängig von etwaigen Zug-/Zugteil-Filter-Einstellungen. Wenn also z. B. ein Zugteil der Linie „21“ nicht exportiert werden soll und Sie dafür einen Filter *nicht Zugteile mit der Linienbezeichnung... 21* gesetzt haben, dieser Zugteil aber in einem zu exportierenden Umlaufplan vorkommt, dann wird er trotzdem exportiert!

Wenn Sie den Export über **Zeitraum einschränken** (unter *Umfang der zu exportierenden Fahrplandaten*) nicht für die gesamte Gültigkeitsperiode des Netzes durchführen, gelten folgende Regeln:

- Kein Umlaufplan darf außerhalb des gewählten Export-Zeitraumes gültig sein. Die Export-Gültigkeit der Umlaufpläne wird deshalb auf die Schnittmenge mit dem gewählten Export-Zeitraum begrenzt.
- Der gewählte Export-Zeitraum darf nie über die Gültigkeit der Fahrplanperiode des Netzes hinausgehen.
- Ein Umlaufplan darf über seine in FBS definierte Gültigkeit hinaus exportiert werden, sofern er im gewählten Export-Zeitraum liegt. Es gilt dann wie oben genannt, dass Sie selbst dafür verantwortlich sind, dass die Züge an den betreffenden Tagen auch verkehren.

Export von Rangierfahrten im Umlaufplan: Sollte ein Zugteil eines zu exportierenden Umlaufplans nicht in den Bildfahrplänen aufzufinden sein, dann wird der Export mit einer Fehlermeldung abgebrochen, da davon auszugehen ist, dass es sich um einen bisher unerkannten Fehler im Umlaufplan handelt (vergessen abzugleichen). In manchen Fällen sind solche (nicht im Bildfahrplan geplanten) Fahrten jedoch gewollt, weil es sich z. B. um Rangierfahrten zwischen Bahnhofsteilen handelt. Sie können mit der Eigenschaft **Fahrt nicht abgleichen** explizit vorgesehen werden. Diese Eigenschaft kann auf bequeme Art im Fenster *Umlaufplan abgleichen* mit der rechten Maustaste für mehrere/alle betreffenden Fahrten gesetzt werden.

Für den RailML-Export gilt: Mit der Option *Fahrt nicht abgleichen* bewirkt man nur, dass für den Zugteil ein „Platzhalter“ im Umlaufplan exportiert wird, *falls er nicht in den Bildfahrplänen aufzufinden ist*. Falls er jedoch trotz gesetzter Option aufzufinden ist (etwa weil nur irgendwelche Eigenschaften nicht abgeglichen werden sollten), dann wird trotzdem *kein* Platzhalter exportiert, sondern der reguläre Bildfahrplan-Zugteil.

Mit der Option **Innere Wochentagsgruppen exportieren** wird der RailML-Export der Umlaufpläne um zusätzliche Informationen erweitert. Diese Informationen sollen es einem lesenden Programm ermöglichen, sich wiederholende Fahrtenabfolgen z. B. an Di-Do auf *einem* Blatt darzustellen, anstatt etwa für Di, Mi und Do jeweils getrennte identische Blätter zu verwenden. Voraussetzung ist natürlich, dass es überhaupt identische Wochentage im Umlaufplan gibt. (Sinnvoll ist, bereits beim Erstellen des Umlaufplans die entsprechende Option im FBS-Umlauf-Modul einzuschalten, wodurch geprüft und überwacht werden kann, dass es identische innere Wochentage gibt.)

Da in den originalen RailML-Strukturen keine Attribute für sich wiederholende Wochentage vorgesehen sind, wird mit dieser Option die entstehende RailML-Datei um einen weiteren Namensraum erweitert. Dieser Namensraum definiert die zusätzlich notwendigen Attribute.

Benutzerdefinierte Felder exportieren

Benutzerdefinierte Felder (BDF) dienen im FBS dem Abbilden von Daten, die für die eigentliche FBS-Funktion nicht von Bedeutung sind, aber im Umfeld von Fahrplandaten auftreten – etwa vertragliche Informationen wie z. B. Vertragsnummern, Vertragspartner, Aufgabenträger. Gerade weil diese Informationen in FBS selbst nicht von Bedeutung sind, kommen sie umso mehr (oder gar nur) für den Datenaustausch in Schnittstellen in Frage.

Die FBS-RailML-Schnittstelle lässt sich dahingehend um benutzerdefinierte Informationen individuell erweitern. Das RailML-Datenformat bietet folgende Möglichkeiten, zusätzliche Informationen abzubilden:

- in einem RailML-Standard-Datenfeld (Attribut), das von FBS sonst nicht verwendet wird. In Frage kommen hier auf Zugteil-Ebene die Attribute *remarks*, *debitcode* und *operator* sowie die zentrale Struktur `<organizationalUnits>` mit möglichen Einträgen für *infrastructureManager*, *vehicleManufacturer*, *vehicleOperator*, *customer*, *railwayUndertaking*, *operationalUndertaking*, *concessionaire* und *contractor*.
- in einer Erweiterung des RailML-Standards um zusätzliche Datenfelder (Attribute). Dies kommt in Frage, wenn das benutzerdefinierte Feld nicht auf Zugteil-Ebene auftritt oder die Standard-Felder in irgendeiner Art unzutreffend oder nicht verwendbar sind. Der Erweiterung sind fast keine Grenzen gesetzt; allerdings erfordert eine solche Erweiterung die Definition eines eigenen Namensraums in der RailML-Datei. Der Namensraum beschreibt die zusätzlichen Attribute; eventuell ist zusätzlich die Fixierung der Attribute in einer sogenannten Schemendatei (*.xsd-Datei) notwendig.

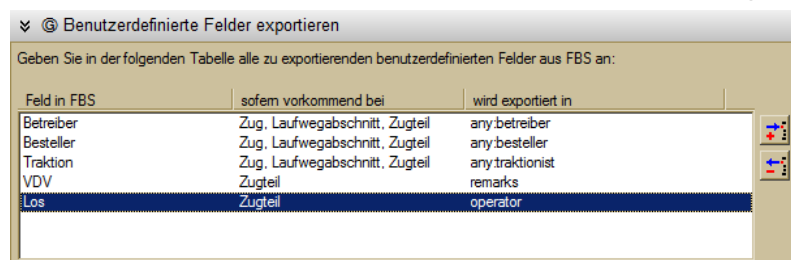
Auch hier gilt einmal mehr: Stimmen Sie solche Einstellungen mit der einlesenden Seite ab; nur wenn die Gegenseite die benutzerdefinierten Felder auch einlesen kann, ist der Export überhaupt sinnvoll.

In den Exporteinstellungen der FBS-RailML-Schnittstelle können Sie Zuordnungen treffen zwischen in FBS vorkommenden BDF einerseits und RailML-Attributen andererseits.

BDF können in FBS am Zug, in einem Laufwegabschnitt oder am Zugteil vorkommen. Diesen drei Stellen sind fix folgende Entsprechungen in RailML zugewiesen:

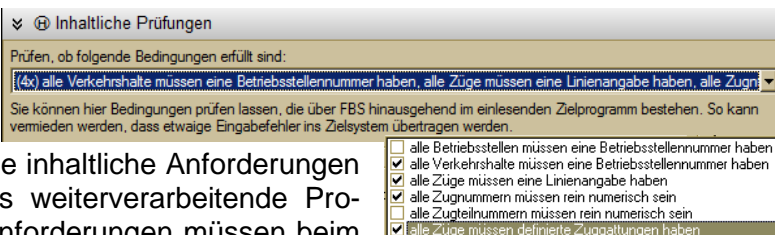
Zug	→ <code><train></code> (hier jedoch nur <i>operational trains</i>)
Laufwegabschnitt	→ <code><trainPartSequence></code>
Zugteil	→ <code><trainPart></code>

Zu beachten ist noch, dass das Standard-Datenfeld *debitcode* seitens RailML auf einen Zahlenwert festgelegt ist (nur Ziffern sind hier zulässig). FBS prüft diese Bedingung nicht. Abgesehen von dieser Ausnahme und den etwas umständlichen `<organizationalUnits>` stehen mit den beiden Feldern *remarks* und *operator* zumindest auf Zugteil-Ebene zwei relativ freizügig und unkompliziert verwendbare Datenfelder zur Verfügung.



Inhaltliche Prüfungen

Wie eingangs erwähnt, können bestimmte über das technische Mindestmaß hinausgehende inhaltliche Anforderungen an die RailML-Daten durch das weiterverarbeitende Programm gestellt werden. Diese Anforderungen müssen beim Eingeben der Daten in FBS beachtet werden. Um leichter sicherzustellen, dass die RailML-Daten bestimmte Anforderungen erfüllen, können während des Exports typische Prüfungen eingeschaltet werden. Wenn eine der zu prüfenden Bedingungen verletzt wird, wird der Export mit einem entsprechenden Hinweis abgebrochen.



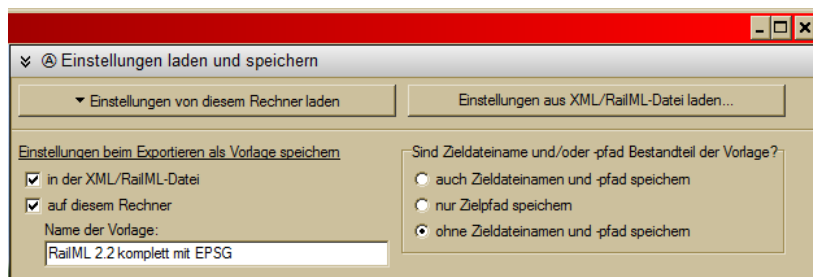
Bitte beachten Sie, dass es sich hierbei nur um Prüfungen handelt, die der Vermeidung von Folgefehlern bzw. der zeitigeren Fehleroffenbarung dient - die Schnittstelle ergänzt oder manipuliert nicht etwa die zu exportierenden Daten!

Für viele Anwendungen der RailML2-Schnittstelle sind die abgebildeten Einstellungen typisch. Sollten Sie regelmäßig mit den gleichen Bedingungen exportieren, empfehlen wir das Speichern der Prüfungseinstellungen als Vorlage (s. u.).

Speichern der Einstellungen als Vorlage

Sie können die vorgenommenen **Einstellungen als Vorlage speichern**. Dabei werden auch die netzabhängigen Einstellungen wie z. B. Umlaufpläne und ein Gattungs-, Produkt- oder Linienfilter gespeichert. Allerdings werden diese Einstellungen beim späteren Export eines anderen Netzes nur dann angewendet, wenn sie tatsächlich zutreffen können.

Ob auch der Dateiname oder -pfad als Voreinstellung gespeichert wird, können Sie mit im rechten Bereich auswählen. Bei *auch Zieldateinamen und -pfad speichern* wird der aktuelle Dateiname bei jedem zukünftigen Export fix eingestellt;

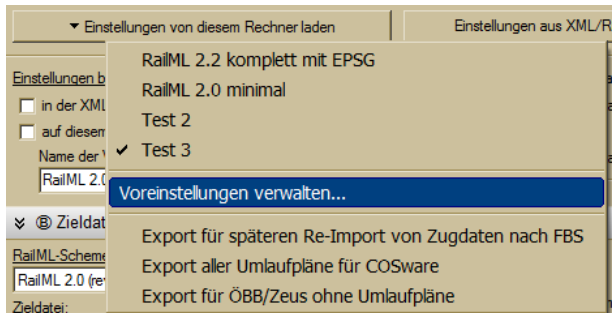


das Eingabefeld *Zieldatei* ist damit überhaupt nicht mehr abhängig vom aktuellen Dateinamen der Netzdatei. Bei *nur Zielpfad speichern* wird nur der aktuelle Pfad gespeichert – der Dateiname selbst bleibt abhängig vom Namen der Netzdatei. Es würde damit immer in den gleichen Zielpfad (z. B. ein RailML-Übergabeverzeichnis), jedoch mit verschiedenen RailML-Dateinamen exportiert. Bei *ohne Zieldateinamen und -pfad speichern* ist die Voreinstellung des Eingabefelds *Zieldatei* der Pfad und Dateinamen der aktuellen Netzdatei (jedoch mit anderer Endung).

Wenn Sie die Einstellungen **in der XML/RailML-Datei** speichern, können Sie beim nächsten Export wieder die zuletzt erfolgreich exportierte Datei als Quelle angeben (Knopf *Einstellungen aus XML/RailML-Datei laden*). Damit die Einstellungen in der RailML-Datei gespeichert werden können, muss aus formellen Gründen ein zusätzlicher Namensraum (eben für die FBS-Export-Einstellungen) in der Datei ergänzt werden. Das lesende Programm muss das korrekt „überlesen“ können.

Wenn Sie die Einstellungen **auf diesem Rechner** speichern wollen, müssen Sie der Vorlage einen eindeutigen Namen geben. Sie kann dann unter diesem Namen wieder geladen werden (Knopf *Einstellungen von diesem Rechner laden*). Diese Vorlagen werden in der Datei *NtzIntf_RailML2.xmlini* gespeichert, und zwar in dem Pfad, der in der **FBS-Konfigurationsverwaltung** für anwenderbezogene Konfigurationen eingestellt ist. Um die

Datei zu finden, rufen Sie im FBS-Startfenster *Installation und Reparatur* → *FBS-Konfigurationsverwaltung* → *Konfigurationsdateien* auf. Sie können diese Datei bei „Umzug“ auf einen anderen Rechner mitnehmen.



Zusätzlich zu den selbst erstellten Vorlagen gibt es unter *Einstellungen von diesem Rechner laden* eine Reihe fest vordefinierter Vorlagen. Unter **Voreinstellungen verwalten** können Sie selbst erstellte Vorlagen löschen, umbenennen oder die Reihenfolge ändern.