

Beschreibung der FBS-RailML-Import[®]-Schnittstelle

Stand: Juni 2012

RailML-Version: 2.0

RailML-Profil-Version: 2.0.2

Kompatibilitätsnummer: 1

Inhalt

Allgemeines.....	2
Kopfinformationen (Versionsnummern, Dublin Core Metadata Element Set).....	3
Infrastruktur (<i>infrastructure</i>).....	4
Strecken (<i>trackGroups</i>).....	4
Streckengleise (<i>tracks</i>).....	4
Betriebsstellen (<i>operationControlPoints</i>).....	5
Fahrzeug- und Zugbildungsdaten (<i>rollingstock</i>).....	5
Fahrzeugdaten (<i>vehicles</i>).....	5
Zugbildungsdaten (<i>formations</i>).....	6
Fahrplandaten (<i>timetable</i>).....	6
Fahrplanperioden (<i>timetablePeriods</i>).....	6
Verkehrstagerregelungen / Saisonierungen (<i>operatingPeriods</i>).....	6
Zuggattungen und Produktbezeichnungen (<i>categories</i>).....	7
Züge und Zugteile (<i>trains</i> und <i>trainParts</i>).....	8
Zugteile (<i>trainParts</i>).....	8
Laufweg von Zugteilen (<i>ocpsTT</i>).....	9
Züge (<i>trains</i>).....	10

© iRFP • Institut für Regional- und Fernverkehrsplanung
Niederlassung Leipzig
Fasanenweg 12
D-04420 Frankenheim bei Leipzig
Telefon: +49 341 9424508
Telefax: +49 341 9424507
www.irfp.de • leipzig@irfp.de



Niederlassung Dresden
Helmholtzstr. 1
D-01069 Dresden
Telefon: +49 351 4706819
Telefax: +49 351 4768190
dresden@irfp.de

Allgemeines

Die FBS-RailML-Import-Schnittstelle erlaubt einen Import von Fahrplandaten im RailML-Format. Eingelesen werden können 8-Bit-Textdateien in XML-Struktur. Derzeit werden XML-Schemen auf Basis des RailML[®]-Standards 2.0 verwendet, die für FBS geringfügig angepasst wurden. Die jeweils aktuelle Version der FBS-RailML-Schemendateien finden Sie unter

www.irfp.de/deutsch/fbs/schnittstelle_railml.html

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-Import-Schnittstelle obligatorisch oder optional ist. Viele der in RailML als optional deklarierten Attribute sind in der FBS-RailML-Import-Schnittstelle obligatorisch.

Die einzulesenden RailML-Dateien müssen grundsätzlich UTF-8-codiert sein. Eine RailML-Datei enthält auch Beschreibungen (Textdarstellung z. B. von Verkehrstagen) in der Sprache, die beim Exportieren als FBS-Programmiersprache 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 benötigte Umfang der Daten in der RailML-Datei nur eingeschränkt skalierbar. So müssen z. B. für den Import von Fahrplandaten auch Infrastrukturdaten in beträchtlichem Umfang in der RailML-Datei vorhanden sein, um den Laufweg des Zuges im Streckennetz eindeutig nachvollziehen zu können.

Hinweise zur Bedienung der Schnittstelle finden Sie im Dokument *FBS-RailML2Import - Hinweise zur Bedienung*, welches ebenfalls unter

www.irfp.de/deutsch/fbs/schnittstelle_railml.html zur Verfügung steht.

Die Implementierung der Schnittstelle ist zu keinem Zeitpunkt als abgeschlossen zu betrachten, d. h. es können jederzeit Erweiterungen erfolgen.

In der folgenden Beschreibung sind alle Strukturen und Attribute aufgeführt, die zum Zeitpunkt der Erstellung dieses Dokuments Bestandteil der FBS-RailML-Import-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 Id-Werte einer RailML-Datei müssen eindeutig sein. Es dürfen keine Verweise auf nicht existente Ids vorkommen. In den Id-Attributen darf kein spezieller (konkreter) Inhalt enthalten sein.

Kopfinformationen (Versionsnummern, Dublin Core Metadata Element Set)

Die zu importierenden RailML-Dateien müssen eine Angabe zur Versionsnummer des angewendeten RailML-Schemas aufweisen, damit eine korrekte Interpretation der Daten durch die Import-Schnittstelle möglich ist. Derzeit können nur RailML-Dateien eingelesen werden, in denen das Attribut **railml.version** einen Wert von **2.0** aufweist.

Die implementierungsabhängigen Versionsnummern und andere Kopfinformationen sind entsprechend dem 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. Die RailML-Import-Schnittstelle setzt eine Format-Version von **2.0.2** oder höher voraus.

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). Die RailML-Import-Schnittstelle setzt bei der RailML-Version 2.0 einen Wert des Attributs metadata.identifizier von **1** voraus.

Die weiteren Attribute des Metadata-Elements (metadata.language, metadata.source, metadata.date, metadata.creator) werden durch die Import-Schnittstelle nicht technisch ausgewertet.

Infrastruktur (infrastructure)

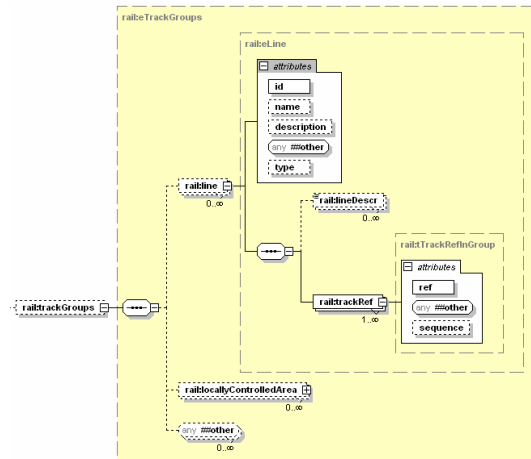
Zum Einlesen von Fahrplandaten wird auch ein beträchtlicher Teil an Infrastrukturdaten benötigt, um den Laufweg eines Zuges im Streckennetz eindeutig nachvollziehen zu können. Vorausgesetzt wird das Vorhandensein der Strukturen *operationControlPoints* mit den Betriebsstellen sowie *tracks* und *trackGroups* mit Strecken und Streckengleisen.

Strecken (trackGroups)

Die Struktur **trackGroups** enthält für jede Strecke einen Eintrag.

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.

Für den Import der Fahrplandaten wird die Struktur *trackGroups* benötigt, um dem Laufweg der Züge über die Attribute *uicNumber* und *lineNumber* Strecken des FBS-Netzes zuzuordnen.

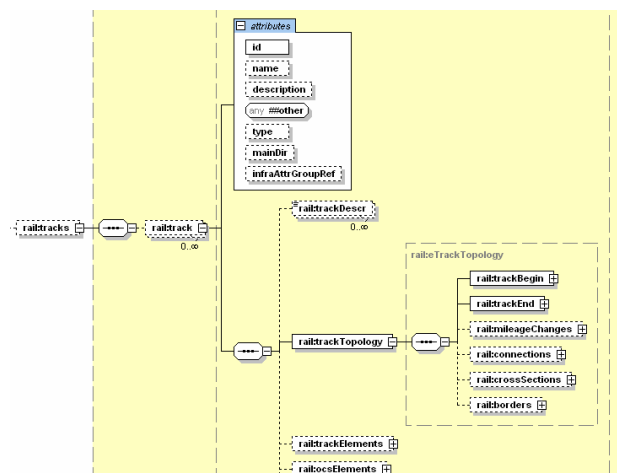


Attribut	Inhalt	optional	Bemerkungen
line.uicNumber	UIC-Nummer der Bahnverwaltung (EIU) der Strecke		
line.lineNumber	Streckennummer der Strecke		
line.trackRef.ref	Querverweis auf <i>id</i> eines Streckengleises in der <i>tracks</i> -Struktur		

Streckengleise (tracks)

Die Struktur **tracks** enthält (sofern vorhanden) Informationen zu jedem Streckengleis. Für den Import wird diese Struktur nur als Zwischenschritt benötigt, um eine Strecke (*trackGroup*) zu referenzieren (siehe *line.trackRef.ref*), um die dort definierte Streckennummer zu ermitteln. Aus diesem Grund werden keine Attribute dieser Struktur ausgewertet.

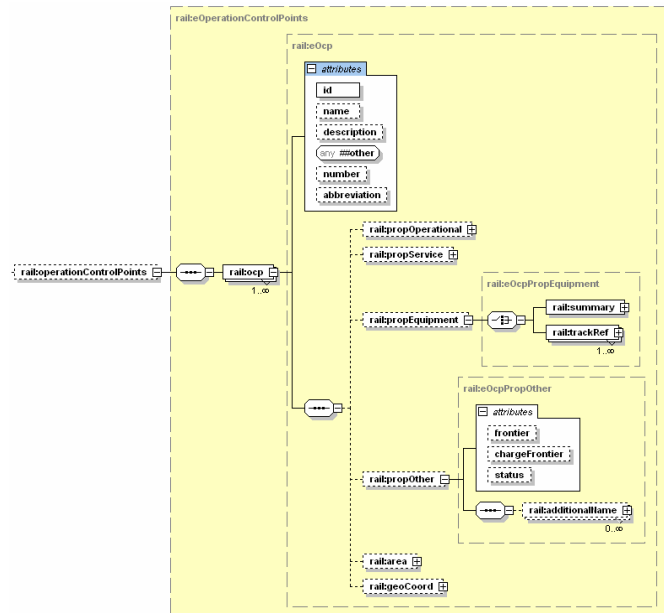
Das RailML-Schema setzt für die Struktur *tracks* eine Reihe von Attributen und Sub-Strukturen voraus, die allerdings beim Import von Fahrplandaten nicht berücksichtigt werden.



Betriebsstellen (*operationControlPoints*)

Die Struktur **operationControlPoints** enthält alle in den Fahrwegen der Züge der vorkommenden Betriebsstellen in loser Folge. Dies sind mindestens 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.

Für den Import von Fahrplandaten ist lediglich die betriebliche Abkürzung der Betriebsstelle relevant. Sie dient als Schlüssel zur Suche der Betriebsstelle im FBS-Netz.



Attribut	Inhalt	opti- onal	Bemerkungen
abbreviation	Abkürzung der Betriebsstelle		

Es ist möglich, dass in RailML mehrere Betriebsstellen die gleiche Abkürzung haben. Dies ist z. B. bei Infrastruktur der DB Netz AG bei Betriebsstellen an parallelen Strecken so (Beispiel: Haltepunkt Kürbitz, DB-Abkürzung DKUR; Templin Stadt, DB-Abkürzung WTV). Da beim Import der Fahrplandaten die Abkürzung jedoch als Primärschlüssel verwendet wird, werden alle RailML-Betriebsstellen mit der gleichen Abkürzung auf dieselbe FBS-Betriebsstelle abgebildet.

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 werden nur genutzt, um ein Fahrzeug zu identifizieren und ein passendes Fahrzeug des FBS-Netzes zuzuordnen. Für den importierten Zug werden innerhalb FBS ausschließlich die technischen Eigenschaften verwendet, die in Triebfahrzeug- und Wagenverzeichnis definiert sind. Diejenigen technischen Angaben der Fahrzeuge, die aus der RailML-Datei ausgelesen werden, dienen ausschließlich der eindeutigen Identifikation des Fahrzeuges durch den Anwender.

Attribut	Inhalt	opti- onal	Bemerkungen
name	Tfz.-Baureihe oder Wagengattung, ggf. ergänzt um Abkürzung des einstellenden EVUs (z. B. DB.234)		
numberDrivenAxles	Anzahl angetriebener Achsen	X	*1*2
length	Länge über alles in Metern		*2

speed	zulässige Höchstgeschwindigkeit in km/h		*2
bruttoWeight	Gesamtmasse in Tonnen		*2
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

*1 nur bei Triebfahrzeugen

*2 nur zur Anzeige bei manueller Identifizierung eines Fahrzeugs

Zugbildungsdaten (*formations*)

Ein **formation**-Element enthält die Zugbildung eines Zugteils. 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	op- ti- onal	Bemerkungen
speed	Minimum der zulässigen Höchstgeschwindigkeit aller Fahrzeuge in km/h	x	
weight	Summe der Bruttomasse aller Fahrzeuge in t	x	
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*)

Ein Import von Fahrplandaten ist nur möglich, wenn genau ein Element der Struktur **timetablePeriods** vorhanden ist. Deren Gültigkeit (Anfangs- und Enddatum) muss mit der der Fahrplanperiode des FBS-Netzes übereinstimmen.

Attribut	Inhalt	op- ti- onal	Bemerkungen
startDate	erster Gültigkeitstag der Fahrplanperiode		
endDate	letzter Gültigkeitstag der Fahrplanperiode		

Verkehrstagerregelungen / Saisonierungen (*operatingPeriods*)

Die Struktur **operatingPeriods** enthält für jede in den Fahrplandaten vorkommende Verkehrstagerregelung je einen Eintrag. Es sind nur Verkehrstagerregelungen importierbar, die sich auf eine definierte Fahrplanperiode beziehen.

Attribut	Inhalt	opti- onal	Bemerkungen
startDate	erster Gültigkeitstag der Verkehrstagerregelung	x	*1
endDate	letzter Gültigkeitstag der Verkehrstagerregelung	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).		

*1 Wenn nicht vorhanden, werden die Werte der Fahrplanperiode verwendet

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

Die verbale Beschreibung der Verkehrstagerregelung (z.B. werktags außer Samstag) wird derzeit ausschließlich aus der Bitmaske abgeleitet.

Zuggattungen und Produktbezeichnungen (categories)

Die Struktur **categories** enthält einen **category**-Eintrag für jede im Fahrplan vorkommende Zuggattung und Produktbezeichnung. Die durch die Import-Schnittstelle ausgewerteten Informationen dienen vorrangig dazu, die Gattung der RailML-Datei zu identifizieren und einer Gattung des FBS-Zugnummern-Verzeichnisses zuzuordnen.

Attribut	Inhalt	opti- onal	Bemerkungen
abbreviation	Kurzbezeichnung / Abkürzung der Zuggattung (z. B. „ICE“)		
name	Langbezeichnung der Zuggattung (z. B. „InterCityExpress“)	x	*1
description	Beschreibung der Verwendung der Zuggattung, z. B. „internationale Ganzgüterzüge“	x	*1
trainUsage	= <i>goods</i> bei Gattungen für Güterzüge = <i>passenger</i> bei Gattungen für Reisezüge = <i>mixed</i> bei Gattungen für beide Zugarten	x	*1
deadrun	= <i>true</i> , wenn die Zuggattung für einen Leerzug steht (Leerreise- oder Leergüterzug oder Tfz.-Leerfahrt)	x	*1

*1 nur zur Anzeige bei manueller Identifizierung einer Gattung

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	opti- onal	Bemerkungen																						
name	Wird als Zugteilnummer in FBS verwendet; kann alphanumerisch sein)																								
line	Linienbezeichnung des Zugteils	x																							
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) 'offered'</td> <td></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	x																							
categoryRef	Querverweis auf id des Eintrags der Produktbezeichnung / Gattung des Zugteils in der Struktur <i>timetable.categories</i> der RailML-Datei	x																							
formationTT. formationRef	Querverweis auf id des Eintrags der Zugbildung des Zugteils in der Struktur <i>rollingstock.formations</i> der RailML-Datei																								
operatingPeriodRef. ref	Querverweis auf Eintrag der Verkehrstageregelung 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>																								

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 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

Innerhalb von FBS bezieht sich die Eigenschaft Status auf den ganzen Zug. Da innerhalb von RailML der Status pro Zugteil (trainPart) definiert werden kann, wird beim Import der Fahrplandaten lediglich der Status des ersten Zugteils für den gesamten Zug übernommen.

Laufweg von Zugteilen (ocpsTT)

Die Struktur **ocpsTT** jedes *trainPart*-Elements enthält einen **ocpTT**-Eintrag für jede Betriebsstelle, die der Zugteil auf seinem Laufweg passiert. Es müssen immer mindestens zwei Betriebsstellen pro ocpTT-Struktur vorhanden sein.

Attribut	Inhalt	op-ti-onal	Bemerkungen
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		
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	
times.scope	Nur <i>times</i> -Strukturen mit <i>times.scope=scheduled</i> werden berücksichtigt		
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.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	* ³

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	x	*3*4
sectionTT.percentageSupplement	linearer Fahrzeitzuschlag zwischen der aktuellen und der nächsten Betriebsstelle in Prozent	x	
stopDescription...	Enthält Informationen zum Halt an der aktuellen Betriebsstelle	x	nicht vorhanden bei Durchfahrten
stopDescription.commercial	= <i>true</i> bei Verkehrshalt = <i>false</i> bei Betriebshalt	x	
stopDescription.stopOnRequest	= <i>true</i> bei Bedarfshalt (bedingtem Verkehrshalt) = <i>false</i> bei unbedingtem Verkehrshalt	x	
stopDescription.onOff	= <i>on</i> für „hält nur zum Einsteigen“ = <i>off</i> für „hält nur zum Aussteigen“	x	
stopDescription.stopTimes minimalTime	Regelaufenthaltszeit des Zuges an dieser Betriebsstelle (kann von der planmäßigen Aufenthaltszeit abweichen)	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 zur Nutzung der Strecke *sectionTT.lineRef* bzw. des Streckengleises *sectionTT.trackRef* können alternativ ausgewertet werden. Sind beide Attribute vorhanden, so wird *sectionTT.lineRef* ausgewertet.

*4 In jeder Struktur *sectionTT* darf nur eine Sub-Struktur *trackRef* vorhanden sein. Im Umkehrschluss bedeutet dies, dass im FBS-Netz an jedem Wechsel der Streckennummer eine ggf. virtuelle Betriebsstelle vorhanden ist und dass jede dieser Streckenwechsel-Betriebsstellen in der Struktur *ocpsTT* enthalten sein muss.

Die RailML-Datei enthält die Zeiten sekundengenau, wie es für XML definiert ist. 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 vor allem der Ankunfts- und Abfahrtszeiten beachtet werden.

Züge (trains)

Es werden nur betriebliche Züge (*type=operational*) ausgewertet.

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 (abschnittweisen) Verstärken oder Flügeln von Zügen vor. Aufeinanderfolgende Abschnitte eines Zuglaufs kommen z. B. vor, wenn unterwegs Verkehrstage wechseln.

Zu beachten ist, dass *trainPartRef.position* nicht unbedingt die tatsächliche Stellung im Zug angibt - durch unterschiedliche Verkehrstage der Zugteile können Teile mit niedrigerem *posi-*

tion-Wert an bestimmten Tagen im Zugverband „fehlen“. Beim Import werden die Zugteile in ihrer Reihenfolge gemäß *position* ins FBS-Netz übernommen.

RailML erlaubt das Vorhandensein mehrerer Triebfahrzeuge innerhalb eines Zugteils, innerhalb FBS ist dies nicht möglich. Wird beim Import erkannt, dass sich mehrere Triebfahrzeuge in einem Zugteil befinden, so wird dieser entsprechend mehrfach in FBS übernommen. Dadurch kann die Anzahl der Zugteile in der RailML-Datei und im FBS-Netz für einen Zug verschieden sein.

Attribut	Inhalt	op- ti- onal	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>	x	wenn nicht vorhanden, wird <i>primary</i> angenommen
additionalTrainNumber	dient der eindeutigen Bezeichnung des Zuges über alle Verkehrstage (Primärschlüsselbildung); laufende Nummer innerhalb gleicher <i>trainNumber</i> und <i>scope</i> -Werte	x	
trainPartRef.ref	Querverweis auf id eines Zugteils innerhalb der Struktur <i>timetable.trainParts</i>		
trainPartRef.position	Stellung des Zugteils im Zugverband, sofern ein Zug gleichzeitig aus mehreren Teilen besteht		Zählung ab 1
equipmentUsage.equipment	Zugbeeinflussung des Zuges.	x	*2
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.airBrakeApplicationPosition	nur bei Druck- oder Saugluftbremse: Grundbremsstellung = <i>G</i> , <i>P</i> oder <i>R</i>	x	
brakeUsage.meanDeceleration	mittlere Bremsverzögerung in m/s ²	x	*1
brakeUsage.regularBrakePercentage	im regulären (planmäßigen) Betrieb anrechenbare Bremsleistung (d. h. meist ohne Magnetschienenbremse)	x	*1
brakeUsage.emergencyBrakePercentage	im Gefahrenfall (bei Not-/Schnellbremsungen) anrechenbare Bremsleistung (d. h. mit Magnetschienenbremse, jedoch ggf. ohne elektrische oder hydraulische Bremse)	x	*1
brakeUsage.brakePercentage	wenn Schnell- und Betriebsbremsleistung identisch definiert sind	x	*1
brakeUsage.auxiliaryBrakes	Aufzählung der planmäßig vorhandenen Zusatzbremsen: <i>Mg</i> , <i>Wb</i> , <i>E</i> , <i>H</i> , <i>ep</i>	x	

*1 Entweder *brakeUsage.regularBrakePercentage* und *brakeUsage.emergencyBrakePercentage* (1. Priorität) oder aber *brakeUsage.brakePercentage* (2. Priorität) müssen gesetzt sein.

*2 Folgende Werte können ausgelesen werden: *Indusi54*, *Indusi60R*, *PZ80* und *PZB90* werden FBS-intern auf den Wert PZB abgebildet; *LZB* und *CIR-ELKE* werden FBS-intern auf den Wert LZB1 abgebildet.

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 (mit Index=1)	„Stammfahrplan“ oder „Stammzug“
	Hauptlauf (mit Index>1)	- <i>nicht möglich</i> -
train.scope=secondaryStart	Vornebenlauf	„Startflügel“
train.scope=secondaryEnd	Nachnebenlauf	„Zielflügel“
train.scope=secondaryInner	Zwischennebenlauf	„Doppelfahrplan“