

## Abstimmung von Voreinstellungen für Datenexport über die FBS-RailML-Schnittstelle

Beim Export von FBS-Daten können Einstellungen in Bezug auf die zu exportierenden Daten vorgenommen werden. Hierdurch können die Daten beim Export schon an die Erfordernisse eines anschließenden Importes angepasst werden. Um dem Anwender den Export zu erleichtern und Problemen beim Import, die durch nicht richtig gewählte Einstellungen entstehen, vorzubeugen, bietet iRFP seinen Schnittstellenpartnern an, programmspezifische Voreinstellungen zu hinterlegen. So braucht der Anwender nur auszuwählen, mit welchem Programm die Daten im Nachhinein wieder importiert werden sollen; die entsprechenden Einstellungen und eventuelle zusätzliche Prüfungen werden dann automatisch beim Export vorgenommen. Unserer Erfahrung nach ist dies eine Lösung mit Vorteilen für alle drei Beteiligten: Einfachere und schnellere Handhabung für den Anwender, weniger Rückfragen und Datenanalysen für die Softwarehersteller.



Um die richtigen Voreinstellungen für Ihre Zielsoftware hinterlegen zu können, ist die Beantwortung einiger Verständnisfragen seitens iRFP über Struktur und Auswertung der Daten beim Import in Ihrem Programm notwendig. Daher möchten wir Sie bitten, den nachfolgenden Fragebogen auszufüllen.

Vielen Dank für Ihr Mitwirken!

**Wie soll die Voreinstellung für den Datenexport aus FBS zu Ihrer Software dem FBS-Anwender gegenüber beschriftet werden?**

(Bitte wählen Sie eine nicht zu lange, aussagekräftige und eindeutige Beschriftung, die für einen Menüpunkt geeignet ist.)

Der nachfolgende Fragebogen dient nur als Orientierung für die möglichen Einstellungen. Sie können uns die Informationen gern auch individuell auf andere geeignete Art zukommen lassen.

Wenn Ihre Software mehrere Anwendungsfälle unterstützt, die unterschiedliche railML-Datenstrukturen benötigen, füllen Sie bitte je Voreinstellung einen Fragebogen aus.

## Allgemeines

Im Vordergrund des Datenexports über die FBS-RailML-Schnittstelle stehen Zugdaten. Typische Anwendungsfälle sind z. B. Fahrgastinformation, Dienstplanung, Abrechnung, Fahrzeugverfolgung. Infrastrukturdaten können optional ergänzend mitgegeben werden.

### 1.1:

Welche railML-Version wird unterstützt? (bitte nur eine – ggf. die bevorzugte – angeben)

railML 2.0 (rev 270 10/2009 ohne iRFP-Anpassungen)

railML 2.0 (rev 270a 10/2009 mit iRFP-Anpassungen, Schemen s. <http://schema.fbsbahn.de/2.0.5/fbs-railML.xsd>)

railML 2.1 (rev 409 06/2011)

railML 2.2 (rev 602 06/2013)

### 1.2:

Gibt es Vorgaben bezüglich Dateiname und/oder Pfad der Ziel-railML-Datei?

keine Vorgaben, Anwender ist frei in der Wahl von Dateinamen und Pfad

fest voreingestellter (Netzwerk-)Pfad, freier Dateiname

fest voreingestellter (Netzwerk-)Pfad, fester Dateiname, der Datum und Uhrzeit des Exports als Signatur enthalten soll:

\_\_\_\_\_

fest voreingestellter Pfad und Dateiname (ohne individuelle Exportsignatur): \_\_\_\_\_

andere Lösung: \_\_\_\_\_

### 1.3:

Sollen komprimierte railML-Dateien erzeugt werden?

unkomprimiert (Dateiendung \*.railml ab V2.2, \*.xml bis V2.1)

komprimiert lt. [www.wiki.railml.org/index.php?title=CO:fileConventions](http://www.wiki.railml.org/index.php?title=CO:fileConventions) (Dateiendung \*.railmlx)

## Infrastruktur

### 2.1:

An Infrastrukturdaten ist eine Mindestaufzählung an Betriebsstellen (<ocp>s), die von den Zügen passiert werden, zwingend notwendig. Als <ocp> im Sinne von railML können optional z. B. auch Blocksignale oder reine Fahrzeitmesspunkte gelten. Welche <ocp>s soll FBS exportieren?

alle in FBS abgebildeten Betriebsstellen  
alles außer reine Zugfolgestellen (Blocksignale usw.) und reine Fahrzeitmesspunkte (ohne funktionalen Hintergrund)

### 2.2:

Wie ordnet Ihre Software die Zugläufe zu eventuellen „Stammdaten“ (z. B. in Ihrer Software vorkonfigurierte Infrastruktur) zu?

#### **2.2.1 bezüglich der <ocp>s:**

Es werden Betriebsstellen anhand ihres RL100-Kürzels („bahnamtliche Abkürzung“) zugeordnet.  
Es werden Betriebsstellen anhand ihrer IBNR (internationale Bahnnummer) zugeordnet.  
sonstige Lösung: \_\_\_\_\_

\_\_\_\_\_

Die Zuordnung erfolgt nur an <ocp>s, an denen Verkehrshalte stattfinden.  
Die Zuordnung erfolgt nur an <ocp>s, an denen Verkehrs- oder Betriebshalte stattfinden.  
Die Zuordnung erfolgt an allen <ocp>s, auch bei Durchfahrten

\_\_\_\_\_

FBS soll prüfen, dass für alle <ocp>s mit Verkehrshalten eine IBNR exportiert wird.  
FBS soll prüfen, dass für alle <ocp>s mit Verkehrs- oder Betriebshalten eine IBNR exportiert wird.  
FBS soll prüfen, dass für alle <ocp>s eine IBNR exportiert wird.  
FBS braucht keine Prüfung bezüglich der IBNR durchführen (IBNR werden nicht verwendet)

- FBS soll prüfen, dass für alle <ocp>s mit Verkehrshalten Koordinaten exportiert werden.
- FBS soll prüfen, dass für alle <ocp>s mit Verkehrs- oder Betriebshalten Koordinaten exportiert werden.
- FBS soll prüfen, dass für alle <ocp>s Koordinaten exportiert werden.
- FBS muss keine Prüfung bezüglich der Koordinaten durchführen (Koordinaten werden nicht verwendet)

*Hinweis: FBS kann naturgemäß nicht sicherstellen, dass Koordinaten oder IBNR korrekt zutreffen, sondern nur das pure Vorhandensein prüfen. Eingabe- bzw. Konfigurationsfehler sind nicht auszuschließen. Koordinaten deutscher Betriebsstellen liegen i. d. R. in WGS84 vor; Abweichungen können mit dem Anwender vereinbart und unterstützt werden.*

### **2.2.2 bezüglich des Laufwegs (Strecken, Netzkanten) der Züge (in railML: <track>s und <line>s):**

- Laufwege werden nicht explizit ausgewertet – nur die Abfolge der o. g. Betriebsstellen.
- Laufwege werden über Streckennummern (<line @code>) ausgewertet.

### **2.2.3 bezüglich von Bahnhofsgleisen / Bahnsteigen**

- Bahnhofsgleise werden nicht ausgewertet.
- Bahnhofsgleise werden über <ocpTT @trackInfo> ausgewertet. Es werden:
  - betriebliche Gleis- bzw. Fahrwegbezeichnungen (des Infrastrukturbetreibers) erwartet.*
  - verkehrliche Gleisbezeichnungen (Bahnsteigbezeichnungen für den Reisenden) erwartet.*

## **2.3:**

Welche Infrastrukturdaten sollen zusätzlich exportiert werden?

- Es sollen Strecken und Streckengleise exportiert werden. Zusätzlich sollen:
  - zulässige Geschwindigkeiten exportiert werden.*
  - Neigungen, Bögen, und Tunnel exportiert werden.*

## Zugdaten

### 3.1:

railML unterstützt zwei Arten von Sichtweisen auf Züge: Verkehrliche Züge (<train @type='commercial'>) und betriebliche Züge (<train @type='operational'>). Welche Arten werden ausgewertet?

nur betriebliche Züge (<train @type='operational'>)  
nur verkehrliche Züge (<train @type='commercial'>) – *FBS wird systembedingt beide Arten exportieren*  
betriebliche und verkehrliche Züge (=Standard)

### 3.2:

*Sofern oben verkehrliche Züge (<train @type='commercial'>) – ggf. auch – angekreuzt wurden:* Es bestehen zwei Möglichkeiten der Bildung von verkehrlichen Zügen in railML:

Als verkehrliche Züge sollen nur Direktverbindungen ausgegeben werden (z. B. wie Tabellenfahrplan, Kursbuch). Verstärker sollen den Direktverbindungen untergeordnet werden. (**Standardfall** z. B. für Fahrgastinformation)  
Alle Mehrfachtraktionen sollen als getrennte verkehrliche Züge ausgegeben werden. (Spezialfall z. B. für Planungssysteme)

### 3.3:

Wie werden die Züge datentechnisch identifiziert?

#### **3.3.1 betriebliche Züge (<train @type='operational'>) – sofern oben angekreuzt:**

Jeder Zug aus railML wird separat eingelesen und nicht explizit identifiziert.  
Das railML-Attribut <train @trainNumber> allein wird ausgewertet. Es darf hier keine Werte-Dopplungen geben, auch nicht mit unterschiedlichen Verkehrstagen.  
Das railML-Attribut <train @trainNumber> wird zusammen mit den Verkehrstagen ausgewertet. Jede @trainNumber darf an jedem Verkehrstag nur einmal vorkommen.  
Die railML-Attribute <train @trainNumber, @additionalTrainNumber, @scope> werden ausgewertet. Jede Kombination der drei darf nur einmal vorkommen.  
andere Art der Identifikation: \_\_\_\_\_

*Hinweise: Der Standard ist hier landesspezifisch. Die Daten werden in FBS i. d. R. vom Infrastrukturbetreiber durchgereicht. In Deutschland ist Option 4 der Standard.*

### 3.3.2 verkehrliche Züge (<train @type='commercial') – *sofern oben angekreuzt:*

Jeder Zug aus railML wird separat eingelesen und nicht explizit identifiziert.

Das railML-Attribut <train @trainNumber> allein wird ausgewertet. Es darf hier keine Werte-Dopplungen geben, auch nicht mit unterschiedlichen Verkehrstagen.

Das railML-Attribut <train @trainNumber> wird zusammen mit den Verkehrstagen ausgewertet. Jede @trainNumber darf an jedem Verkehrstag nur einmal vorkommen. (=Standard)

Das railML-Attribut <train @name> allein wird ausgewertet. Es darf hier keine Werte-Dopplungen geben, auch nicht mit unterschiedlichen Verkehrstagen.

Das railML-Attribut <train @name> wird zusammen mit den Verkehrstagen ausgewertet. Jeder @name darf an jedem Verkehrstag nur einmal vorkommen.

andere Art der Identifikation: \_\_\_\_\_

*Hinweise: Im Standardfall enthält das Feld <train @trainNumber> eine i. d. R. alphanumerische, betriebsinterne Identifikation des verkehrlichen Zuges, die eindeutig je Verkehrstag ist (entspricht Option 3). Das Feld <train @name> enthält i. d. R. eine numerische, externe Identifikation des verkehrlichen Zuges zur Vermarktung, z. B. für Platzreservierungssysteme, die ebenfalls eindeutig je Verkehrstag ist (entspricht Option 5). Jede andere Option erfordert spezielle Vereinbarungen mit dem Anwender.*

### **3.4:**

Sofern das Attribut **<train @trainNumber>** von betrieblichen Züge (<train @type='operational'>) ausgewertet wird:

Die Werte dieses Attributs müssen rein numerisch sein.

Werte dieses Attributs dürfen alphanumerisch sein.

sonstige zu prüfende Restriktionen (z. B. Längenbegrenzung): \_\_\_\_\_

### **3.5:**

Sofern das Attribut **<train @trainNumber>** von verkehrlichen Züge (<train @type='commercial'>) ausgewertet wird:

Die Werte dieses Attributs müssen rein numerisch sein.

Die Werte dieses Attributs dürfen alphanumerisch sein.

sonstige zu prüfende Restriktionen (z. B. Längenbegrenzung): \_\_\_\_\_

**3.6:**

Sofern das Attribut **<train @name>** von verkehrlichen Zügen (<train @type='commercial'>) ausgewertet wird:

Die Werte dieses Attributs müssen rein numerisch sein.

Die Werte dieses Attributs dürfen alphanumerisch sein.

sonstige zu prüfende Restriktionen (z. B. Längenbegrenzung): \_\_\_\_\_

**3.7:**

Einige Infrastrukturbetreiber wie z. B. die DB Netz AG unterstützen das Modell von *Nebenläufen* („Ergänzungsfahrplänen“). Nebenläufe stellen abweichende Trassen (Zeiten und/oder Laufwege) an einer Teilmenge der Verkehrstage eines Zuges dar. Wie soll die FBS-RailML-Schnittstelle beim Vorkommen von Nebenläufen vorgehen?

Nebenläufe sollen unverändert ausgegeben werden (=Standard). Eine unter 3.3 genannte Eindeutigkeit je Verkehrstag gilt damit zwangsläufig nicht für Nebenläufe.

Nebenläufe werden vom Zielsystem nicht unterstützt. Sofern Sie im konkreten FBS-Datenbestand vorkommen, soll FBS den Export verweigern.

Nebenläufe werden vom Zielsystem nicht unterstützt. Sofern Sie im konkreten FBS-Datenbestand vorkommen, soll FBS sie in „normale Züge“ (Hauptläufe) mit disjunkten Verkehrstagen auflösen. Diese Hauptläufe haben dann zwangsläufig gleiche Identifikationen (@trainNumber, @name), jedoch unterschiedliche Verkehrstage.

**3.8:**

Im Normalfall würde die FBS-RailML-Schnittstelle den Export aller Zugteile des FBS-Datenbestands erlauben, wobei der Anwender individuelle Einschränkungen (Filterungen) vornehmen kann. Für einige Anwendungsfälle ist jedoch die Begrenzung z. B. auf „Vollzugteile“ (keine Leer-, Dienst- oder abgesperrten Züge oder Zugteile) sinnvoll oder notwendig. Soll die FBS-RailML-Schnittstelle eine solche Filterung implizit vornehmen?

Nein, es sollen alle Zugteile exportierbar sein.

Ja, bitte nur Zugteile exportieren, deren railML-Attribut <category @trainUsage> den Wert ='passenger' hat.

Ja, bitte nur Zugteile exportieren, deren railML-Attribut <trainPart>.<formationTT>.<passengerUsage>.<places @count> eine Sitzplatzanzahl >0 ausweist.

Ja, bitte nur Zugteile exportieren, deren Produkt oder Zuggattung (railML-Attribut <category @code>) einen Wert aus der Aufzählung \_\_\_\_\_ hat.

Ja, bitte nur Zugteile exportieren, deren Produkt oder Zuggattung (railML-Attribut <category @code>) *keinen* Wert aus der Aufzählung \_\_\_\_\_ hat.

Ja, bitte nur Zugteile exportieren, für die gleichzeitig auch ein Umlaufplan exportiert wird (die in mindestens einem exportierten Umlaufplan vorkommen).

Sonstige Filterung gewünscht, und zwar nach: \_\_\_\_\_

### 3.9:

Im Normalfall würde die FBS-RailML-Schnittstelle den Export über alle Verkehrstage der Fahrplanperiode des FBS-Datenbestands erlauben, wobei der Anwender individuelle Einschränkungen (Reduzierung des Export-Zeitraums) vornehmen kann. Soll die FBS-RailML-Schnittstelle eine implizite Filterung vornehmen? (*Hinweis: Dies würde zur Begrenzung des railML-Attributs <operatingPeriod @bitMask> auf die entsprechende Anzahl Tage führen und ist u. U. semantisch nicht kombinierbar mit Umlaufplänen.*)

Es braucht keine Fahrplanperiode definiert zu sein (<timetablePeriod> und <operatingPeriod @bitMask> fehlen).

Es muss eine Fahrplanperiode definiert sein:

*Eine Einschränkung ist nicht zulässig; es muss immer die ganze Fahrplanperiode exportiert werden.*

*Es sollen alle Verkehrstage exportierbar und beliebige Einschränkungen erlaubt sein.*

*Es müssen immer genau \_\_\_\_\_ zusammenhängende Tage exportiert werden. Diese müssen immer mit einem \_\_\_\_\_ beginnen.*

*Es dürfen nur 1-Tages-Exporte erfolgen, d. h. <timetablePeriod> @startDate und @endDate müssen identisch sein und alle <operatingPeriod @bitMask> müssen genau einen Tag lang sein. Der Tag kann vom Anwender beliebig aus der Fahrplanperiode gewählt werden.*

### 3.10:

In welcher Genauigkeit werden die Ankunfts-/Abfahrtszeiten unter <ocpTT>.<times @arrival, @departure> erwartet?

Zehntelminuten (Standard für klassische Fahrpläne; zur XML-Kompatibilität wird auf 6-Sekunden-Schritte gerundet)

volle FBS-interne Genauigkeit (nur für Spezialfälle; zur XML-Kompatibilität werden Sekundenbruchteile erzeugt)

andere Rundungsregel: \_\_\_\_\_

### 3.11:

Jeder Zugteil kann eine optionale Liniennummer oder -bezeichnung haben (railML-Attribut <trainPart @line>). Diese ist oftmals für Zwecke der Fahrgastinformation erforderlich. Alle Abweichungen vom Standard erfordern spezielle Vereinbarungen mit dem Anwender.

Die Linienbezeichnung kann beliebig, auch leer sein (=Standard).

Die Linienbezeichnung muss vorhanden (darf nicht leer) sein.

Die Linienbezeichnung muss rein numerisch sein (=Liniennummer).

Die Linienbezeichnung darf innerhalb eines durchgehenden verkehrlichen Zuges nicht wechseln.

Weitere Restriktionen / erwartete Inhalte: \_\_\_\_\_

**3.12:**

Jeder Zugteil kann optional auf eine Produkt- oder Zuggattung verweisen (railML-Attribut <trainPart @categoryRef>). Diese ist oftmals für Zwecke der Fahrgastinformation erforderlich. Alle Abweichungen vom Standard erfordern spezielle Vereinbarungen mit dem Anwender.

Die Produkt- oder Zuggattung ist optional (=Standard).

Die Produkt- oder Zuggattung muss vorhanden sein.

Weitere Restriktionen / erwartete Inhalte: \_\_\_\_\_

**3.13:**

Die FBS-RailML-Schnittstelle bietet optional die Möglichkeit, Züge mit den berechneten Fahrzeitdifferenzen und eingegebenen Fahrzeitzuschlägen zu exportieren (unter <sectionTT>.<runTimes>, immer ungerundet). Allerdings erfordert dies, dass alle Infrastruktur-Fahrzeitmesspunkte als <ocp>s ausgegeben werden und führt daher zu tendentiell großen railML-Dateien.

bitte mit <sectionTT>.<runTimes> exportieren

bitte ohne <sectionTT>.<runTimes> exportieren

**3.14:**

*Sofern oben betriebliche Züge (<train @type='operational'>) angekreuzt wurden:*

Die FBS-RailML-Schnittstelle bietet optional die Möglichkeit, Bremseneinstellungen der betrieblichen Züge unter <trainPartSequence>.<brakeUsage> auszugeben. Allerdings ist dann zwangsläufig ein Aufteilen eines an sich durchgehenden Zuglaufs in mehrere <trainPart>s notwendig, wenn sich Bremsvorgaben ändern. Der Verzicht auf diese Option vermeidet, dass Zugläufe allein wegen wechselnder Bremseneinstellungen unterbrochen werden.

bitte mit <trainPartSequence>.<brakeUsage> exportieren

bitte ohne <trainPartSequence>.<brakeUsage> exportieren

**3.15:**

*Sofern oben betriebliche Züge (<train @type='operational'>) angekreuzt wurden:*

Die FBS-RailML-Schnittstelle bietet optional die Möglichkeit, Zugbeeinflussung der betrieblichen Züge unter <trainPartSequence>/<formationTT>.<equipmentUsage> auszugeben. Allerdings ist dann zwangsläufig ein Aufteilen eines an sich durchgehenden Zuglaufs in mehrere <trainPart>s notwendig, wenn sich die Zugbeeinflussung ändert. Der Verzicht auf diese Option vermeidet, dass Zugläufe allein wegen wechselnder Zugbeeinflussung unterbrochen werden.

bitte mit <trainPartSequence>/<formationTT>.<equipmentUsage> exportieren

bitte ohne <trainPartSequence>/<formationTT>.<equipmentUsage> exportieren

**3.16:**

Werden railML-Umlaufpläne verarbeitet bzw. erwartet?

Nein, Umlaufpläne sind nicht zutreffend und sollen nicht exportierbar sein.

Ja, Umlaufpläne soll der Anwender optional zusätzlich exportieren können.

Ja, Umlaufpläne sind zwingend erforderlich. Es können nur Zugteile verarbeitet werden, für die ein Umlaufplan existiert. Es muss damit mindestens ein Umlaufplan exportiert werden (entspricht vorletzter Option unter 3.8).

**3.17:**

Für Nicht-Standard-Informationen unterstützt die FBS-RailML-Schnittstelle den Export „benutzerdefinierter Felder“. Benutzerdefinierte Felder können beispielsweise Auftraggeber, Auftragsnummern oder andere Querverweise enthalten. Sie erfordern in jedem Falle eine Abstimmung mit dem Anwender. In railML werden folgende Felder unterstützt: @remarks, @debitcode, @operator, beliebige Untermenge des Elements <organisationalUnits> oder beliebiges @any-Attribut mit Namensraum-Erweiterung, jeweils bei den Elementen <train>, <trainPart> oder <trainPartSequence>.

folgendes benutzerdefinierte Feld soll exportiert werden: \_\_\_\_\_

der Inhalt muss numerisch / alphanumerisch sein, ggf. Längen- oder sonstige Begrenzung: \_\_\_\_\_

das benutzerdefinierte Feld ist optional / zwingend erforderlich

benutzerdefinierten Felder werden nicht benötigt und sollen nicht exportiert werden.

**Weitere Bemerkungen:**

**Ansprechpartners in ihrem Unternehmen für Rückfragen:**

Firma: \_\_\_\_\_

Name: \_\_\_\_\_

Tel.-Nr.: \_\_\_\_\_

E-Mail: \_\_\_\_\_

Unser Unternehmen und unser Produkt dürfen durch iRFP z. B. auf dessen Internet-Seiten und Referenzlisten bezüglich der Austauschbarkeit von railML-Daten genannt werden.

Habe Sie noch Fragen, dann wenden Sie sich bitte an uns!

Ihre Ansprechpartner: Maurice Hindenburg *maurice.hindenburg{at}irfp.de*, Dirk Bräuer *dirk.braeuer{at}irfp.de*.

iRFP e. K.  
Institut für Regional- und Fernverkehrsplanung  
Hochschulstraße 45  
01069 Dresden  
Telefon: +49 351 4706819