

Beschreibung der FBS-RailML®-Schnittstelle

Stand: 20.11.2019

RailML-Versionen: 2.0 / 2.1 / 2.2

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

Inhalt

Allgemeines.....	2
Kopfinformationen (Versionsnummern, Dublin Core Metadata Element Set).....	3
Übersicht über die unterstützten Versionen.....	5
Hinweise zu Verwendbarkeit und Einschränkungen der RailML-Versionen.....	6
Infrastruktur (<i>infrastructure</i>).....	7
Strecken (<i>trackGroups</i>).....	7
Streckengleise (<i>tracks</i>).....	8
Kilometrierungsbereiche (<i>mileageChanges</i>).....	9
Querverweise auf Betriebsstellen (<i>trackBegin</i> , <i>trackEnd</i> und <i>crossSection</i>).....	10
Geschwindigkeitswechsel (<i>speedChanges</i>).....	10
Neigungswechsel (<i>gradientChanges</i>).....	11
Krümmungswechsel (<i>radiusChanges</i>).....	11
Tunnel (<i>tunnel</i>).....	11
Wechsel des Infrastrukturbetreibers (<i>ownerChanges</i>).....	12
Beginn, Ende und Wechsel der Elektrifizierung (<i>electrificationChanges</i>).....	12
Streckenklassen (<i>axleWeightChanges</i>).....	12
Spurweite (<i>gaugeChanges</i>).....	13
Geschwindigkeitsprofile (<i>speedProfiles</i>).....	13
Betriebsstellen (<i>operationControlPoints</i>).....	14
Verwaltungsdaten (<i>organizationalUnits</i>).....	17
Fahrzeug- und Zugbildungsdaten (<i>rollingstock</i>).....	18
Fahrzeugdaten (<i>vehicles</i>).....	18
Zugbildungsdaten (<i>formations</i>).....	19
Fahrplandaten (<i>timetable</i>).....	19
Fahrplanperioden (<i>timetablePeriods</i>).....	19
Verkehrstageregelungen / Saisonierungen (<i>operatingPeriods</i>).....	19
Zuggattungen und Produktbezeichnungen (<i>categories</i>).....	21
Züge und Zugteile (<i>trains</i> und <i>trainParts</i>).....	22
Zugteile (<i>trainParts</i>).....	22
Laufweg von Zugteilen (<i>ocpsTT</i>).....	24
Betriebliche und verkehrliche Züge (<i>trains</i>).....	26
Betriebliche Züge (<i>operational trains</i>).....	27
Verkehrliche Züge (<i>commercial trains</i>).....	28
Umlaufpläne (<i>rosterings</i>).....	29
Umlauf-Kopfangaben (<i>rostering</i>).....	30
Fahrten und Dienste (<i>blockParts</i>).....	30
Fahrtengruppen (<i>blocks</i>).....	31
Umläufe (<i>circulations</i>).....	32
Änderungsübersicht.....	34

Allgemeines

Die FBS-RailML-Schnittstelle erlaubt einen Export der meisten in FBS enthaltenen Anwendungsdaten (Infrastruktur und Fahrplan/Zugdaten einschl. Kalenderdaten usw.) in strukturierter, systematischer Form zur technischen Auswertung in anderen Programmen. Erzeugt werden 8-Bit-Textdateien mit XML-Struktur. Für die nachfolgend beschriebene Version 2.0.5 werden XML-Schemen auf Basis des RailML[®]-Standards 2.0 verwendet, die für FBS geringfügig angepasst wurden. Diese Version der FBS-RailML-Schemendateien finden Sie unter www.irfp.de/deutsch/fbs/schnittstelle_railml.html
<http://schema.fbsbahn.de/2.0.5/fbs-railML.xsd>

Die übrigen Versionen (2.0.0, 2.1.0, 2.2.0) verwenden die unveränderten RailML-Schemen, jedoch können zusätzliche, optionale Erweiterungen enthalten sein, deren Schemen unter http://schema.fbsbahn.de/2.x/fbs_railml_extension.xsd verfügbar sind.

Die Abweichungen vom bzw. Ergänzungen zum RailML-Standard sind in der folgenden Dokumentation gekennzeichnet. Zusätzlich wird zu jedem Attribut angegeben, ob es in der konkreten FBS-RailML-Schnittstelle obligatorisch oder optional ist. Viele der in RailML als optional deklarierten Attribute sind in der FBS-RailML-Schnittstelle obligatorisch.

Die FBS-RailML-Dateien sind grundsätzlich UTF-8-codiert. Eine RailML-Datei enthält auch Beschreibungen (Textdarstellung z. B. von Verkehrstagen) in der Sprache, die beim Exportieren als FBS-Programmsprache eingestellt war und ist daher nicht in jeder Hinsicht sprachneutral.

Auf Grund der Abhängigkeiten, die sich durch die im RailML-Standard festgelegten Referenzen (Id-Ref-Beziehungen) ergeben, ist der Exportumfang vom Anwender nur relativ eingeschränkt beeinflussbar. So ist es z. B. nicht möglich, Züge ohne Infrastruktur zu exportieren, da der RailML-Standard erfordert, dass die verwendeten Betriebsstellen auch in der RailML-Datei definiert sein müssen. Der Anwender kann beim Export u. a. wählen:

- Vollständigkeit der Infrastruktur (nur Mindestumfang - Betriebsstellen - oder auch Strecken, Gleise und ggf. Geschwindigkeiten, Höhen usw.),
- alle Züge oder nur einen eingeschränkten Teil der Züge (Filtermöglichkeit nach Zuggattung, Produkt oder Linienbezeichnung),
- Zugteile mit oder ohne Fahrzeiten, Entfernungen und Verweise auf Streckengleise der Infrastruktur (d. h. mit oder ohne *sectionTT*-Struktur),
- mit oder ohne *commercialTrains*-Struktur,
- Umfang der zu exportierenden Umlaufpläne.

Eine Bedienungsanleitung der Schnittstelle finden Sie im Dokument *FBS-RailML2 – Hinweise zur Bedienung*, welches ebenfalls unter www.irfp.de/deutsch/fbs/schnittstelle_railml.html zur Verfügung steht.

Hinweis für Entwickler von RailML-Import-Schnittstellen, die (auch) RailML-Dateien aus FBS einlesen können sollen: Üblicher Weise bestehen je nach Anwendungsfall individuelle Anforderungen an Inhalte von RailML-Dateien. Diese können i. d. R. beim Export vom Anwender eingestellt werden. Wegen der vielfältigen Optionen besteht die Möglichkeit, für bestimmte Anwendungsfälle (Programm-Paarungen wie z. B. „Export aus FBS zur Fahrgastinformation mit ...“) konkrete Voreinstellungen fest zu hinterlegen, so dass inhaltliche Abhängigkeiten festgesetzt sind oder durch FBS bereits beim Export geprüft werden. Wenn Sie eine Import-Schnittstelle entwickeln, bitten wir Sie, mit uns Kontakt aufzunehmen zur Abstimmung der notwendigen Inhalte und eventuellen Voreinstellung der Optionen in FBS.

Die Implementierung der Schnittstelle ist zu keinem Zeitpunkt als abgeschlossen zu betrachten, d. h. es können jederzeit Erweiterungen erfolgen. Entsprechend dem Grundprinzip von XML sollte ein lesendes Programm daher grundsätzlich unbekannte Strukturen und Attribute überspringen. Gleichfalls können Sie uns als FBS-Anwender oder Schnittstellenpartner jederzeit kontaktieren, sofern Sie in der Schnittstelle zusätzliche (noch nicht verfügbare) Informationen vermissen, die in FBS enthalten sind.

In der folgenden Beschreibung sind alle Strukturen und Attribute aufgeführt, die zum Zeitpunkt der Erstellung dieses Dokuments Bestandteil der FBS-RailML-Schnittstelle sind – mit Ausnahme des Attributs **id**. Das Attribut **id** ist in den meisten Strukturen obligatorisch, hat jedoch nur innerhalb der RailML-Datei eine Bedeutung als Primärschlüssel und Verweisziel. Die FBS-RailML-Schnittstelle sorgt dafür, dass die Id-Werte eindeutig und integer sind; der eigentliche Inhalt dieser Werte ist jedoch nicht von Bedeutung. Ein lesendes Programm darf in den Id-Attributen keinen speziellen (konkreten) Inhalt erwarten.

Kopfinformationen (Versionsnummern, Dublin Core Metadata Element Set)

Der Anwender hat beim Export die Wahl einer RailML-Schemenversion. Dadurch werden neben den eigentlichen inhaltlichen Änderungen auch die Versionsnummern bestimmt:

Das Attribut **<railml>.version** ist derzeit je nach Schemenversion festgelegt auf **2.0**, **2.1** oder **2.2**. Bei Vorab-Veröffentlichungen oder Tests zukünftiger Versionen kann dort auch eine Release-Nummer der RailML-Schemen angegeben sein, z. B. **2.2r611** (für RailML 2.3).

Die implementierungsabhängigen Versionsnummern und andere Kopfinformationen sind entsprechend RailML-Standard im **Dublin Core Metadata Element Set** (Namensraum „dc:“, Struktur **<railml>.<metadata>**) enthalten. Die Interpretation der einzelnen dc-Elemente ist derzeit nicht weiter standardisiert und wird daher für die FBS-RailML-Schnittstelle wie folgt festgelegt:

Das Attribut **<metadata>.format** enthält die interne Versionsnummer der Schemen-Ausprägung (auch *RailML-Profil* genannt). Diese Versionsnummer ändert sich dann, wenn sich die Interpretation oder die Vollständigkeit der Umsetzung des RailML-Schemas durch FBS ändert. Ein lesendes Programm sollte prüfen, dass diese Versionsnummer nicht niedriger ist als die Version, mit der das Programm frühestens getestet wurde. Insbesondere kann hiermit vom lesenden Programm einfach geprüft werden, ob bestimmte notwendige Daten vorhanden sein werden, die in RailML als optional gekennzeichnet sind und in die FBS-Implementation erst nachträglich (ab einer bestimmten Versionsnummer) ergänzt wurden.

Das Attribut **<metadata>.identifizier** enthält eine Kompatibilitätsnummer als einfachen Integer-Wert. Diese Nummer wird nur dann geändert, wenn ein bestehender Datenwert (Attribut oder Element) nachträglich uminterpretiert wird (z. B. in Folge einer Fehlerkorrektur). Ein lesendes Programm sollte dieses Attribut auf exakt den erwarteten Wert prüfen - sonst läuft es Gefahr, dass die auszuwertenden Datenfelder nicht mehr den erwarteten Inhalt enthalten. Beispielsweise könnte ein Geschwindigkeitsfeld bis zu einem bestimmten Zeitpunkt die Geschwindigkeit in km/h enthalten. Es sei angenommen, dass sich das nachträglich als nicht RailML-konform herausstellte. Um die RailML-Konformität wiederherzustellen, müsste das Geschwindigkeitsfeld nachträglich – ohne umbenannt zu werden – auf die Einheit m/s umgestellt werden. In diesem Falle würde **<metadata>.identifizier** um eins weitergezählt werden, um nicht aktualisierte Programme davon abzuhalten, den neuen Wert als km/h einzulesen.

Das Attribut **<metadata>.identifizier** wird nicht aktualisiert, wenn neue Daten hinzukommen, was es von **<metadata>.format** unterscheidet. Es wird erwartet, dass sich **<metadata>.identifizier** im Gegensatz zu **<metadata>.format** nur sehr selten ändert.

Das Attribut **<metadata>.language** enthält

- bis Vers. 2.1: Nummer und Namen des Zeichensatzes,
- ab Vers. 2.2: das BCP47-crypt-Tag des Zeichensatzes (mit prilang=*und*-)

in dem die FBS-Daten im FBS vorliegen. Dieser Wert kann z. B. von Bedeutung sein, wenn die in der RailML-Datei enthaltenen UTF-8-Zeichenketten (Bahnhofsnamen usw.) vom lesenden Programm in eine Nicht-Unicode-Zeichenkette umgewandelt werden müssen.

Das Attribut **<metadata>.source** enthält Dateinamen der exportierten FBS-Dateien (je nach Position des übergeordneten Elements **<metadata>** die FBS-Netz-Datei oder FBS-Fahrzeugdatendateien oder die FBS-Betriebsstellendatei). Es werden nur der Dateiname, relative Pfade und Pfadplatzhalter angegeben, nicht jedoch ein eventueller absoluter Pfad. Die Information kann zur zusätzlichen Orientierung für den Anwender beim Import einer FBS-RailML-Datei angezeigt werden.

Ein weiteres Attribut **<metadata>.source** dient zur Identifizierung des schreibenden Programms und enthält eine Zeichenkette mit Versionsnummern der verwendeten FBS-Module (ausführbare Datei und Schnittstellen-Bibliothek). Diese Informationen sind für lesende Programme wenig aussagekräftig, können aber zur Einordnung von **<metadata>.format** und **<metadata>.identifizier** dienen.

Das Attribut **<metadata>.date** enthält Datum und Uhrzeit des Exports (des Erzeugens der RailML-Datei) im xs:dateTime-Format.

Das Attribut **<metadata>.creator** enthält den FBS-Lizenznamen der FBS-Lizenz, die beim Erzeugen der RailML-Datei verwendet wurde.

Übersicht über die unterstützten Versionen

Folgende Versionen können derzeit aus FBS exportiert werden:

Version <railml>. version	Profil <metadata>. format	Kompatibi- litätsnummer <metadata>. identifizier	Bemerkungen	Revision
2.0	2.0.0	4	nur sehr eingeschränkt verwendbar	270
2.0	2.0.5	1	iRFP-eigene Anpassungen	(270)
2.1	2.1.0	4	unveränderte Original-Schemen	409
2.2	2.2.1	4	unveränderte Original-Schemen	602

Versions-Verlauf der Kompatibilitätsnummer (<metadata>.identifizier):

<metadata>. identifizier	seit	fortgeschrieben wegen
2	23.05.2012	Wechsel der Einheit von <ocpTT>.<sectionTT>.distance von km auf m
3	06.05.2015	Orientierungswechsel auf dir='down' bei km-abwärtigen <speedChange>s
4	15.09.2015	Reihenfolgetausch von Längen- und Breitengrad in Koordinaten

Übersicht über die verwendeten Schemen und Namensräume:

Version / Profil	URI des Namensraums (xmlns) URL des Schemas (xsi:schemaLocation)	Bemerkungen
alle	http://www.w3.org/2001/XMLSchema-instance (xmlns:xsi) http://purl.org/dc/elements/1.1/ (xmlns:dc)	
	http://schema.fbsbahn.de/2.x/fbs_railml_extension (xmlns:fbs) http://schema.fbsbahn.de/2.x/fbs_railml_extension.xsd	iRFP-eigene Erweiterungen
2.0.0	http://www.railml.org/schemas/2009 http://www.railml.org/schemas/2009/railML-2.0/railML.xsd	
2.0.5	http://schema.fbsbahn.de/2.0.5 http://schema.fbsbahn.de/2.0.5/fbs-railML.xsd	iRFP-eigene Anpassungen
2.1.0	http://www.railml.org/schemas/2011 http://www.railml.org/schemas/2011/railML-2.1/railML.xsd	
2.2.1	http://www.railml.org/schemas/2013 http://www.railml.org/schemas/2013/railML-2.2/railML.xsd	

Darüber hinaus kann der Anwender beim Export eigene Schemen-Erweiterungen für benutzerdefinierte Felder aus FBS hinzufügen.

Der FBS-Namensraum (http://schema.fbsbahn.de/2.x/fbs_railml_extension) wird derzeit in folgenden Fällen verwendet:

- Wenn die Export-Einstellungen vom Anwender in der RailML-Datei gespeichert werden (*Einstellungen als Vorlage speichern – in der XML/RailML-Datei*; dies dient dem wiederholten Export mit denselben Einstellungen).
- Wenn „Tagesnummern“ in Umlaufplänen exportiert werden (Einstellung *innere Wochentagsgruppen exportieren*; ermöglicht einem lesenden Programm, 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).

Über diese Export-Einstellungen kann der Anwender indirekt steuern, ob der FBS-Namensraum eingebunden wird.

Hinweise zu Verwendbarkeit und Einschränkungen der RailML-Versionen

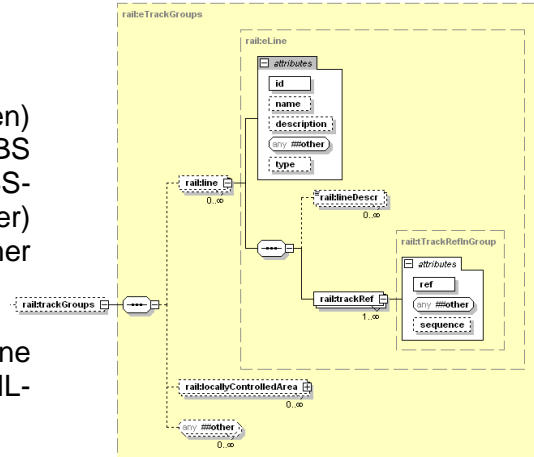
Version	Hinweise
2.0.0	<ul style="list-style-type: none"> - keine Geschwindigkeitsprofile - Fahrwege der Züge nicht eindeutig nachvollziehbar, daher: - kein Re-Import der Daten aus Version 2.0.0 nach FBS möglich - Die Last von Zügen (mit Wagen) wird nicht explizit exportiert (Attribut <formationTT>.load fehlt). Sie wird jedoch indirekt über <formationTT>.weight exportiert, wo die Gesamtmasse eines Zugteils steht. - Bedarfzüge können nicht als solche gekennzeichnet werden, da das Attribut <operatingDay>.onRequest fehlt. - In RailML 2.0 wurde noch nicht zwischen Schnell- und Betriebsbrems Hundertstel unterschieden. Es werden daher nur die Betriebsbrems Hundertstel exportiert. Da beide aber i. d. R. unterschiedlich sind, gehen hier bei einem Re-Import wichtige Informationen verloren. - Es können Zugteile, die trotz vorhandener Sitzplätze nicht zur Fahrgastbeförderung freigegeben sind (leerüberführte Wagen einschl. Triebwagen - DLt am Vollzug) nicht als solche gekennzeichnet werden, weil das Attribut <passengerUsage>.places zwingend >0 sein muss. (<passengerUsage> wird daher bei abgesperrten Wagen weggelassen.) - Die verkehrliche Linienbezeichnung von Zugteilen wird nicht ausgegeben. - keine Wagenreihungsnummern - keine Umlaufpläne
2.0.5	<ul style="list-style-type: none"> - von iRFP angepasste Schemendateien, um die Nachteile der Version 2.0.0 zu vermeiden
[2.0.5]	<p>alle Versionen außer V2.0.5: Es fehlt das erste <mileageChange>-Element am Anfang eines Streckengleises in der <mileageChanges>-Struktur. Es ist damit nicht möglich, die Kilometrierung einer Strecke im Zusammenhang zu erfassen. Die initiale absolute Position eines Gleises ist jedoch aus dem Element <trackBegin> ersichtlich.</p>
2.1.0	<ul style="list-style-type: none"> - keine Geschwindigkeitsprofile - <ocp>.number wird für IBNR verwendet entgegen der Deklaration als "deprecated", da ansonsten keine Betriebsstellennummern exportierbar wären - keine Fahrzeugindizes und Gruppennummern in Umlaufplänen
2.2.0	<ul style="list-style-type: none"> - keine der o. g. Einschränkungen außer initialer <mileageChange>
2.2.1	<ul style="list-style-type: none"> - wie 2.2.0, zusätzlich Unterstützung der hierarchischer Betriebsstellen über das Attribut <ocp>.parentOcpRef, um betrieblich unterschiedliche Betriebsstellen verkehrlich zusammenfassen zu können
alle, optional	<p>mit iRFP-eigenen Erweiterungen: Die Erweiterungen ermöglichen</p> <ul style="list-style-type: none"> - den Export benutzerdefinierter Felder aus FBS (z. B. mit Verwaltungsinformationen wie Besteller, Aufgabenträger, Vertragsnummer, durchführendes EVU) in flexiblerer Art als mit Standard-RailML-Feldern und auch an Stellen, an denen entsprechende Standard-RailML-Felder fehlen, - ein Wiedergeben „gefalteter“ Umlaufplan-Darstellungen, d. h. mehrere „innere“ Wochentage auf einem Blatt zusammengefasst.

Infrastruktur (*infrastructure*)

Die Infrastruktur enthält immer mindestens die Struktur *operationControlPoints* mit den Betriebsstellen, die von den Zügen und Zugteilen aus *timetable* referenziert werden. Der Anwender kann wählen, ob zusätzlich die Strukturen *tracks* und *trackGroups* mit Strecken und Streckengleisen sowie innerhalb der Streckengleise die Strukturen für Geschwindigkeiten (*speedChanges*), Neigungen (*gradientChanges*), Bögen (*radiusChanges*) und Tunnel (*tunnels*) exportiert werden sollen.

Strecken (*trackGroups*)

Die Struktur **trackGroups** enthält (sofern vorhanden) für jede Strecke aus FBS einen Eintrag. Sofern in FBS Streckennummern definiert sind, wird eine FBS-Strecke in Bereiche zusammenhängender (gleicher) Streckennummer zerlegt, da die Streckennummer nach außen hin im Allgemeinen einen Primärschlüssel darstellt. Sofern in FBS keine Streckennummern angegeben sind oder eine FBS-Strecke eine durchgehende Streckennummer hat, sind die RailML-Strecken identisch mit den FBS-Strecken.



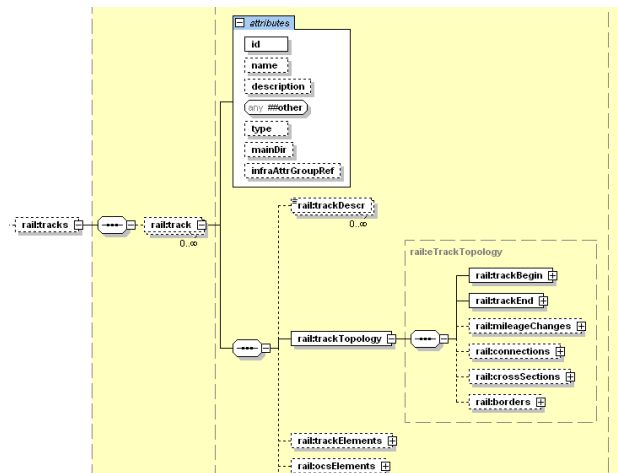
Attribut	Inhalt	optional	Bemerkungen
<line>.name	sofern in FBS definiert: Streckennummer der Strecke	x	nur V2.0.0
<line>.name	Abkürzung der ersten und letzten Betriebsstelle der Strecke		ab V2.0.5
<line>.code	sofern in FBS definiert: UIC-Nummer der Bahnverwaltung (EIU) und Streckennummer der Strecke, durch einen Punkt getrennt	x	nur V2.1
<line>.code	sofern in FBS definiert: Streckennummer der Strecke	x	ab V2.2
<line>.infrastructure-ManagerRef	sofern in FBS definiert: Referenz auf ein Element <infrastructureManager> unter <organizationalUnits>	x	ab V2.2
<line>.uicNumber	sofern in FBS definiert: UIC-Nummer der Bahnverwaltung (EIU) der Strecke	x	nur V2.0.5 ergänzend zu RailML
<line>.lineNumber	sofern in FBS definiert: Streckennummer der Strecke		nur V2.0.5 ergänzend zu RailML
<line>.<trackRef>.ref	Querverweis auf id eines Streckengleises in der tracks-Struktur		

Innerhalb der <line>-Struktur ist für jedes zur Strecke gehörende Streckengleis ein track.ref-Querverweis vorhanden. Jedes Streckengleis gehört zu genau einer Strecke. Jede Strecke enthält mindestens ein Streckengleis. Strecken, die abwechselnd ein- und zweigleisig sind, werden nach folgendem Prinzip in Streckengleise zerlegt:



Streckengleise (*tracks*)

Die Struktur **tracks** enthält (sofern vorhanden) Informationen zu jedem Streckengleis. Sofern **tracks** vorhanden ist, ist in jedem **track** auch je eine **trackTopology**-Struktur vorhanden, während **trackElements** optional ist und **ocsElements** derzeit nicht vorhanden ist.



Attribut	Inhalt	optional	Bemerkungen
track.name	Abkürzung der ersten und letzten Betriebsstelle des Streckengleises in Reihenfolge der Regelfahrrichtung		
track.type	derzeit immer <i>mainTrack</i> , nach Erweiterung auf nicht durchgehende Bahnhofshauptgleise zukünftig auch <i>stationTrack</i>		
trackTopology.trackBegin.pos	relative Kilometrierung der Betriebsstelle des Streckenanfangs in Metern		
trackTopology.trackBegin.absPos	absolute Kilometrierung der Betriebsstelle des Streckenanfangs in Metern		
trackTopology.trackBegin.macrosopicNode.ocpRef	Querverweis auf erste Betriebsstelle der Strecke in der Struktur <i>infrastructure.operationControlPoints</i>		
trackTopology.trackEnd.pos	relative Kilometrierung der Betriebsstelle des Streckenendes in Metern		
trackTopology.trackEnd.absPos	absolute Kilometrierung der Betriebsstelle des Streckenendes in Metern		
trackTopology.trackEnd.macrosopicNode.ocpRef	Querverweis auf letzte Betriebsstelle der Strecke in der Struktur <i>infrastructure.operationControlPoints</i>		
trackTopology.mileageChanges	s. u.	X	
trackTopology.crossSections	s. u.		
trackTopology.trackElements.speedChanges	s. u.	X	
trackTopology.trackElements.radiusChanges	s. u.	X	
trackTopology.trackElements.tunnels	s. u.	X	
trackTopology.trackElements.ownerChanges	s. u.	X	
trackTopology.trackElements.electrificationChanges	s. u.	X	

trackTopology. trackElements. axleWeightChanges	s. u.	x	
trackTopology. trackElements. gaugeChanges	s. u.	x	

Die relative Kilometrierung eines Gleises (Attribute *pos*) ist immer ununterbrochen fortlaufend steigend, jedoch nicht unbedingt bei 0 beginnend. (Dann nicht, wenn innerhalb einer Strecke das Gleis gewechselt wird, d. h. wenn das Gleis nicht am Anfang der Strecke beginnt.) Die absolute Kilometrierung eines Gleises ist willkürlich (i. d. R. historisch bedingt) und kann beliebige Unstetigkeitsstellen aufweisen („springen“) sowie entgegen der relativen Kilometrierung verlaufen („fallen“). Der Zusammenhang zwischen relativer und absoluter Kilometrierung erschließt sich durch die *mileageChanges*-Struktur. Die Angabe der absoluten Kilometrierung an den Betriebsstellen (in *crossSection*) ist redundant.

Kilometrierungsbereiche (*mileageChanges*)

Die Struktur **mileageChanges** definiert den Zusammenhang zwischen relativer (fortlaufender) und absoluter (außen sichtbarer) Kilometrierung jedes Streckengleises und damit jeder Strecke. In FBS sind beide Gleise einer zweigleisigen Strecke grundsätzlich identisch kilometriert (ansonsten wären die beiden Streckengleise als zwei eingleisige Strecken abgebildet). In dem Ausnahmefall, bei dem die absolute Kilometrierung auf der gesamten Strecke identisch ist mit der relativen Kilometrierung, kann die gesamte *mileageChanges*-Struktur in der RailML-Datei fehlen. Es sind dann dennoch die absoluten Positionen der Betriebsstellen (in *trackBegin*, *trackEnd* und *crossSection*) angegeben und immer identisch mit der jeweiligen relativen Position.

Nur Version 2.0.5: Sofern die *mileageChanges*-Struktur vorhanden ist, ist immer mindestens ein *mileageChange*-Element am Anfang des Streckengleises (d. h. mit gleicher relativer *pos* wie *trackBegin*) vorhanden. Dieses Element definiert daher eigentlich keinen Wechsel, sondern die initiale Kilometrierung des Gleises vor dem ersten Wechsel. *Diese Regel wurde bisher für die offiziellen RailML-Versionen nicht übernommen. In diesen Versionen fehlt daher das erste Element, und es ist nicht möglich, die Kilometrierung einer Strecke im Zusammenhang zu erfassen. Die initiale absolute Position eines Gleises ist jedoch aus dem Element <trackBegin> ersichtlich.*

Es werden nur „effektiv wirksame“ Kilometrierungswechsel aus FBS ausgegeben – für alle anderen hätte das obligatorische Attribut *type* in RailML keinen Wert.

Attribut	Inhalt	optional	Bemerkungen
absPosIn	absolute Kilometrierung des Kilometrierungswechsels in Richtung fallender relativer Kilometrierung	(x)	nicht beim initialen (ersten) Element in V2.0.5
type	= <i>missing</i> , wenn absPosIn < absPos gilt = <i>overlapping</i> , wenn absPosIn > absPos gilt	(x)	
absPos	absolute Kilometrierung des Kilometrierungswechsels in Richtung steigender relativer Kilometrierung		
pos	relative Kilometrierung des Kilometrierungswechsels		
dir	= <i>up</i> , wenn absolute Pos. in Richtung steigender relativer Kilometrierung (= in Definitionsrichtung des Gleises) steigen = <i>down</i> , wenn absolute Pos. in Richtung steigender relativer Kilometrierung (= in Definitionsrichtung des Gleises) fallen		nur bis V2.1
absDir	= <i>raising</i> , wenn absolute Pos. in Richtung steigender relativer Kilometrierung (= in Definitionsrichtung des Gleises) steigen = <i>falling</i> , wenn absolute Pos. in Richtung steigender relativer Kilometrierung (= in Definitionsrichtung des Gleises) fallen		erst ab V2.2

Alle Positionen sind in Metern angegeben.

Querverweise auf Betriebsstellen (*trackBegin*, *trackEnd* und *crossSection*)

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung der Betriebsstelle in Metern		
absPos	absolute Kilometrierung der Betriebsstelle in Metern		
ocpRef	Querverweis auf die Betriebsstelle in der Struktur <i>infrastructure.operationControlPoints</i>		

Als „Betriebsstelle“ können hier im weitesten Sinne (je nach Konfiguration in FBS und Einstellung beim Export) auch Blocksignale, Blockkennzeichen oder reine Fahrzeitmesspunkte (ohne funktionalen Hintergrund) aufgezählt sein.

Geschwindigkeitswechsel (*speedChanges*)

Die Struktur **<speedChanges>** enthält für jeden Geschwindigkeitswechsel eines Gleises je ein Element **<speedChange>**. Jeder **<speedChange>** bezieht sich mit seinem Attribut *ProfileRef* auf ein Geschwindigkeitsprofil aus der Struktur **<speedProfiles>**. Die Geschwindigkeitsprofile jedes Gleises sind disjunkt; d. h., die Geschwindigkeit eines **<speedChange>**s gilt bis zum nächsten **<speedChange>** **mit der gleichen *ProfileRef*** bzw. bis zum Streckenende, falls kein weiterer **<speedChange>** **mit der gleichen *ProfileRef*** enthalten ist. Je nach Gültigkeitsrichtung des Geschwindigkeitsprofils ist an der Position des Streckenanfangs (**<trackBegin>.pos**) bzw. Streckenendes (**<trackEnd>.pos**) immer ein Geschwindigkeitswechsel enthalten. Die Geschwindigkeitswechsel sind entsprechend der Gültigkeitsrichtung ihres Geschwindigkeitsprofils angegeben. Im Falle eines Geschwindigkeitsprofils, das für beide Fahrrichtungen gültig ist, sind die Informationen redundant doppelt vorhanden – je einmal pro Fahrrichtung.

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung in Metern		
absPos	absolute Kilometrierung in Metern		
description	Bezeichnung des Elements wie in FBS eingegeben	x	
dir	=‘up’ wenn gültig in Definitionsrichtung des Gleises =‘down’ wenn gültig entgegen Definitionsrichtung		
profileRef	Querverweis auf das Attribut id eines Elements <speedProfile> aus <speedProfiles> .		erst ab V2.2
status	Nummer des Geschwindigkeitsprofils in FBS (dient als Ersatz für das noch nicht vorhandene <i>profileRef</i>)		nur bis V2.1
vMax	Geschwindigkeit ab diesem <speedChange> in km/h; =999 bei Ende einer Geschwindigkeitsbegrenzung in einem Zusatzprofil (bis V2.2); =end bei Ende einer Geschwindigkeitsbegrenzung in einem Zusatzprofil (ab V2.3)		
trainRelation	= <i>headOfTrain</i> , wenn der Geschwindigkeitswechsel für die Zugspitze gilt auch bei erhöhender Geschwindigkeit	x	erst ab V2.2
mandatoryStop	= <i>True</i> wenn ein Zwangshalt an diesem Geschwindigkeitswechsel definiert ist	x	erst ab V2.2
signalised	= <i>False</i> wenn der Geschwindigkeitswechsel wegen fehlender (Vor-)Signalisierung besonders zu kennzeichnen ist	x	erst ab V2.2

Neigungswechsel (*gradientChanges*)

Die Struktur **<gradientChanges>** enthält für jeden (Längs-)Neigungswechsel eines Gleises je ein Element **<gradientChange>**. Die Neigungswechsel sind immer in Definitionsrichtung des Gleises angegeben unabhängig von einer Fahrtrichtung.

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung in Metern		
absPos	absolute Kilometrierung in Metern		
description	Bezeichnung des Elements wie in FBS eingegeben	x	
dir	immer =‘up‘		
slope	Neigung ab hier bis zum nächsten <gradientChange> bzw. bis zum Ende des Gleises in Promille; positive Werte = Steigung, negative Werte = Gefälle		
<geoCoord>.coord	=‘0 0 xxx‘ (bis V2.1) mit xxx = Höhe des Neigungswechsels =‘0 0‘ (ab V2.2); nur aus formalen Gründen enthalten		Höhenwert in Metern mit zwei Nachkommastellen
<geoCoord>.extraHeight	enthält die Höhe des Neigungswechsels	x	
<geoCoord>.heightEpsgCode	enthält den EPSG-Code des Referenzsystems der Höhenangabe	x	

Krümmungswechsel (*radiusChanges*)

Die Struktur **<radiusChanges>** enthält für jeden Krümmungswechsel eines Gleises je ein Element **<radiusChange>**. Die gesamte Struktur ist optional; Krümmungsangaben müssen in FBS nicht eingegeben sein. Die Krümmungswechsel sind immer in Definitionsrichtung des Gleises angegeben unabhängig von einer Fahrtrichtung.

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung in Metern		
absPos	absolute Kilometrierung in Metern		
description	Bezeichnung des Elements wie in FBS eingegeben	x	
dir	immer =‘up‘		
radius	Bogenradius ab hier bis zum nächsten <gradientChange> bzw. bis zum Ende des Gleises in Metern; positive Werte = Rechtsbögen, negative Werte = Linksbögen; =0 für gerade Abschnitte		
superelevation	Überhöhung im Bogenscheitel in mm	x	

Die übrigen Bogenkennwerte aus FBS (Überhöhungsrampen, Übergangsbögen) werden derzeit mangels Bedarfs nicht ausgegeben.

Tunnel (*tunnel*)

Die Struktur **<tunnels>** enthält für jeden Tunnel eines Gleises je ein Element **<tunnel>**. Die gesamte Struktur ist optional.

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung in Metern		
absPos	absolute Kilometrierung in Metern		
dir	immer =‘up‘		
name	Bezeichnung des Tunnels wie in FBS eingegeben	x	
length	Länge des Tunnels in m		
crossSection	Tunnelquerschnittsfläche in m ²	x	
kind	Ausmauerungsart i. S. des Tunnelwiderstands		

Wechsel des Infrastrukturbetreibers (*ownerChanges*)

Die Struktur **<ownerChanges>** enthält für jeden Wechsel des Infrastrukturbetreibers eines Gleises je ein Element **<ownerChange>**. Die gesamte Struktur ist optional.

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung in Metern		
absPos	absolute Kilometrierung in Metern		
description	Bezeichnung des Elements wie in FBS eingegeben	x	
dir	immer =‘up‘		
uic-no	UIC-Nummer der Bahnverwaltung	x	
ownerName	Kürzel des Infrastrukturbetreibers wie in FBS eingegeben		bis Vers. 2.1
infrastructureManagerRef	Querverweis auf ein Element <infrastructureManager> der Struktur <organizationalUnits>		ab Vers. 2.2

Beginn, Ende und Wechsel der Elektrifizierung (*electrificationChanges*)

Die Struktur **<electrificationChanges>** enthält für jeden Beginn, Wechsel und das Ende der Elektrifizierung eines Gleises je ein Element **<electrificationChange>**. Die gesamte Struktur ist optional und nicht angegeben, wenn das Gleis nirgendwo elektrifiziert ist.

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung in Metern		
absPos	absolute Kilometrierung in Metern		
description	Bezeichnung des Elements wie in FBS eingegeben	x	
dir	immer =‘up‘		
type	= <i>none</i> am Elektrifizierungs-Ende = <i>overhead</i> am Anfang oder Wechsel (In FBS werden keine Elektrifizierungsarten i. S. Stromschiene usw. unterschieden.)		
voltage	Fahrleitungsspannung in Volt		
frequency	Frequenz in Herz, nur bei Wechselspannung	x	

Streckenklassen (*axleWeightChanges*)

Die Struktur **<axleWeightChanges>** enthält für jeden Wechsel der zulässigen Achslast oder spezifischen Achsfahrmasse (Meterlast) eines Gleises je ein Element **<axleWeightChange>**. Die gesamte Struktur ist optional und nur angegeben, wenn diese Werte überhaupt in FBS definiert sind.

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung in Metern		
absPos	absolute Kilometrierung in Metern		
description	Bezeichnung des Elements wie in FBS eingegeben	x	
dir	immer =‘up‘		
value	Achslast in Tonnen		
meterload	spezifische Achsfahrmasse (Meterlast) in t/m		

Spurweite (*gaugeChanges*)

Die Struktur **<gaugeChanges>** enthält für jeden Wechsel der Spurweite eines Gleises je ein Element **<gaugeChange>**. Da *Wechsel* von Spurweiten hinreichend selten sein sollten, steht i. d. R. nur ein **<gaugeChange>** pro Gleis am Anfang des Gleises.

Attribut	Inhalt	optional	Bemerkungen
pos	relative Kilometrierung in Metern		
absPos	absolute Kilometrierung in Metern		
description	Bezeichnung des Elements wie in FBS eingegeben	x	
dir	immer =‘up‘		
value	Spurweite in mm		

Geschwindigkeitsprofile (*speedProfiles*)

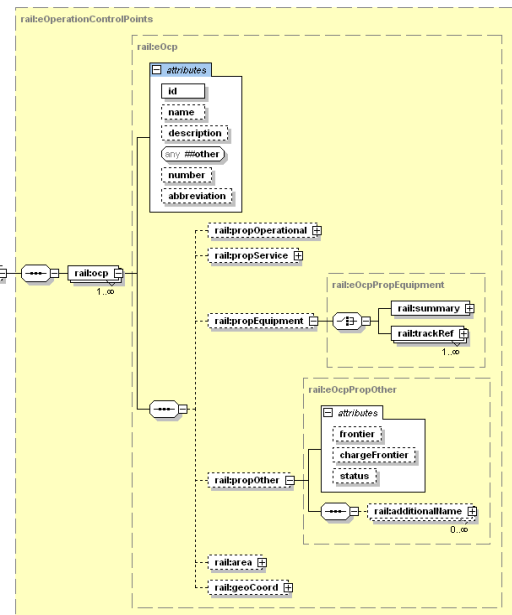
Die Struktur **<speedProfiles>** enthält für jedes Geschwindigkeitsprofil jedes Gleises einen Geschwindigkeitsprofil-Kopfeintrag in Form eines **<speedProfile>**-Elements. Ein Geschwindigkeitsprofil ergibt sich aus dem Kopfelement und allen **<speedChange>**-Elementen, die mit ihrer Eigenschaft *profileRef* auf das Kopfelement verweisen. Das Kopfelement beschreibt die Gültigkeit und andere Eigenschaften des Geschwindigkeitsprofils.

Attribut	Inhalt	optional	Bemerkungen
name	Bezeichnung des Geschwindigkeitsprofils wie vom Anwender in FBS eingegeben; wird i. d. R. nur benutzt, wenn zur Unterscheidung von anderen Geschwindigkeitsprofilen notwendig	x	
description	Hinweise zum Geschwindigkeitsprofil wie vom Anwender in FBS eingegeben. z. B. Datenquelle	x	
influence	= <i>decreasing</i> : Wirkt verringernd auf andere Geschwindigkeitsprofile (z. B. Achs- und Meterlastbeschränkung) = <i>increasing</i> : Wirkt erhöhend auf andere Geschwindigkeitsprofile (z. B. Neigetechnik)		nur relevant für „Zusatzprofile“, die mit anderen Profilen kombiniert werden; bei Grundprofilen immer = <i>increasing</i> (Tippfehlerkorrektur 10/2017)
maxAxleLoad	enthält die in FBS eingegebene Achslast, wenn das Profil nur für Fahrzeuge bestimmter Achslasten gilt	x	
maxMeterLoad	enthält die in FBS eingegebene Meterlast, wenn das Profil nur für Fahrzeuge bestimmter Meterlasten gilt	x	
<tilting>.maxTiltingAngle	enthält den max. Fahrzeugkasten-Neigewinkel, wenn das Geschwindigkeitsprofil für Fahrzeuge mit gleisbogenabh. Wagenkastenneigung gültig ist	x	
<tiltingSpeed>.tiltingSpeed	enthält die Verstellgeschwindigkeit, wenn das Geschwindigkeitsprofil für Fahrzeuge mit gleisbogenabh. Wagenkastenneigung gültig ist	x	

Betriebsstellen (*operationControlPoints*)

Die Struktur **operationControlPoints** enthält mindestens jede im Fahrplan (Struktur *timetable*) im Zuglauf oder in Umlaufplänen vorkommende Betriebsstelle in loser Folge. Dies sind mindestens (je nach Einstellung beim Export) die Betriebsstellen, an denen Züge beginnen, enden oder einen Verkehrshalt haben. Im Normalfall sind jedoch auch alle anderen betrieblich relevanten oder als Infrastruktur definierten Betriebsstellen bis hin z. B. zu Blocksignalen und Blockkennzeichen vorhanden.

Die Reihenfolge der Betriebsstellen entlang von Strecken wird (sofern vorhanden) durch die Strukturen *trackGroups* und *tracks* definiert.



Attribut	Inhalt	optional	Bemerkungen
name	(innerbetrieblicher) Name der Betriebsstelle	x	
abbreviation	Abkürzung der Betriebsstelle		nur V2.0.0
abbreviation	Abkürzung der Betriebsstelle		nur V2.0.5
code	Abkürzung der Betriebsstelle		nur V2.1
	FBS-interne Abkürzung der Betriebsstelle (nur bei FBS-intern erweiterten Abkürzungen) – nicht identisch mit <designator>.entry=...	x	ab V2.2.1
parentOcpRef	Querverweis auf eine Id eines anderen Elements <ocp>, das die übergeordnete Betriebsstelle enthält.	x	ab V2.2.1; *3
<designator> register=... entry=...	Abkürzung der Betriebsstelle; der register-Wert hängt von Anwendereinstellungen ab; üblich sind z.B. ,RL100' (Deutschland), ,DB640' (Österreich)	x	ab V2.2; mehrere Vorkommen sind möglich
number	„interne Betriebsstellennummer“ aus FBS-Betriebsstellenverzeichnis (sofern dort definiert); typischer Weise verwendet für IBNR	x	nur bis V2.1
<designator> register=IBNR entry=...	Internationale Bahnhofnummer (IBNR) sofern in FBS definiert	x	ab V2.2
<propOperational>.operationalType	nur vorhanden, wenn in FBS als funktionale Betriebsstelle definiert (ensuresTrainSequence=true): =junction bei Abzweigstelle =crossover bei Überleitstellen =station bei allen übrigen Zugmeldestellen =blockSignal bei Blocksignalen =blockPost bei allen übrigen Zugfolgestellen	x	
<propOperational>.orderChangeable	=true bei Zugmeldestellen, sonst =false	x	
<propOperational>.ensuresTrainSequence	=true bei Zugfolgestellen (einschl. Zugmeldestellen), sonst =false	x	
<propOperational>.uptime	sofern in FBS (zeitweise) ausgeschaltet oder (permanent) außer Betrieb, wird die Ausschaltzeitspanne mit mode=off angegeben.	x	*1
<propService>.passenger	=true, wenn in FBS als Zugangsstelle Reiseverkehr definiert	x	default=false

<propService>.service	=true, wenn in FBS mit Servicekategorie definiert	x	default=false
<propService>.ship	=true, wenn in FBS mit Übergang (Verknüpfungsstelle) zum Schiff definiert	x	default=false
<propService>.bus	=true, wenn in FBS mit Übergang (Verknüpfungsstelle) zum Bus definiert	x	default=false
<propService>.tariffpoint	=true, wenn in FBS als Tarifpunkt oder -grenze definiert	x	default=false
<propService>.goodsLoading	=true, wenn in FBS als Umschlagbahnhof oder Lade- stelle definiert	x	default=false
<propService>.goodsSiding	=true, wenn in FBS als Anschlussstelle definiert	x	default=false
<propService>.goodsIntermodal	=true, wenn in FBS als Umschlagbahnhof definiert	x	default=false
<propService>.goodsMarshalling	=true, wenn in FBS als Rangierbahnhof definiert	x	default=false
<propEquipment>. <summary>.hasHomeSignals	=true, wenn in FBS als Bahnhof mit Einfahrsignalen definiert, sonst =false	x	
<propEquipment>. <summary>.hasStarterSignals	=true, wenn in FBS als Betriebsstelle mit Ausfahr- oder Blocksignalen definiert, sonst =false	x	
<propEquipment>. <summary>.hasSwitches	=true, wenn in FBS als Zugmeldestelle, jedoch nicht als Endpunkt definiert, sonst =false	x	
<propOther>.status	=planned, wenn in FBS als in Planung definiert =closed, wenn in FBS als außer Betrieb oder stillgelegt definiert	x	*1
<propOther>. <additionalName>.value.type	.value: alternativer Name der Betriebsstelle .type: s. unten		nur bis V2.1 *2
<additionalName>.name.type	.name: alternativer Name der Betriebsstelle .type=operationalName bei abweichender Schreibweise des innerbetrieblichen Namens der Betriebsstelle im Bildfahrplan vom Betriebsstellenverzeichnis .type=trafficName bei abweichender Schreibweise des verkehrlichen Namens, alternativer Sprache (Görlitz - Zhorjelc) oder alternativem Zeichensatz (Thessaloniki - ΘΕΣΣΑΛΟΝΙΚΗ).	x	type=trafficName kann mehrfach vorkommen (verschiedene Sprachen) erst ab V2.2
<area>.name	Name der Gemeinde der Betriebsstelle	x	*2
<area>.number	Nummer der Gemeinde der Betriebsstelle	x	*2
<area>.zip	Postleitzahl der Betriebsstelle	x	*2
<geoCoord>.coord	Längen- und Breitengrad bzw. x- und y-Koordinate (je 6 Nachkommastellen) und Höhe*2 (2 Nachkommastellen) der Betriebsstelle	x	nur bis V2.1
	Breiten- und Längengrad bzw. y- und x-Koordinate (je 6 Nachkommastellen) der Betriebsstelle	x	erst ab V2.2
<geoCoord>.extraHeight	enthält die Höhe der Betriebsstelle*2	x	erst ab V2.2
<geoCoord>.heightEpsgCode	enthält den EPSG-Code des Referenzsystems der Höhenangabe	x	

*1 Funktionale Betriebsstellen (Zugmelde- und Zugfolgestellen), die mit *propOperational.uptime.mode=off* oder *propOther.status=closed* definiert sind, wirken wie funktionslose Betriebsstellen (d. h. beeinflussen die Zugfolge nicht).

*2 sofern im FBS-Betriebsstellenverzeichnis entsprechend definiert

*3 In Fällen hierarchisch strukturierter Betriebsstellen sind bestimmte externe Referenzen (z. B. RL100, IBNR usw.) nur bei übergeordneten Betriebsstellen vorhanden.

Bei widersprüchlichen Angaben zwischen FBS-Bildfahrplan und FBS-Betriebsstellenverzeichnis wird die Angabe aus dem FBS-Bildfahrplan ausgegeben (betrifft insbesondere *propOperational* und *propService*). Betriebsstellen (nur Zugmeldestellen), die in FBS streckenabhängig unterschiedliche Eigenschaften haben (z. B. auf einer Strecke Bahnhof, auf einer anderen Strecke Abzweigstelle), werden mit der Vereinigungsmenge der Eigenschaften exportiert.

In mehreren Ausprägungen des Unter-Elements **<designator>** (ab V2.2) kann die Abkürzung und/oder Nummer, d. h. je ein externer Primärschlüssel der Betriebsstelle angegeben sein. Das Attribut *register* des Elements bestimmt, aus welchem Verzeichnis die Abkürzung oder Nummer stammt. Typische Ausprägungen sind *register='DS100'* für Abkürzungen der DB Netz AG, *register='IBNR'* für Internationale Bahnhofsnummern oder *register='DB640'* für Abkürzungen der ÖBB Infrastruktur AG.

ACHTUNG: FBS garantiert nicht zwangsläufig die Eindeutigkeit (Primärschlüssel-Eigenschaft) aller exportierten Register. Es ist möglich, dass in railML mehrere Betriebsstellen die gleiche Codierung (*register/entry*-Kombination) haben. Dies ist z. B. bei Infrastruktur der DB Netz AG mit *register=RL100* bei Betriebsstellen an parallelen Strecken so (Beispiel: Haltepunkt Kürbitz, DB-Abkürzung DKUR; Templin Stadt, DB-Abkürzung WTV). Bei *register=IBNR* haben beispielsweise die drei Ausprägungen von *Berlin Hauptbahnhof - Lehrter Bahnhof* mit den RL-100-Abkürzungen BL, BLS und BHBF die gleiche IBNR. Durch das gleiche Kürzel wird in all diesen Fällen indirekt codiert, dass hier eine verkehrliche Verbindung (Umsteigemöglichkeit) zwischen betrieblich getrennten Stellen vorliegt.

Verwaltungsdaten (organizationalUnits)

Einige Verwaltungsinformationen aus der Infrastruktur (z. B. Eigner von Strecken) wie auch aus Fahrplandaten (z. B. Besteller) werden in RailML in der zentralen Struktur <organizationalUnits> abgebildet. Sofern solche Informationen vorkommen, ist diese Struktur einmalig unter <railML>.<metadata> vorhanden. Darunter können beliebig viele Vorkommen der folgenden Elemente enthalten sein:

Element	Inhalt	optional	Bemerkungen
<infrastructureManager>	Eigner einer Strecke / eines Gleises, aus <line> oder <ownerChange> oder benutzerdefiniert	x	
<vehicleManufacturer>	Fahrzeughersteller, aus <vehicle>.<classification>.<manufacturer> oder benutzerdefiniert	x	
<vehicleOperator>	derzeit nur benutzerdefiniert	x	
<customer>	derzeit nur benutzerdefiniert	x	
<railwayUndertaking>	derzeit nur benutzerdefiniert	x	
<operationalUndertaking>	derzeit nur benutzerdefiniert	x	
<concessionaire>	derzeit nur benutzerdefiniert	x	
<contractor>	derzeit nur benutzerdefiniert	x	

Sofern vorkommend, trägt jedes dieser Elemente die folgenden Attribute:

Attribut	Inhalt	optional	Bemerkungen
id	Eindeutiger Bezeichner innerhalb der RailML-Datei, der als Verweisziel für Referenzen dient		
name	Name der Verwaltungseinheit / Institution; oftmals auch nur dessen Firmenkürzel		

Zu jedem Element unter <organizationalUnits> gibt es mindestens eine entsprechende Referenz. Eine Referenz kann sein

- Ein Element <organizationalUnitBinding> mit dem Attribut **ref** und dem gleichen Wert wie **id** des zugehörigen <organizationalUnits>-Unterelements.
- Ein Attribut infrastructureManagerRef mit dem gleichen Wert wie **id** des zugehörigen <infrastructureManager>-Elements.

Die Referenzen stehen an ihrem jeweiligen inhaltlich zutreffenden Vorkommen.

Fahrzeug- und Zugbildungsdaten (rollingstock)

Die Struktur **rollingstock** enthält grundsätzlich jedes im Fahrplan (Struktur *timetable*) vorkommende Fahrzeug (in **vehicles**) sowie die im Fahrplan vorkommenden Zugbildungen (in **formations**) in loser Folge, jedoch nicht alle möglichen (jedoch im exportierten Fahrplan nicht vorkommenden) Fahrzeuge.

Fahrzeugdaten (vehicles)

Die in **vehicles** angegebenen Informationen stellen nur einen groben Auszug aus den in FBS vorgehaltenen Informationen über die Fahrzeuge dar, die im Wesentlichen zum Interpretieren der Fahrpläne (verkehrlichen Eigenschaften) dienen. Insbesondere sind technische Angaben zu Zugkraft und Bremsen hier nicht vorgesehen, da als Hauptzweck dieser Schnittstellenimplementierung das Austauschen von Fahrplaninformationen angesehen wird. Zum Austausch von Tfz.-Daten gibt es eine spezielle FBS-Schnittstellenimplementierung.

Attribut	Inhalt	optional	Bemerkungen
name	Tfz.-Baureihe oder Wagengattung, ggf. ergänzt um Abkürzung des einstellenden EVUs (z. B. DB.234)		
axleSequence	Achsfolge / Achsformel	x	*1
numberDrivenAxles	Anzahl angetriebener Achsen	x	*1
numberNonDrivenAxles	Anzahl nicht angetriebener Achsen	x	optional bei Tfz.
trackGauge	Spurweite in Metern (mit Nachkommastellen)	x	
length	Länge über alles in Metern		
speed	zulässige Höchstgeschwindigkeit in km/h		
bruttoWeight	Gesamtmasse in Tonnen		
nettoWeight	Leermasse in Tonnen	x	
nettoAdhesionWeight	Reibungsmasse in Tonnen	x	*1
<classification>. <manufacturer>. manufacturerName	Kurzbezeichnung des Herstellers	x	*1
<classification>. <manufacturer>. manufacturerType	Bezeichnung des Fahrzeugs beim Hersteller	x	*1
<classification>. <operator>. operatorName	Kurzbezeichnung des Betreibers (EVUs)	x	*1
<classification>. <operator>. operatorClass	Bezeichnung des Fahrzeugs beim EVU	x	*1
<wagon>. kinematicEnvelope	Kurzbezeichnung des Lichtraumprofils	x	
<wagon>.<passenger>. drivingCab	=true bei Triebwagen	x	*1
<wagon>. <passenger>.tilting	=true bei Neigetechnik-Fahrzeugen	x	*1
<wagon>. <passenger>.places	.count: Anzahl Plätze u. a. Kategorie .category: mögliche Werte: <i>class1, class2, other: foldingSeat, standing, wheelchair, bicycle, couchette, bed, other: WC</i>	x	
<wagon>. <passenger>.service	.name: mögliche Werte: <i>WR, Bistro</i> .count=1 wenn vorhanden	x	

*1 nur bei Triebfahrzeugen

Zugbildungsdaten (*formations*)

Ein **formation**-Element enthält die Zugbildung eines Zugteils aus FBS. Dies kann eine Kombination aus Triebfahrzeug und Wagen sein, kann jedoch auch nur aus einem oder mehreren Wagen (z. B. einzelner Kurswagen oder Kurswagengruppe) oder nur aus Triebfahrzeugen bestehen (z. B. einzeln fahrende Lokomotive oder Triebwagen).

Attribut	Inhalt	optional	Bemerkungen
name	zusammenfassender Text aus Tfz.-Baureihe (sofern vorhanden), Wagengattungen (sofern vorhanden) und jeweiligen Anzahlen, z. B.: DB.234+2xBocm242+WLAB176+2xDDm...		
length	Summe der Länge aller Fahrzeuge in Metern		
speed	Minimum der zulässigen Höchstgeschwindigkeit aller Fahrzeuge in km/h		
weight	Summe der Bruttomasse aller Fahrzeuge in t		
<vehicleRef>. orderNumber	laufende Nummer der Fahrzeuggruppe im Zug		Zählung ab 1
<vehicleRef>. vehicleRef	Querverweis auf id des Fahrzeugtyps der Fahrzeuggruppe in der Struktur <i>vehicles</i>		
<vehicleRef>. vehicleCount	Anzahl der Fahrzeuge des angegebenen Typs in der Gruppe		

Fahrplandaten (*timetable*)

Fahrplanperioden (*timetablePeriods*)

Die Struktur **timetablePeriods** ist nur vorhanden, wenn im exportierten FBS-Netz eine Fahrplanperiode definiert ist. Sie enthält dann immer nur einen **timetablePeriod**-Eintrag, da eine FBS-Netzdatei nicht mehr als eine Fahrplanperiode haben kann und derzeit nicht aus mehreren Netzdateien gleichzeitig exportiert werden kann.

Bei nicht definierter Fahrplanperiode (nicht vorhandener *timetablePeriods*-Struktur) können in den folgenden Fahrplandaten nur allgemeine Verkehrstagerregelungen der sieben Wochentage (Mo-So) und Wochenfeiertage, jedoch kein konkreter Datumsbezug vorkommen.

Attribut	Inhalt	optional	Bemerkungen
name	Kurzbezeichnung der Fahrplanperiode; je nach Länge z. B. „2010“ oder „2010/2011“ oder „06-09/2010“ oder „22.-28.08.2010“		
startDate	erster Gültigkeitstag der Fahrplanperiode		
endDate	letzter Gültigkeitstag der Fahrplanperiode		
<holidays>. <holiday>. holidayDate	Aufzählung der Feiertage innerhalb der Fahrplanperiode (sofern definiert)	x	

Verkehrstagerregelungen / Saisonierungen (*operatingPeriods*)

Die Struktur **operatingPeriods** enthält für jede in den Fahrplan- und Umlaufdaten vorkommende Verkehrstagerregelung je einen Eintrag. Diese Struktur ist immer dann vorhanden, wenn Züge oder Umläufe in der RailML-Datei enthalten sind - auch dann, wenn keine Fahrplanperiode angegeben ist.

Attribut	Inhalt	optional	Bemerkungen
name	kurze Textdarstellung der Verkehrstagerregelung, z. B.: „täglich; nicht 24.,31.12.; 9.,16.,23.7.“		sprachabhängig
description	lange Textdarstellung der Verkehrstagerregelung, z. B.: „verkehrt Montag-Freitag, auch an Feiertagen, nicht am 24., 31.12.“		sprachabhängig
timetablePeriodRef	sofern vorhanden, Querverweis auf id der (einzigen definierten) Fahrplanperiode	x	*1
bitMask	Binärzahl* ² , die so viele Ziffern enthält wie die Fahrplanperiode Tage (d. h. Anzahl Ziffern = endDate - startDate + 1); gibt für jeden Tag der Fahrplanperiode an, ob die Verkehrstagerregelung an diesem Tag gilt (verkehrt, =1) oder nicht (=0).	x	*1
<operatingDay>. operatingCode	Binärzahl* ² mit 7 Ziffern für die Wochentage Montag (erste Ziffer) bis Sonntag (letzte Ziffer); gibt an, ob die Verkehrstagerregelung am jeweiligen Wochentag gilt (verkehrt, =1) oder nicht (=0).		
<operatingDay>. onRequest	=true, wenn Züge mit dieser Verkehrstagerregelung an den angegebenen Tagen nur bei Bedarf verkehren (d. h. ohne Ankündigung ausfallen können)	x	default=false
<operatingDay>. startDate	erster Gültigkeitstag der operatingDay-Angabe (liegt immer innerhalb der Fahrplanperiode)	x	*1
<operatingDay>. endDate	letzter Gültigkeitstag der operatingDay-Angabe (liegt immer innerhalb der Fahrplanperiode)	x	*1
<operatingDay>. <operatingDayDeviance>. operatingCode	Binärzahl* ² mit 7 Ziffern für die Wochentage Montag (erste Ziffer) bis Sonntag (letzte Ziffer); gibt an, dass die Verkehrstagerregelung abweichend von <i>operatingCode</i> verkehrt (=1) oder nicht verkehrt (=0), wenn der Wochentag in der unter <i>holidayOffset</i> angegebenen Relation zu einem Feiertag steht	x	
<operatingDay>. <operatingDayDeviance>. holidayOffset	Versatz in Anzahl Tagen des <i>operatingDayDeviance</i> -Elements zu einem Feiertag; z. B.: <i>holidayOffset</i> =0 bedeutet „an einem Feiertag“ <i>holidayOffset</i> =1 „einen Tag nach einem Feiertag“ <i>holidayOffset</i> =-1 „einen Tag vor einem Feiertag“	x	
<operatingDay>. <operatingDayDeviance>. ranking	definiert die Priorität bei Vorhandensein mehrerer gleichzeitig zutreffender <i>operatingDayDeviance</i> -Elemente (z. B. 25.12. ist gleichzeitig Feiertag und ein Tag vor einem Feiertag): <i>operatingDayDeviance</i> -Elemente mit niedrigerem <i>ranking</i> -Wert überschreiben solche mit höherem	x	
<specialService>	Aufzählung von Ausnahmeverkehrstagen, an denen von den durch die <i>operatingDay</i> -Elemente definierten Bildungsregeln abgewichen wird <i>.type=include</i> für Zusatzverkehrstage <i>.type=exclude</i> für Ausfalltage <i>.startDate</i> : erster Gültigkeitstag der Ausnahme <i>.endDate</i> : letzter Gültigkeitstag der Ausnahme <i>.singleDate</i> : verwendet im Fall <i>startDate=endDate</i>	x	*1

*1 nur bei definierter Fahrplanperiode

*2 Zeichenkette, bestehend nur aus den Zeichen 0 und 1

Jedes <operatingPeriod>-Element hat mindestens ein <operatingDay>-Element. Falls mehrere <operatingDay>-Elemente in einem <operatingPeriod>-Element vorkommen, sind diese durch ihre *startDate*-*endDate*-Bereiche oder ihre *operatingCode*-Bitmaske immer disjunkt.

Beispiele zu Verkehrstagerregelungen finden Sie in der iRFP-Beispielbeschreibung zu RailML.

Zuggattungen und Produktbezeichnungen (*categories*)

Die Struktur **<categories>** enthält je ein **<category>**-Element für jede im Fahrplan vorkommende Zuggattung und Produktbezeichnung.

Version 2.0.0:

Attribut	Inhalt	optional	Bemerkungen
name	Kurzbezeichnung / Abkürzung der Zuggattung (z. B. „ICE“)		nur in V2.0.0
description	Langbezeichnung der Zuggattung (z. B. „InterCityExpress“)	x	nur in V2.0.0

Version 2.0.5:

Attribut	Inhalt	optional	Bemerkungen
abbreviation	Kurzbezeichnung / Abkürzung der Zuggattung (z. B. „ICE“)		nur in V2.0.5
name	Langbezeichnung der Zuggattung (z. B. „InterCityExpress“)	x	erst ab V2.0.5
description	Beschreibung der Verwendung der Zuggattung, z. B. „internationale Ganzgüterzüge“	x	erst ab V2.0.5

ab Version 2.1.0:

Attribut	Inhalt	optional	Bemerkungen
code	Kurzbezeichnung / Abkürzung der Zuggattung (z. B. „ICE“)		erst ab V2.1
name	Langbezeichnung der Zuggattung (z. B. „InterCityExpress“)	x	
description	Beschreibung der Verwendung der Zuggattung, z. B. „internationale Ganzgüterzüge“	x	

in allen bisherigen Versionen:

Attribut	Inhalt	optional	Bemerkungen
trainUsage	= <i>goods</i> bei Gattungen für Güterzüge = <i>passenger</i> bei Gattungen für Reisezüge = <i>mixed</i> bei Gattungen für beide Zugarten	x	nicht vorhanden bei Gattungen für Tfz.-Leerfahrten oder Dienstzüge
deadrun	= <i>true</i> , wenn die Zuggattung für einen Leerzug steht (Leerreise- oder Leergüterzug oder Tfz.-Leerfahrt)	x	default= <i>false</i>

Die optionalen Attribute sind nur dann vorhanden, wenn die entsprechende Zuggattung bzw. das Produkt im FBS-Gattungsverzeichnis definiert wurden.

Züge und Zugteile (*trains* und *trainParts*)

Die Grundphilosophie der RailML-*timetable*-Version 2.0 ist, den vielfältigen Praxisanforderungen des Eisenbahnbetriebs wie z. B. Flügeln von Zügen, Verstärken, Wagenzugdurchlauf, Einsatz von Kurswagen usw. durch „Zerlegen“ von Zügen in kleinste, unteilbare Einheiten gerecht zu werden. Diese kleinsten unteilbaren Zugeinheiten werden **Zugteile** genannt. Die eigentlichen Zuginformationen wie Zeiten, Fahrzeuge usw. sind an den Zugteilen zu finden. Die **Zug**-Struktur (**trains**) in RailML hingegen fasst lediglich Zugteile zu Zügen zusammen, enthält aber darüber hinausgehend i. d. R. keine weiteren Informationen.

Weitere Informationen und Beispiele zu Zügen und Zugteilen finden Sie in der iRFP-Beispielbeschreibung zu RailML.

Zugteile (*trainParts*)

Die Struktur *trainParts* enthält je einen *trainPart*-Eintrag für jede zusammenhängende Teilmenge eines Zuges mit homogenen Eigenschaften. Während sich innerhalb eines Zuglaufes z. B. die Verkehrstage oder die Anzahl Fahrzeuge ändern können, sind innerhalb eines *trainParts* immer alle Eigenschaften konstant.

Attribut	Inhalt	optional	Bemerkungen																						
name	Zugteilnummer (in FBS vom Anwender eingegeben; kann alphanumerisch sein)																								
line	Linienbezeichnung des Zugteils aus FBS	x																							
trainNumber	Zugnummer des (betrieblichen) Trägerzuges aus FBS (s. a. <i><timetable></i> . <i><train></i> . <i>trainNumber</i>)		kann alphanumerisch sein																						
processStatus	<table border="0"> <tr> <td><u>Status in FBS</u></td> <td><u>Ausgabe in RailML</u></td> </tr> <tr> <td>angelegt/konstruiert</td> <td>'planned'</td> </tr> <tr> <td>zu bestellen</td> <td>'toBeOrdered'</td> </tr> <tr> <td>bestellt (EVU→EIU)</td> <td>'ordered'</td> </tr> <tr> <td>Bestellung bearbeitet (EIU)</td> <td>'edited'</td> </tr> <tr> <td>angeboten (EIU→EVU)</td> <td>'offered'</td> </tr> <tr> <td>bestätigt (EVU→EIU)</td> <td>'confirmed'</td> </tr> <tr> <td>veröffentlicht</td> <td>'published'</td> </tr> <tr> <td>zurückgestellt</td> <td>'placedBack'</td> </tr> <tr> <td>gefahren</td> <td>'driven'</td> </tr> <tr> <td>abgerechnet</td> <td>'settled'</td> </tr> </table>	<u>Status in FBS</u>	<u>Ausgabe in RailML</u>	angelegt/konstruiert	'planned'	zu bestellen	'toBeOrdered'	bestellt (EVU→EIU)	'ordered'	Bestellung bearbeitet (EIU)	'edited'	angeboten (EIU→EVU)	'offered'	bestätigt (EVU→EIU)	'confirmed'	veröffentlicht	'published'	zurückgestellt	'placedBack'	gefahren	'driven'	abgerechnet	'settled'		sofern ergänzend zu RailML mit 'other:'+...
<u>Status in FBS</u>	<u>Ausgabe in RailML</u>																								
angelegt/konstruiert	'planned'																								
zu bestellen	'toBeOrdered'																								
bestellt (EVU→EIU)	'ordered'																								
Bestellung bearbeitet (EIU)	'edited'																								
angeboten (EIU→EVU)	'offered'																								
bestätigt (EVU→EIU)	'confirmed'																								
veröffentlicht	'published'																								
zurückgestellt	'placedBack'																								
gefahren	'driven'																								
abgerechnet	'settled'																								
remarks	Bemerkungen zum Zugteil aus FBS	x	erst ab V2.0.3																						
debitcode		x																							
operator		x																							
<i><organizationalUnit-Binding></i>	vom Anwender als Ausgabeziel für benutzerdefinierte Felder anwählbar	x	erst ab V2.2																						
timetablePeriodRef	sofern das FBS-Netz eine definierte Fahrplanperiode hat, enthält dieses Feld die id der Fahrplanperiode aus der Struktur <i>timetable.timetablePeriods</i> der RailML-Datei.	x																							
categoryRef	Querverweis auf id des Eintrags der Produktbezeichnung (verkehrlichen Gattung) des Zugteils in der Struktur <i>timetable.categories</i> der RailML-Datei	x	s. a. <i><train></i> . <i><trainPartSequence></i> . <i>categoryRef</i>																						
<i><formationTT></i> . <i>formationRef</i>	Querverweis auf id des Eintrags der Zugbildung des Zugteils in der Struktur <i>rollingstock.formation</i> s der RailML-Datei																								
<i><formationTT></i> . <i>orientationReversed</i>	= <i>True</i> , wenn die Formation im Zug in umgekehrter Reihung verkehrt ggü. ihrer Definition	x	erst ab V2.2																						
<i><formationTT></i> . <i>weight</i>	Gesamtmasse des Zugteils in t		*1																						
<i><formationTT></i> . <i>load</i>	Last des Zugteils in t (Masse ohne Tfz.)		*1 nur bis V2.1																						

<formationTT>.timetableLoad			*1 erst ab V2.2
<formationTT>.length	Länge des Zugteils in m		*2
<formationTT>.speed	Höchstgeschwindigkeit in km/h		*1
<formationTT>.<equipmentUsage>.equipment	Zugbeeinflussung des Trägerzuges des Zugteils	x	
<formationTT>.<passengerUsage>.<places>	.count: Anzahl Plätze u. a. Kategorie .category: mögliche Werte: <i>class1, class2, other: foldingSeat, standing, wheelchair, bicycle, couchette, bed, other: WC</i>	x	*1 *3
<formationTT>.<passengerUsage>.<service>	.name: mögliche Werte: <i>WR, Bistro</i> .count=1 wenn vorhanden	x	*1 *3
<formationTT>.<reservationInfo>.booking	enthält Auflistung der Wagennummern für die Platzreservierung (Reservierungs-/Buchungsnummern) zu jedem Fahrzeug in der Formation .bookingNumber: Reservierungsnummer .posInFormation: Index des Wagens innerhalb der Formation (Zählung ab 1)*4 .vehicleRef: Querverweis auf id des Fahrzeugs innerhalb der Struktur <i>rollingstock.vehicles</i>	x	erst ab V2.0.5
<operatingPeriodRef>.ref	Querverweis auf Eintrag der Verkehrstagerregelung des Zugteils in der Struktur <i>timetable.operatingPeriods</i> der RailML-Datei <i>Die übrigen Felder von operatingPeriodRef werden nicht benutzt, da die sich dann ergebenden Verkehrstagerregelungen keine Bitmaske hätten.</i>		

*1 kann vom entsprechenden Wert der Formation abweichen

*2 ist identisch mit dem entsprechenden Wert der Formation

*3 Wenn die Fahrzeug-Formation eines Zugteils Plätze besitzt (Sitzplätze, Bettplätze usw.) und der Zugteil nicht öffentlich ist (in FBS: Gattung oder Produkt nicht als Reisezug definiert oder Zugteil explizit auf „nicht veröffentlichen“ gesetzt), werden die Sitzplätze der Formation unter *trainPart.formationTT.passegerUsage.places* explizit mit *count=0* angegeben (überschrieben). Wenn der Zugteil öffentlich ist (in FBS: Gattung oder Produkt als Reisezug definiert und Zugteil nicht explizit auf „nicht veröffentlichen“ gesetzt), sind die Sitzplätze des Zugteils identisch mit denen der Fahrzeug-Formation. Wenn Gattung und Produkt des Zugteils nicht angegeben oder undefiniert sind, fehlt auch die Struktur *trainPart.formationTT.passegerUsage*.

*4 Wenn Fahrzeuge ohne Buchungsnummer in der Formation sind, werden diese von *posInFormation* übersprungen; *posInFormation* ist dann nicht fortlaufend. *posInFormation* beginnt ab 1 für das erste Fahrzeug zu zählen; wenn das erste Fahrzeug ein Triebfahrzeug ist und keine Buchungsnummer hat, ist der erste *posInFormation*-Wert demzufolge 2.

Im Allgemeinen gibt es zu jedem Zug einen Status der Bearbeitung. Aus diesem geht hervor, wo der Zug in der Planung bzw. „Verhandlung zwischen EVU und EIU“ steht. Alle letztendlich zu veröffentlichenden Züge sollten beim Infrastrukturunternehmen einen abgeschlossenen Trassenvertrag haben, d. h. mindestens den Status *confirmed*. Der Rang der Werte des Attributs *processStatus* ist steigend von oben nach unten. Es sind (aus FBS heraus) folgende Zustände möglich:

<i>planned</i>	Zug in FBS vorhanden, aber nicht zum Bestellen freigegeben
<i>other:toBeOrdered</i>	Zug soll beim EIU bestellt werden, ist aber noch nicht bestellt
<i>other:ordered</i>	Zug wurde beim EIU bestellt
<i>other:edited</i>	Zug wurde vom EIU bearbeitet
<i>other:offered</i>	Zug wurde vom EIU an EVU angeboten
<i>other:confirmed</i>	EVU hat Trasse beim EIU bestätigt = Vertrag ist gültig

<i>other:published</i>	EVU hat Trasse veröffentlicht bzw. zum Veröffentlichen freigegeben
<i>other:placedBack</i>	Zug wurde zurückgestellt - vmtl. Detailverhandlungen zur Trasse
<i>other:driven</i>	Zug ist bereits gefahren
<i>other:settled</i>	Zug ist gefahren und abgerechnet

Laufweg von Zugteilen (*ocpsTT*)

Die Struktur **<ocpsTT>** jedes *trainPart*-Elements enthält ein **<ocpTT>**-Element für jede Betriebsstelle, die der Zugteil auf seinem Laufweg passiert. Es sind immer mindestens zwei Betriebsstellen vorhanden; je nach Export-Einstellungen können jedoch bestimmte Zwischenbetriebsstellen „übersprungen“ werden (z. B. reine Blocksignale/Zugfolgestellen oder auch Bahnhöfe, an denen kein Verkehrshalt stattfindet).

Attribut	Inhalt	optional	Bemerkungen
sequence	laufende Nummer, ab 1 gezählt		erst ab V2.2
ocpRef	Querverweis auf die id der Betriebsstelle in der Struktur <i>infrastructure.operatingControlPoints</i>		
ocpType	= <i>pass</i> bei Durchfahrt * ¹ = <i>begin</i> bei beginnendem Gesamtzuglauf = <i>end</i> bei endendem Gesamtzuglauf = <i>stop</i> bei Betriebs- oder Verkehrshalt		<i>begin</i> und <i>end</i> nur bis V2.1
trackInfo	Gleisbezeichnung des Bahnhofsgleises, welches der Zug(teil) innerhalb der Betriebsstelle benutzt; keine Angabe steht hier i. A. für das durchgehende Hauptgleis der jeweiligen Fahrtrichtung	x	
remarks	Bemerkungen des Anwenders zum Halt oder zur Durchfahrt an dieser Betriebsstelle	x	
trainReverse	= <i>true</i> , wenn der Zug(teil) an dieser Betriebsstelle die Fahrtrichtung wechselt („kopfmacht“)	x	default= <i>false</i>
<times>.scope	= <i>scheduled</i>		
	= <i>published</i>	x	
<times>.arrival	Ankunftszeit des Zuglaufs an dieser Betriebsstelle	x	nicht vorhanden bei Durchfahrten und an der ersten Betriebsstelle des Gesamtlaufs
<times>.arrivalDay	Anzahl der Mitternachtsübergänge (Folgetage) zwischen Abfahrt des ersten Zugteils am ersten Bahnhof und Abfahrt an dieser Betriebsstelle * ²	x	
<times>.departure	Abfahrts- oder Durchfahrtszeit an dieser Betriebsstelle	x	nicht vorhanden an der ersten Betriebsstelle des Gesamtlaufs
<times>.departureDay	Anzahl der Mitternachtsübergänge (Folgetage) zwischen Abfahrt des ersten Zugteils am ersten Bahnhof und Abfahrt an dieser Betriebsstelle * ²	x	
<sectionTT>	enthält Informationen zum Abschnitt zwischen der aktuellen und der nächsten Betriebsstelle im Zuglauf		enthält <i>keine Attribute</i> an der letzten Betriebsstelle
<sectionTT>.section	Abkürzung der aktuellen und der nächsten Betriebsstelle im Zuglauf, durch - getrennt	x	
<sectionTT>.lineRef	Querverweis auf id der Strecke aus der Struktur <i>infrastructure.trackGroups</i> , welche vom Zug von hier bis zur nächsten Betriebsstelle benutzt wird	x	
<sectionTT>.trackRef	Querverweis auf id des Streckengleises aus der Struktur <i>infrastructure.tracks</i> , welches vom Zug von hier bis zur nächsten Betriebsstelle benutzt wird * ³	x	abweichend von RailML; nur V2.0.5
<sectionTT>.<trackRef>.ref		x	ab V2.1

<sectionTT>. <trackRef>. dir	Fahrtrichtung auf dem Gleis in relative km-Richtung: ='up' wenn in Definitionsrichtung des Gleises, ='down' wenn entgegen Definitionsrichtung des Gleises	x	ab V2.1
<sectionTT>.trackInfo	=1 für Regelgleis =2 für Gegengleis	x	nur auf zweigleisigen Abschnitten * ³
<sectionTT>. percentageSupplement	linearer Fahrzeitzuschlag zwischen der aktuellen und der nächsten Betriebsstelle in Prozent	x	
<sectionTT>. distance	Entfernung zwischen der aktuellen und der nächsten Betriebsstelle in km (V2.0.x) bzw. in m (ab V2.1)	x	* ⁴
<sectionTT>. <runTimes>. minimalTime	kürzeste (berechnete) Fahrzeit zwischen der aktuellen und der nächsten Betriebsstelle ohne linearen Fahr- zeitzuschlag	x	
<sectionTT>. <runTimes>. operationalReserve	Differenz aus kürzester und planmäßiger Fahrzeit der aktuellen und der nächsten Betriebsstelle	x	
<sectionTT>. <runTimes>. additionalReserve	nichtlinearer Fahrzeitzuschlag (sofern vorhanden), d. h. planmäßige Fahrzeit - kürzeste Fahrzeit - linearer Fahrzeitzuschlag = operationalReserve - linearer Fahrzeitzuschlag	x	
<stopDescription>...	Enthält Informationen zum Halt an der aktuellen Be- triebsstelle	x	nicht vorhanden bei Durchfahrten
<stopDescription>. commercial	=True bei Verkehrshalt =False bei Betriebshalt	x	
<stopDescription>. stopOnRequest	=True bei Bedarfshalt (bedingtem Verkehrshalt) =False bei unbedingtem Verkehrshalt	x	nur bei <stopDescription>. commercial=True
<stopDescription>. onOff	=on für „hält nur zum Einsteigen“ =off für „hält nur zum Aussteigen“	x	
<stopDescription>. operationalStop- Ordered	=True bei Betriebshalt EVU =False bei Betriebshalt EIU	x	erst ab V2.2, nur bei <stopDescription>. commercial=False
<stopDescription>. <stopTimes> minimalTime	Regelaufenthaltszeit des Zuges an dieser Betriebsstel- le (kann von der planmäßigen Aufenthaltszeit abwei- chen)	x	

*¹ *pass* kann auch an der ersten oder letzten Betriebsstelle vorkommen bei durchgehend ineinander übergebenden Zugteilen (Laufwegabschnitten)

*² Die Tageszählung von *arrivalDay* und *departureDay* beginnt mit 0 an der Abfahrt des ersten Zugteils einer Reihe sequentiell zusammenhängender Zugteile. Daher kann der Wert >0 sein an der ersten Betriebsstelle eines Zugteils.

*³ Die Angaben *sectionTT.trackRef* und *sectionTT.trackInfo* sind redundant: Wenn der Zug(teil) auf dem Gegengleis fährt (*sectionTT.trackInfo*=2), verweist auch *sectionTT.trackRef* auf die *id* des jeweiligen (abhängig von der Fahrtrichtung) Gegengleises.

*⁴ Die Einheit des Attributs *distance* wurde mit V2.1 von km auf m geändert nach Konkretisierung innerhalb von RailML. Aus diesem Grunde wurde mit V2.1 die Kompatibilitätsnummer (**metadata.identifizier**) von 1 auf 2 erhöht.

Der Anwender kann beim Export wählen, ob die Ankunfts-/Abfahrtszeiten

- auf ganze Minuten gerundet (nach wählbarer Rundungsregel),
- mit Sekundenbruchteilen,
- zehntelminutengenau oder
- sekundengenau

ausgegeben werden sollen. In jedem Falle sind die Zeitangaben xml-konform; es werden nur mehr oder weniger Stellen ausgegeben bzw. (im Falle von Zehntelminuten) nur bestimmte (ganzzahlig durch 6 teilbare) Sekundenwerte wie :00, :06, :12, :18 usw.

In der Eisenbahnwelt ist es jedoch üblich, Zeiten nur 0,1-min-genau zu planen; dem Anwender von FBS werden die Zeiten i. d. R. auch nur auf 0,1 min angezeigt. Dies sollte beim Auswerten der Ankunfts- und Abfahrtszeiten beachtet werden. **Genauere Angaben als Zehntelminuten dienen nur internen Zwecken (Vermeidung von Rundungsfehlern) und sollten nicht angezeigt oder gar veröffentlicht werden. Es wird empfohlen, solche Werte auf 6 Sekunden / 0,1 min zu runden.**

Bei in das FBS-Netz „einbrechend“ beginnendem oder „ausbrechend“ endendem Zuglauf werden die Übergabezeiten in das / aus dem Netz nicht angegeben, da dies potentiell Widersprüche zu *ocpType* (Halteart: *begin* oder *stop*?) und problematische Mitternachtsübergänge (*arrivalDay*=-1 bei Mitternachtsübergang im Abfahrtsbahnhof) ergeben würde. Bei durchfahrend ein- oder ausbrechenden Zügen wird jedoch zwangsläufig auch die Übergabezeit angegeben.

Betriebliche und verkehrliche Züge (trains)

Das Zusammensetzen der Zugteile zu Zügen kann aus zwei verschiedenen Sichtweisen erfolgen: aus betrieblicher und aus verkehrlicher Sicht.

Das Zusammensetzen aus betrieblicher Sicht ergibt Züge, wie sie der Eisenbahner und das Eisenbahninfrastrukturunternehmen „sehen“, jedoch nicht unbedingt wie sie der Reisende sieht. Die Grundeigenschaft betrieblicher Züge ist, dass zu einem Zeitpunkt auf einem Streckengleis nur ein Zug fahren kann. Dieser Zug muss klar durch eine betriebliche Zugnummer definiert sein. Dieser Sichtweise kommt teilweise ein Sicherheitsaspekt zu (z. B. Verständigung zwischen den Betriebsstellen); sie wird daher als grundlegend empfunden.

Das Zusammensetzen aus verkehrlicher Sicht ergibt „Züge“, wie der Reisende sie i. d. R. sieht und wie sie z. B. in Aushang- oder Tabellenfahrplänen (Kursbuch) oder elektronischen Fahrplanmedien angegeben werden. Hierbei können zu einem Zeitpunkt auf (quasi) einem Streckengleis mehrere (verkehrliche) Züge gleichzeitig „fahren“: In einem Tabellenfahrplan werden gekuppelte Züge durch zwei benachbarte Spalten mit gleichen Zeiten dargestellt. In einem Aushangfahrplan können zwei Zeilen mit gleicher Abfahrtszeit und gleichem Gleis vorhanden sein, wobei beide Zeilen unterschiedliche Ziele angeben, die Zugteile aber erst im späteren Zuglauf getrennt werden. Bedingt durch die Sichtweise trifft der Begriff „Zug“ hier nur im weitesten Sinne zu: So müssen „verkehrliche Züge“ nicht unbedingt ein Triebfahrzeug haben (z. B. Kurswagen).

Jeder Zugteil wird durch genau je einen betrieblichen und einen verkehrlichen Zug erfasst. (Der Export verkehrlicher Züge aus FBS ist jedoch optional: Wenn verkehrlichen Züge exportiert worden sind, wird jeder Zugteil genau durch einen betrieblichen und einen verkehrlichen Zug erfasst, ansonsten nur durch genau einen betrieblichen.)

Jeder Zug (<train>-Eintrag) führt alle Zugteile, aus denen er besteht, in seiner Struktur <trainPartRef> mit *ref* und *position* auf. Ein Zug kann dabei entweder gleichzeitig (an einer Stelle seines Zuglaufs) oder nacheinander (in aufeinanderfolgenden Abschnitten seines Zuglaufs) aus verschiedenen Teilen bestehen. Der Wert <trainPartRef>.position zählt die Stellung im Zug an einer Stelle im Zuglauf. Es kann daher mehrere *trainPartRef*-Einträge mit gleicher *position* geben, wenn die entsprechenden Zugteile an verschiedenen Abschnitten im Zuglauf vorkommen.

Mehrere Zugteile an einer Stelle eines Zuglaufs kommen z. B. beim (abschnittsweisen) Verstärken oder Flügeln von Zügen vor. Aufeinanderfolgende Abschnitte eines Zuglaufs kommen z. B. vor, wenn unterwegs Verkehrstage oder andere Eigenschaften wechseln.

Zu beachten ist, dass *trainPartRef.position* nicht unbedingt die tatsächliche Stellung im Zug angibt – durch unterschiedliche Verkehrstage der Zugteile können Teile mit niedrigerem *position*-Wert an bestimmten Tagen im Zugverband „fehlen“.

Betriebliche Züge (*operational trains*)

Betriebliche Züge sind in der RailML-Datei immer vorhanden, wenn Zugteile vorhanden sind. Sie sind in der Struktur **trains** enthalten und werden dort durch das Attribut **type=operational** von verkehrlichen Zügen unterschieden.

Attribut	Inhalt	optional	Bemerkungen
type	= <i>operational</i>		
trainNumber	innerbetriebliche Zugnummer; dient der eindeutigen Bezeichnung des Zuges (innerhalb eines Verkehrstags) beim Infrastrukturbetreiber		kann alphanumerisch sein
scope	dient der eindeutigen Bezeichnung des Zuges über alle Verkehrstage (Primärschlüsselbildung); mögliche Werte: <i>primary</i> , <i>secondary</i> , <i>secondaryStart</i> , <i>secondaryEnd</i> , <i>secondaryInner</i>		
additionalTrainNumber	dient der eindeutigen Bezeichnung des Zuges über alle Verkehrstage (Primärschlüsselbildung); laufende Nummer innerhalb gleicher <i>trainNumber</i> und <i>scope</i> -Werte	x	ab V2.0.5
<trainPartSequence>.sequence	laufende Nummer des Abschnitts innerhalb des Zuges		Zählung ab 1
<trainPartSequence>.categoryRef	Querverweis auf id des Eintrags der betrieblichen Gattung des Zuges in der Struktur <i>timetable.categories</i> der RailML-Datei	x	erst ab V2.1 s. a. <trainPart>.categoryRef
<trainPartSequence>. <trainPartRef>.ref	Querverweis auf id eines Zugteils innerhalb der Struktur <i>timetable.trainParts</i>		Element kann mehrfach vorkommen
<trainPartSequence>. <trainPartRef>.position	Stellung des Zugteils im Zugverband, sofern ein Zug gleichzeitig aus mehreren Teilen besteht		Zählung ab 1
<trainPartSequence>. <speedProfileRef>.ref	Querverweis auf id eines Geschwindigkeitsprofils innerhalb der Struktur <infrastructure>.<speedProfiles>		Element kann mehrfach vorkommen
<equipmentUsage>. <equipment>	Zugbeeinflussung des Zuges	x	
<brakeUsage>...	Bremsangabe des Gesamtzuges (d. h. über alle eventuell gekuppelt verkehrenden Zugteile)		
<brakeUsage>.brakeType	= <i>compressedAir</i> bei Druckluftbremse = <i>vacuum</i> bei Saugluftbremse = <i>handBrake</i> bei handgebremstem Zug = <i>cableBrake</i> bei Seilzugbremse		
<brakeUsage>.airBrakeApplication-Position	nur bei Druck- oder Saugluftbremse: Grundbremsstellung = <i>G</i> , <i>P</i> oder <i>R</i>	x	
<brakeUsage>.meanDeceleration	nur wenn in FBS keine Brems Hundertstel definiert wurden: mittlere Bremsverzögerung in m/s ²	x	
<brakeUsage>.regularBrake-Percentage	im regulären (planmäßigen) Betrieb anrechenbare Brems Hundertstel (d. h. meist ohne Magnetschienenbremse)	x	erst ab V2.0.5

<brakeUsage>.emergencyBrake-Percentage	im Gefahrenfall (bei Not-/Schnellbremsungen) anrechenbare Bremsleistung (d. h. mit Magnetschienenbremse, jedoch ggf. ohne elektrische oder hydraulische Bremse)	x	erst ab V2.0.5
<brakeUsage>.brakePercentage	nur wenn in FBS identische Schnell- und Betriebsbremsleistung definiert wurden	x	in V2.0.0 auch wenn unterschiedlich
<brakeUsage>.auxiliaryBrakes	Aufzählung der planmäßig vorhandenen Zusatzbremsen: <i>Mg, Wb, E, H, ep</i>	x	

Durch das Attribut **scope** werden mehrere Züge mit gleicher Zugnummer (die disjunkte Verkehrstage haben müssen) unterschieden. Man kann diese auch als alternative zeitliche Lagen oder „Varianten“ ein und desselben Zuges auffassen. Das Attribut **additionalTrainNumber** dient der Unterscheidung von Einträgen mit gleicher Zugnummer und gleichem Scope-Attribut. In Anwendungsfällen, in denen die Zugnummer selbst immer eindeutig ist, haben *scope* und *scopeIndex* keine Bedeutung; hier ist immer *scope=primary*.

Den terminologischen Zusammenhang zwischen FBS-Bezeichnungen, RailML-Bezeichnungen und Bezeichnungen der DB Netz AG gibt folgende Tabelle wieder:

in RailML	in FBS	bei DB Netz
train.scope	Mehrfachzuglauf	„Ergänzungsfahrplan“
train.scope=primary	Hauptlauf	„Stammfahrplan“ oder „Stammzug“
train.scope=secondaryStart	Vornebenlauf	„Startflügel“
train.scope=secondaryEnd	Nachnebenlauf	„Zielflügel“
train.scope=secondaryInner	Zwischennebenlauf	„Doppelfahrplan“

Verkehrliche Züge (*commercial trains*)

Verkehrliche Züge sind in der RailML-Datei optional vorhanden, d. h. können beim Export explizit ein- oder ausgeschlossen werden. Sie sind in der Struktur **trains** enthalten und werden dort durch das Attribut **type=commercial** von betrieblichen Zügen unterschieden.

Attribut	Inhalt	optional	Bemerkungen
type	= <i>commercial</i>		
trainNumber	innerbetriebliche Zugteilnummer; dient der eindeutigen Bezeichnung des Zugteils (im Sinne einer durchlaufenden Fahrzeuggruppe)		kann alphanumerisch sein
name	Nummer zur Identifikation des verkehrlichen Zuges (z. B. Spaltentitel im Kursbuch - nicht unbedingt mit der betrieblichen Zugnummer identisch!)	x	nur bei Direktverbindungen; kann alphanumerisch sein
<trainPartSequence>.sequence	laufende Nummer des Abschnitts innerhalb des Zuges		Zählung ab 1
<trainPartSequence>.trainPartRef.ref	Querverweis auf id eines Zugteils innerhalb der Struktur <i>timetable.trainParts</i>		
<trainPartSequence>.trainPartRef.position	Stellung des Zugteils im Zugverband, sofern ein Zug gleichzeitig aus mehreren Teilen besteht		Zählung ab 1

Der Wert *<trainPartRef>.position* gibt bei verkehrlichen Zügen die Stellung des Zugteils im eigentlichen Zugverband (des betrieblichen Zuges!) an. Es können in der Zählung daher scheinbar Indizes fehlen, weil die entsprechenden Zugteile zu anderen verkehrlichen Zügen gehören.

Zugbeeinflussung, Bremsen usw. sind bei verkehrlichen Zügen nicht angegeben.

In der FBS-RailML-Schnittstelle können die verkehrlichen Züge in zwei Modi exportiert werden:

- a) Direktverbindungen mit untergeordneten Verstärkern,
- b) Direktverbindungen und Verstärker als unabhängige „commerical trains“.

Als *Direktverbindungen* werden solche verkehrlichen Züge bezeichnet, deren Laufweg ein Alleinstellungsmerkmal hat, d. h. die eine eigene umsteigefreie Verbindung darstellen.

Als *Verstärker* werden solche verkehrlichen Züge bezeichnet, deren Laufweg eine Teilmenge (gleich oder kleiner) mindestens eines anderen verkehrlichen Zuges ist, d. h. kein Alleinstellungsmerkmal darstellt. Verstärker können damit Direktverbindungen untergeordnet werden.

Für Tabellenfahrpläne und andere Auskunftsmidien gilt i. d. R., dass nur die Direktverbindungen, nicht jedoch die Verstärker veröffentlicht werden. Daher haben auch nur die Direktverbindungen das Attribut *name*; darin angegeben ist die übliche Spaltenbezeichnung des verkehrlichen Zuges in einem Tabellenfahrplan. (Über das Attribut *name* können im Modus (b) die Direktverbindungen von den Verstärkern unterschieden werden.)

Im Modus (b) *Direktverbindungen und Verstärker als unabhängige „commerical trains“* gilt die Regel, dass jeder verkehrliche Zug in jedem seiner Abschnitte (<trainPartSequence>) genau einen Zugteil (<trainPart>) referenziert und nicht mehr.

Im Modus (a) *Direktverbindungen mit untergeordneten Verstärkern* kann ein verkehrlicher Zug in seinen Abschnitten (<trainPartSequence>) mehr als einen Zugteil (<trainPart>) referenzieren, nämlich außer der Direktverbindung selbst noch die untergeordneten Verstärker.

Umlaufpläne (rosterings)

Beim Export kann der Anwender wählen, ob und in welchem Umfang die Umlaufpläne aus FBS exportiert werden sollen. Jeder Umlaufplan von FBS führt zu einem **rostering**-Element.

Ein Umlaufplan gilt i. d. R. nur für *eine* Fahrzeugbaureihe und ggf. nur für einen begrenzten (Teil-)Zeitraum der Fahrplanperiode. Es kann daher mehrere Umlaufpläne für die gleiche Fahrzeugbaureihe geben. Die Teilzeiträume müssen nicht disjunkt sein, sondern können sich überlappen. 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. Das Überschreiben von Umlaufplänen ist in der Praxis notwendig, da zu Beginn eines Fahrplanjahres i. d. R. noch nicht in jedem Falle feststeht, wann der Grundumlaufplan nicht gefahren werden kann (z. B. wann Bauarbeiten sein werden).

In RailML sind **offene** und **geschlossene Umläufe** möglich. Geschlossene Umläufe können (theoretisch beliebig oft) wiederholt werden, da auf die letzte Fahrt im Umlauf wieder die erste Fahrt folgt. Offene Umläufe hingegen müssen nicht wiederholt werden können - bei ihnen folgt auf die letzte Fahrt entweder ein anderer Umlaufplan oder das Ende der Fahrplanperiode. Typischer Weise sind geschlossene Umläufe solche in einem Grund- oder Regelumlaufplan und offene Umläufe solche in einem Sonderumlaufplan. Im Allgemeinen kann man davon ausgehen, dass geschlossene Umläufe über einen längeren Zeitraum gelten (daher die Wiederholung, um nicht jeden Tag des längeren Zeitraums individuell abzubilden), während offene Umläufe über einen kürzeren Zeitraum gelten, in dem keine Wiederholungen möglich oder notwendig sind. Eine typische Zeitgrenze ist eine Woche, denn spätestens nach einer Woche sind Umlaufwiederholungen üblich. Offene Umläufe sind daher meist (fast immer) kürzer als eine Woche.

Wichtig für RailML ist, dass auf Grund der Möglichkeit offener Umläufe zu bestimmten Fahrten keine nachfolgende Fahrten angegeben sind. Zur begrifflichen Eindeutigkeit werden im Folgenden als *offene* Umlaufpläne solche bezeichnet, bei denen nicht an allen Fahrten eine Folgefahrt angegeben ist. *Geschlossene* Umläufe sind demzufolge solche, bei denen an allen Fahrten eine Folgefahrt angegeben ist - ungeachtet ihrer Gültigkeitsdauer und der Frage, ob sie tatsächlich wiederholt werden sollen.

Ein Umlaufplan kann mehrere **Umlaufgruppen** enthalten. Umlaufgruppen sind eine Eigenart insbesondere von Eisenbahn-Umlaufplänen: Jede Umlaufgruppe stellt für sich einen eigentlichen Umlauf dar; daher kann es sein, dass in anderen terminologischen Zusammenhängen das als *Umlauf* bezeichnet wird, was in der (deutschsprachigen) Eisenbahn-Fachwelt nur eine *Umlaufgruppe* ist. Jede Umlaufgruppe könnte theoretisch in einem eigenen Umlaufplan abgebildet werden. Dies wird in der Praxis aber oft so nicht vorgenommen, weil man u. U. bemüht ist, *alle* Fahrzeuge einer Baureihe in *einem* Umlaufplan darzustellen.

Wagenzugumläufe können für einzelne Wagen oder Wagengruppen vorhanden sein. Triebfahrzeugumläufe gelten für Einzelfahrzeuge (wobei in diesem Sinne jedoch *betrieblich* nicht trennbare Einheiten als Einzelfahrzeug gelten, z. B. ein Triebzug der BR 403 oder eine Kombination aus BR 628+928).

Umlauf-Kopfangaben (*rostering*)

Jeder Umlaufplan wird in einem **rostering**-Element abgebildet. Die Attribute des *rostering*-Elements enthalten allgemeine Angaben zum Umlaufplan:

Attribut	Inhalt	optional	Bemerkungen
name	Bezeichnung des Umlaufplans durch den Anwender in FBS (enthält meist Fahrzeugbezeichnung)		
description	Überschrift des Umlaufplans durch den Anwender in FBS sowie Fahrzeugbezeichnung	x	
vehicleRef	Querverweis auf id des Fahrzeugs des Umlaufplans im der Struktur <i>rollingstock.vehicles</i> (nur bei Umläufen von Einzelfahrzeugen vorhanden)	x	
formationRef	Querverweis auf id des Wagenzugs des Umlaufplans im der Struktur <i>rollingstock.formations</i> (nur bei Wagenzugumläufen vorhanden)	x	
defaultPreProcessingTime	definiert die (Standard-)Vorbereitungszeit, die gilt, wenn keine abweichende Angabe bei <i>blockPartSequence</i> erfolgt		
defaultPostProcessingTime	definiert die (Standard-)Nachbereitungszeit, die gilt, wenn keine abweichende Angabe bei <i>blockPartSequence</i> erfolgt		

Fahrten und Dienste (*blockParts*)

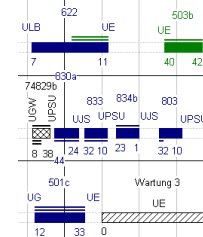
Die Struktur **blockParts** enthält je einen **blockPart**-Eintrag für jeden im Umlaufplan benötigten Zugteil oder Dienst. Ein Dienst (hier im Gegensatz zu Zugteil) kann z. B. Tanken, Wartung, Reinigung, Rangierdienst usw. sein und unterscheidet sich i. d. R. dadurch von den Zugteilen, dass die Betriebsstelle nicht gewechselt wird.

Attribut	Inhalt	optional	Bemerkungen
begin	Anfangszeit des Dienstes bzw. Abfahrtszeit des Zugteils an dessen ersten Bahnhof		

end	Endzeit des Dienstes bzw. Ankunftszeit des Zugteils an dessen letzten Bahnhof		
endDay	Anzahl der Mitternachtsübergänge (Folgetage) zwischen <i>begin</i> und <i>end</i>	x	default=0
startOcpRef	Querverweis auf id der Betriebsstelle des Anfangs des Dienstes oder Zugteils in der Struktur <i>infrastructure.operatingControlPoints</i>		
endOcpRef	Querverweis auf id der Betriebsstelle des Endes des Dienstes oder Zugteils in der Struktur <i>infrastructure.operatingControlPoints</i>		
trainPartRef	Querverweis auf id des Zugteils in der Struktur <i>timetable.trainParts</i> (sofern dem <i>blockPart</i> ein Zugteil zugrunde liegt)	x	
mission	= <i>timetable</i> bei Verweisen auf Zugteile = <i>fullRun</i> oder <i>emptyRun</i> bei Fahrten, die vom Anwender manuell in den Umlaufplan eingegeben wurden und daher nicht auf Zugteile verweisen = <i>shunting</i> bei Rangierdiensten = <i>maintenance</i> bei Wartungsdiensten = <i>standBy</i> bei Bereitschafts- und Reservediensten = <i>preheating</i> bei Vorheizdiensten = <i>refuel</i> bei Auffüllen von Vorräten (Tanken usw.)		
operatingPeriodRef	Querverweis auf id der Verkehrstagerregelung des Dienstes in der Struktur <i>timetable.operatingPeriods</i> der RailML-Datei; beschreibt alle Tage, an denen der Dienst im Umlauf vorkommt	x	nicht bei Zugteilen (d. h. nicht wenn <i>trainPartRef</i> angegeben ist)
runLength	Entfernung, die während des Dienstes angerechnet oder während der Fahrt zurückgelegt wird in km (mit drei Nachkommastellen)		

Fahrtengruppen (*blocks*)

Die Struktur **blocks** enthält für zusammenhängende Gruppen oder Abfolgen aus *blockParts* einen **block**-Eintrag. Jeder *blockPart* kommt in mindestens einem *block* vor. Im einfachsten (und recht häufig vorkommenden) Fall enthält ein *block* genau einen *blockPart*. Die Gruppierung von *blockParts* zu *blocks* ist in RailML im Allgemeinen willkürlich (nicht weiter spezifiziert); bei aus FBS exportierten Umläufen sind jedoch insbesondere die unmittelbar zu einem (betrieblich durchgehenden) Zug verketteten *blockParts* zu einem *block* zusammengefasst. Damit entsprechen die *blocks* im Wesentlichen den in grafischen Umlaufplänen typischer Weise dargestellten Blöcken (Rechtecken).



Attribut	Inhalt	optional	Bemerkungen
name	Titel des Blocks im FBS-Umlaufplan; i. d. R. bestehend aus Zuggattung/Produkt + Zugteilnummer + Verkehrstagekürzeln		
blockPartSequence.sequence	laufende Nummer der <i>blockPartSequence</i> -Elemente, gezählt ab 1		
blockPartSequence.blockPartRef.ref	Querverweis auf id eines Elements der Struktur <i>blockParts</i> des Umlaufplans		genau 1x je <i>blockPartSequence</i> vorhanden *1
blockPartSequence...preProcessingTime	Vorbereitungszeit vor Beginn des Block sofern abweichend von <i>rostering.defaultPreProcessingTime</i>	x	*2
blockPartSequence...postProcessingTime	Nachbereitungszeit nach Ende des Block sofern abweichend von <i>rostering.defaultPostProcessingTime</i>	x	*2

*1 Bei aus FBS exportierten RailML-Dateien kann ein *block* nicht gleichzeitig aus mehreren *blockParts* bestehen (d. h. durch ein Fahrzeug können nicht gleichzeitig mehrere Dienste oder Züge gefahren werden). Daher enthält jedes *blockPartSequence*-Element immer nur genau ein *blockPartRef*-Element.

*2 Die in Umläufen üblicherweise für Fahrten oder Dienste reservierte / in Anspruch genommene Zeit ist um die Vor- und Nachbereitungszeiten größer als die durch *begin* und *end* des jeweils ersten bzw. letzten *blockParts* definierte Zeitspanne.

blocks dienen auch dazu, die Mitternachtsgrenze im Sinne eines Betriebstages zu überwinden („Fahrten nach Mitternacht zählen zum Vortag“), indem Fahrten nach Mitternacht mit denen vor Mitternacht zu einem Block zusammengefasst werden. Da sich *operatingPeriodRef* eines Blocks bzw. eines *circulation*-Elements nur auf den Anfangstag bezieht, kann der Block in jeden beliebigen Folgetag hineinreichen - unabhängig davon, um welche Tagesart es sich handelt.

Umläufe (*circulations*)

Die eigentliche Verkettung (Reihenfolge) der im Umlauf vorkommenden Fahrten und Dienste (im Folgenden allgemein als „Blöcke“ bezeichnet) ist in der **circulations**-Struktur enthalten. Es ist je ein **circulation**-Element für jeden Block an jedem Umlaufstag enthalten. Durch die Attribute **nextBlockRef** und **nextOperatingPeriodRef** jedes *circulation*-Elements ergibt sich die Verkettung der Blöcke und damit die eigentliche Umlaufabfolge. Das Attribut **operatingPeriodRef** jedes *circulation*-Elements gibt (indirekt) an, ob und wie oft ein Umlaufstag wiederholt werden muss, um die gesamte Gültigkeit des Umlaufplans abzudecken.

Bei aus FBS exportierten Umlaufplänen verweist jede *circulation.operatingPeriodRef* immer nur auf gleiche Wochentage (z. B. nur auf Montage oder nur auf Sonntage) - in FBS sind Umläufe immer wochentagsgenau. (Dies muss jedoch bei RailML-Dateien im Allgemeinen nicht immer so sein.) Wenn ein *circulation.operatingPeriodRef* z. B. auf drei aufeinanderfolgende Montage verweist, dann soll der Umlauf offensichtlich diese drei Wochen lang wiederholt werden - es ist dann immer auch ein geschlossener Umlauf. Im Grund- oder Regelumlauf einer Fahrzeugbaureihe sind zunächst i. d. R. Verweise für alle Wochentage der Fahrplanperiode enthalten. Dies ist die theoretische Ausgangssituation jeder Fahrplanperiode - ein ununterbrochen geltender Umlauf. Da einige „unterbrechend wirkende“ Ereignisse - z. B. Weihnachten, Ostern oder Ferien - jedoch bereits von vornherein feststehen, kann es ebenso sein, dass der Regelumlauf von Anfang an Unterbrechungen enthält oder von anderen Umläufen überschrieben wird.

Attribut	Inhalt	optional	Bemerkungen
blockRef	aktueller Block: Querverweis auf id eines Elements der Struktur <i>blocks</i> des Umlaufplans		
operatingPeriodRef	aktueller Umlaufstag: Querverweis auf id einer Verkehrstagerregelung der Struktur <i>timetabe.operatingPeriods</i>		
vehicleIdx	laufende Nummer des Fahrzeugs im Umlauf		Zählung ab 1, nur in V2.0.5
groupIdx	laufende Nummer der Umlaufgruppe	x	
vehicleCounter	laufende Nummer des Fahrzeugs im Umlauf		Zählung ab 1, erst ab V2.2
vehicleGroupCounter	laufende Nummer der Umlaufgruppe	x	
fbs:blatt	Verwendet für die Blattbezeichnung von „gefalteten“ Umlaufplänen (s. u.). Das Attribut enthält eine Bitmaske aus genau 7 Ziffern 0 oder 1, wobei die erste Ziffer für Montag und die letzte für Sonntag steht. Es ist immer mindestens eine 1 enthalten.	x	nur bei iRFP-eigenen Erweiterungen

fbs:zeile	Verwendet für die Zeilennummer in „gefalteten“ Umlaufplänen (auch „Tagesnummer“ genannt, jedoch nicht zu verwechseln mit Wochentagen).	x	
nextBlockRef	nachfolgender Block: Querverweis auf id eines Elements der Struktur <i>blocks</i> des Umlaufplans	x	nicht bei letzten Fahrten des jeweiligen Fahrzeugs in offenen Umläufen
nextOperatingPeriodRef	nachfolgender Umlauftag: Querverweis auf id einer Verkehrstagerregelung der Struktur <i>timetable.operatingPeriods</i>	x	

Die Attribute *vehicleCounter* (*vehicleIdx*) und *vehicleGroupCounter* (*groupIdx*) sind redundant, da sie genau genommen aus *nextBlockRef* und *nextOperatingPeriodRef* rekonstruiert werden könnten. Sie erleichtern das Auswerten jedoch deutlich, da der Zusammenhang zwischen *nextBlockRef*, *nextOperatingPeriodRef*, *vehicleIdx* und *groupIdx* u. U. nicht trivial ist:

- Bei einem offenen Umlauf ist die Anzahl der Fahrzeuge gleich der Anzahl der *circulation*-Elemente ohne *nextBlockRef* - wenn ein Fahrzeug durch einen offenen Übergang den Umlauf „verlässt“, muss für die folgende Fahrt ein anderes Fahrzeug verwendet werden.
- Bei einem geschlossenen Umlauf ist die Anzahl der Fahrzeuge gleich der Anzahl der *circulation*-Elemente, bei denen *nextOperatingPeriodRef* in der Zeit „zurückspringt“: Da ein Fahrzeug nicht gleichzeitig verschiedene Fahrten durchführen kann, enthält der zeitliche Rücksprung implizit das nächste Fahrzeug.
- Ein Umlauf ist geschlossen, wenn jedes *circulation*-Element die Attribute *nextBlockRef* und *nextOperatingPeriodRef* hat.
- Da es bei einem offenen Umlauf keine Wiederholungen gibt, gibt es auch nie zeitliche Rücksprünge von *nextOperatingPeriodRef*.
- Wenn ein *circulation*-Element durch seine Attribute *nextBlockRef* und *nextOperatingPeriodRef* auf einen Block verweist, der bereits durch ein früheres *circulation*-Element erfasst wurde, ist die Umlaufgruppe beendet (geschlossen). Wenn an dieser Stelle noch weitere Blöcke (d. h. unerfasste) im Umlauf sind, wird eine weitere Umlaufgruppe begonnen. Das nächste *circulation*-Element wird einen um eins erhöhten Gruppenindex haben.
- Die Gruppenindizes sind nur bei geschlossenen Umläufen angegeben und nur dann, wenn mehr als eine Gruppe im Umlauf vorhanden ist. Bei offenen Umläufen ist der Gruppenindex irrelevant (da ohnehin keine Wiederholungen stattfinden, ist es eine rein philosophische Ansicht, ob alle Fahrzeuge eine Gruppe bilden oder jedes Fahrzeug eine eigene Gruppe).

Die Attribute **fbs:blatt** und **fbs:zeile** stehen optional zur Verfügung, um Umlaufpläne in „gefalteter“ Darstellung wiedergeben zu können. In „gefalteten“ Umlaufplänen kann ein Blatt mehr als einen Wochentag darstellen, z. B. zusammengefasst Dienstag-Freitag oder Montag-Donnerstag. Es gilt die Regel, dass der Folgetag immer in der folgenden Zeile wiedergegeben ist. (Springe zur nächsten Zeile, um den nächsten Tag zu lesen.) Um gefaltete Umlaufplandarstellungen nach dieser Regel ausgeben zu können, exportiert FBS eine Tagesblatt-Codierung und eine Zeilennummer.

Änderungsübersicht

ab Profil-Version	Änderung
2.0.0	nachträglich für RailML 2.0 gemäß unveränderten Original-Schemen
2.0.1	erste veröffentlichte Basis-Version
2.0.2	keine Änderungen in der Ausgabe ggü. V2.0.1
2.0.3	<code><trainPart>.remarks</code> hinzugekommen Bei Wechsel der Bemerkungen (Attribut <i>remarks</i>) innerhalb des Laufwegs eines Zugteils wird dieser in zwei Zugteile aufgeteilt.
2.0.5	<code><train>.additionalTrainNumber</code> hinzugekommen
2.1.0	<p> <code><trackGroup>.<line>.uicNumber</code> → code (zusammengefasst) <code><trackGroup>.<line>.lineNumber</code> → code (zusammengefasst) <code><ocp>.abbreviation</code> → code (umbenannt) <code><ocp>.number</code> wird für IBNR verwendet entgegen der Deklaration als "deprecated", da ansonsten keine Betriebsstellennummern exportierbar wären <code><category>.abbreviation</code> → code (umbenannt) <code><ocpTT>.<sectionTT>.trackRef</code> zu <code><ocpTT>.<sectionTT>.<trackRef>.ref</code> verschoben <code><ocpTT>.<sectionTT>.<trackRef>.dir</code> hinzugekommen <code><ocpTT>.<sectionTT>.distance</code> Einheit von km auf m geändert (verbunden mit der Erhöhung von <code><metadata>.identifier</code> von 1 auf 2) <code><train>.<trainPartSequence>.categoryRef</code> (neu) – zur Unterscheidung Produkt (verkehrlich = <code><trainPart>.categoryRef</code>) und Gattung (betrieblich = <code><trainPartSequence>.categoryRef</code>) <code><circulation>.vehicleIdx</code> entfallen (ersatzlos, s. jedoch 2.2.0) <code><circulation>.groupIdx</code> entfallen (ersatzlos, s. jedoch 2.2.0) </p>
2.2.0	<p> <code><metadata>.<dc:language></code> umgestellt von Codepage auf BCP47-script <code><metadata>.<organizationalUnits></code> (neu) <code><line>.code</code> (Inhalt geändert) <code><line>.infrastructureManagerRef</code> (neu) <code><mileageChange>.dir</code> → <code>absDir</code> (umbenannt) <code><speedChange>.profileRef</code> (neu) <code><speedChange>.status</code> (entfallen) <code><speedChange>.trainRelation</code>, <code>mandatoryStop</code>, <code>signalised</code> (neu) <code><ownerChanges>.ownerName</code> (entfallen) <code><ownerChanges>.infrastructureManagerRef</code> (neu) <code><ocp>.code</code> (entfallen) → <code><ocp>.<designator></code> (neu) <code><ocp>.number</code> (entfallen) → <code><ocp>.<designator>.register='IBNR'...</code> (neu) <code><ocp>.<propOther>.<additionalName></code> → <code><ocp>.<additionalName>...</code> (verschoben) <code><ocp>.<propOther>.zip</code> (neu optional) <code><ocp>.<geoCoord>.coord</code> → Höhenangabe umgezogen zu <code>extraHeight</code> <code><ocp>.<geoCoord>/<gradientChange>.<geoCoord></code>: Koordinaten und Höhen werden nur noch mit Angabe eines EPSG-Codes exportiert (EPSG-Code ist Pflichtfeld geworden) <code><train>.trainNumber</code> (neu bei <code>type='commercial'</code>) <code><trainPart>.<formationTT>.orientationReversed</code> (neu) <code><trainPart>.<formationTT>.load</code> → <code>timetableLoad</code> (umbenannt) <code><trainPart>.<ocpsTT>.<ocpTT>.sequence</code> (neu) <code><trainPart>.<ocpsTT>.<ocpTT>.ocpType</code>: Entfall der Werte <i>begin</i> und <i>end</i> <code><trainPart>.<ocpsTT>.<ocpTT>.<stopDescription>.operationalStopOrdered</code> (neu) <code><trainPart>.<organizationalUnitBinding></code> ↔ benutzerdefinierte Felder aus FBS <code><circulation>.vehicleCounter</code> (wiedereingeführt als Nachfolge von <code>vehicleIdx</code>) <code><circulation>.vehicleGroupCounter</code> (wiedereingeführt als Nachfolge von <code>groupIdx</code>) </p>
2.2.1	<p> <code><ocp>.parentOcpRef</code> (neu, optional) <code><ocp>.code</code> (wiedereingeführt, neu optional und mit abweichendem Inhalt) </p>

(Fortsetzung auf folgender Seite)

allgemein 12/2014	<p>alle Versionen außer 2.0.5: kein initiales <mileageChange>-Element am Anfang eines Gleises</p>
allgemein 05/2015	<p>alle Versionen: <metadata>.<dc:source> erweitert auf Quelldateinamen aus FBS; auch unter <infrastructure> und <rollingstock> (nicht nur unter <railML>) <geoCoord>.coord: Höhenangabe vom dritten Wert in <i>coord</i> umgezogen zu <i>extraHeight</i> <geoCoord>.epsgCode/heightEpsgCode hinzugefügt (optional bis V2.1, obligatorisch ab V2.2) <speedChange>s werden fahrtrichtungsbezogen exportiert (dir='down', identifier=3) <speedProfile>.influence bei Grundprofilen immer = 'decreasingincreasing'^(Tippfehlerkorrektur 10/2017) <train>.scope='secondary' als möglicher Wert entfallen (bei type='operational') <train>.type='commercial': in zwei Modi möglich – optional mit 1:1-Zugteil-Zuordnung <trainPart>.remarks, debitcode, operator ↔ benutzerdefinierte Felder aus FBS <trainPart>.<ocpsTT>.<ocpTT>.<times>: Genauigkeit der Zeitangaben wählbar <trainPart>.<ocpsTT>.<ocpTT>.<times>.scope='published' (neuer Wert möglich) <circulation>.fbs:blatt, fbs:zeile (neu)</p>
allgemein 09/2015	<p>alle Versionen außer 2.0.5: Reihenfolgetausch von Längen- und Breitengrad bei Koordinaten nach Festlegung der Reihenfolge durch das railML-Konsortium. Um dies für externe Programme erkennbar zu machen, wurde <metadata>.identifier auf =4 erhöht. Bitte kontaktieren Sie iRFP, wenn Sie als Anwender Kompatibilitätsprobleme durch diese nachträgliche Änderung haben (Koordinaten werden falsch herum eingelesen).</p>