

Interface description for Viriato 8.11

Liike

10/5/2016

Content

1. Introduction	1
1.1 Conventions used in this document	1
1.1.1 Syntax	1
1.1.2 Minimal Requirements, import strategies	1
1.1.3 Occurrence of elements	1
1.1.4 Use of attributes	1
2. Envelope Information	2
2.1 Element <capacityID>	2
2.2 Element <command>	2
2.3 Element <logging>	3
2.4 Element <error>	3
2.5 Element <railml>	3
3. General structure of the schema	4
4. Train Information (railML)	5
4.1 Master Data (MD)	5
4.2 Timetable	6
4.2.1 Element <timetable>	6
4.2.2 Element <operatingPeriod>/<service>	6
4.2.3 Element <timetablePeriod>	7
4.2.4 Element <train>	8
4.2.5 Element <train>	10
4.2.6 Element <timetableentries>	11
4.2.7 Element <entry>	11
4.2.8 Element <composition>	13
4.2.9 Element <formation>	14
4.2.10 Element <engine>	15
4.2.11 Element <wagon>	15
4.2.12 Element <service>	16
4.2.13 Element <dynamic>	16

4.2.14 Element <section>	16
4.2.15 Element <stopDescription>	17

Table of versions

Version	Datum	Autor	Bemerkung	Status
0.01	29.07.2016	udi	Initial Version / Transfer from old document	draft
0.02	19.09.2016	cf	Content revision	draft
1.00	05.10.2016	mho, sh	Review and release	released

1. Introduction

This document describes the xml-schema for the Liike interface. The main xml-schema is called 'LIIKE-Interface.xsd' and includes several other schemas to describe train data according to the railML standard (version 1.1). These schemas are part of the Liike interface.

For the general structure and the use of the schema LIIKE-Interface.xsd see (→ 3. General structure of the schema)

For the logical description and a detailed description of the interface see (→ 2. Envelope Information) and (→ 4. Train Information (railML))

1.1 Conventions used in this document

1.1.1 Syntax

Throughout this document an XML element is written as <element> while an XML attribute is written as {attribute}.

1.1.2 Minimal Requirements, import strategies

The Viriato-Liike interface has to handle railML trains referencing master data that are missing or unknown on importing a train. Therefore the interface uses two different strategies, depending on the attribute:

- checked: missing or unknown data leads to import abortion
- default: missing or unknown data is substituted by default values

1.1.3 Occurrence of elements

Any element or sub element may be required or optional. There are four different kinds of occurrences:

- one: 1

required, only one set of data allowed

- one or more: 1..∞

required, several sets of data allowed

- none or one: 0..1

optional, only one set of data allowed

- None, one or more: 0.. ∞

optional, several sets of data allowed

1.1.4 Use of attributes

Some information is written when exporting data, but is ignored while importing it. For instance the {description} of a <timetablePeriod> is not necessary for importing data, if {timetableID} is filled correctly.

This document focuses on the import case. Therefore the fields related to master data will be transferred via ID and will be "checked" while importing.

- required: data is required for import and will be exported
- optional: data could be imported and will be exported

2. Envelope Information

The envelope information is held by an 'envelopeType', which contains the following five elements.

Element	Description
capacityID	Contains the capacityID as a string. The capacityID must consist of a unique operator ID and an operator-wide unique ID. Mandatory element
command	Contains the information about the type of request and response exchanged by the operators and the capacity manager. Mandatory element
logging	Logging information for the command, contains time, user and free text. Mandatory element
error	In case an error occurred processing a request it is possible to give some information about the error (error code and description). Optional element
railml	This is a particular train attached to the request and is a valid railML standard node. Optional element (as in some cases the train information is not needed, e.g. when deleting a train)

2.1 Element <capacityID>

Every <envelope> contains exactly one capacityID. The capacityID is a unique identifier for a train in FTA Viriato. As the capacityID can be requested by any operator requesting to a train to be allocated, the capacityID has to have a part identifying the operator (unique within the Finnish rail system) and a part identifying the train (unique among all trains of one operator).

<capacityID> is a plain string. It complies with following structure XX.Z.nnnn, where XX are two alphanumeric characters identifying the operator, Z identifies the system that created the train and can take one of two values, L for Liike and V for Viriato, nnn is a sequence of numbers (examples VR.L.0123456789, VR.V.0123456789).

2.2 Element <command>

The <command> element consists of two sub elements as described in the following table.

Element	Possible value	Description
request	CREATE	The operator sends the request to allocate a new train.
	MODIFY	The operator sends the request to modify a train already imported by FTA Viriato.
	CANCEL	The operator sends the request to cancel / remove a train already imported by FTA Viriato.
response	REJECTED	The capacity manager rejects a requested train which has already been confirmed.
	CONFIRMED	The capacity manager confirms a requested train. This means the train is imported to FTA Viriato but is not processed or allocated yet. After such a response the state of the train on the side of the operator is frozen until a decision from the capacity manager is received.
	ALLOCATED	The capacity manager has allocated the train without modifications.
	DELETED	The capacity manager has deleted a train imported or already allocated.
	MODIFIED	The capacity manager modified a train requested by an operator.

2.3 Element <logging>

The <logging> element is mandatory and consists of two string sub elements:

- <text> Free text describing this request (mandatory, text can be empty)
- <user> The user name doing this request (mandatory)

The attribute {timestamp} is a mandatory attribute containing the time stamp of when the request was sent.

2.4 Element <error>

Not supported by Viriato.

2.5 Element <railml>

The <railml> element contains all the information about the train. It is built according to the railML standard. The elements <rollingstock>, <infrastructureVisualisation> and <infrastructure> are not supported in this interface.

It is possible that a Liike request (or response) is sent without containing the train information (particularly when deleting a train). In this case the <railml> element is not needed as the train is identified by the capacityID.

For the detailed description see (→ 4. Train Information (railML)).

3. General structure of the schema

'LIKE-Interface.xsd' defines two elements: <railml> of type railmlType and <envelope> of type envelopeType. Therefore it is possible to create an XML file with either a railml or an envelope type.

The 'LIKE-Interface.xsd' is used for exchanging information in both directions, from an operator to FTA (Liike) and from FTA (Liike) to the operator.

To send a valid request according to the schema you must use an envelope root element which must contain the request information and which can contain a railml element which describes a train according to the railML 1.1 standard.

```
<xml>
<envelope>
  <capacity-id>xyz-abc</capacity-id>
  <request>...</request>
  ...[more envelope information]
  <railml>
    <timetable>
      ...
    <train>
      [train information]
    </train>
  </railml>
</envelope>
```

Figure 1 Structure of a Liike – xml – request file. The railml – node (in blue, bold) can be used as an own valid railml node.

In 'LIKE-Interface.xsd', are the description of both 'envelopeType' and 'railMLType'. The details of the 'railmlType', are mostly defined in included xml-schema files adapted from the railML 1.1 standard.

4. Train Information (railML)

This XML schema for the railmlType is based on the railML standard (version 1.1). Therefore an XML file of the Liike interface is also valid railML.

On the other hand, valid railML must not necessarily include enough information to be valid for an import in Viriato.

4.1 Master Data (MD)

Master data includes the description of the infrastructure, e.g. the identification of the operational points (station, junctions) and the lines or sections. Other elements of the master data include the timetable period and the product (or train type such as intercity, suburban, express cargo etc.). If the capacity manager has the authority for the sectional running times, also the information about the used locomotives belongs to the master data.

A problem with the master data may occur when transferring only IDs. The ID must be interpreted in the same way by the sender and the receiver of the data. Unknown master data in the import will be treated as incorrect, therefore lists of all valid master data must have been declared externally before importing.

Master data is typically referenced with required fields. Referencing wrong or unknown data is treated as data inconsistency and will lead to the abortion of the import.

Master Data	ID	Description
train type	kind	Describes the type of product of a train (intercity, regional, suburban, container block train, mixed cargo etc.)
timetable period	timetablePeriodID	Defines the timetable period within which a train is running. The capacity manager typically has to deal concurrently with multiple timetable periods (ad hoc requests for the current timetable, applications for the next timetable)
node sequence	posID	An enumeration of operational points describes the train path. It contains at least the start point of the path and possibly intermediate points, where the train has a planned stop or where it changes to another section.
section	sectionID	Each node of the train path is followed by the identification of the section the train is running on.
operator	operator	List of accepted operators
braking system	brakingSystem	The chosen braking type while running ("G", "P", "R", "R+Mg", "R+Wb")
rolling stock	vehicleID	List of engines

For several of these fields, the usage as "required" or "optional" still has to be declared by the capacity manager. For instance the rolling stock with an engine specified with {vehicleID} is required for correct running time calculations, but may be optional or still unknown for the slot ordering process.

- "required" fields in combination with master data (MD) have to be "checked" at the import.
- "optional" fields are combined with "default" import strategies if available.

In this document fields are often declared as "optional", but they have to be "checked" if they are used as master data.

The train paths exported from Viriato contain all nodes of type station or junction even if the train does not stop there. At nodes where the train does not stop, only the departure time is used and the arrival time is left empty.

If nodes have no defined running time in Viriato they are exported either without minRunTime or without arrival and departure times. Their departure has to be interpolated linearly. For the interpolation only the distance between the section nodes of this sequence and the next node with a defined run time in the train path is relevant. The speed is irrelevant in the interpolation. The interpolated departure time on a section node is the previous departure time added to the proportional run time relative to the distance to the previous section node in the train path.

If nodes are missing in a train path imported into Viriato, the programme adds the nodes when constructing the train from the import and interpolates the running times. If the import contains nodes unknown to Viriato, the request is rejected.

4.2 Timetable

4.2.1 Element <timetable>

Every file for import or export contains exactly one <timetable>.

Attributes:

Name	Type	Use	Values	Description
version	double	required	1.1	version of the timetable schema used in the interface

Sub elements:

Name	Occurrence	Description
operatingPeriods	one	one list of operating periods that can be referenced in the trains
timetablePeriods	one or more	one list of timetable periods an operating period and therefore a train can belong to
train	one or more	<p>A train can be either:</p> <ul style="list-style-type: none"> - a single train - a train of a group of interval trains (trains with the same {intervalGroupID}). This option is not used in the Liike interface. Each train of an interval family is written to a separate file. - a train of a train group (trains with the same {trainGroupID})

Additional conditions:

A "train group" means trains with the same {trainGroupID}. Operationally it may be considered to be one train, but if any property of a train such as it's configuration or the operating period changing within the train path, it has to be treated as two separate trains (within a train group).

4.2.2 Element <operatingPeriod>/<service>

All operating periods referenced by any train must be listed in the "operatingPeriods" element. They are used together with the timetable periods to determine the validities of the imported trains.

Attributes:

Name	Type	Use	Values	Description
serviceID	string	required		ID of operating period or validity

If both the start date and the end date of the timetable period are set, all timetable periods in Viriato with this start date and this end date are considered. There must be at least one timetable period in Viriato with the given start and end date. If there is one with the value of the "timetablePeriodID" attribute as its ID, this is the matching timetable period. Otherwise the timetable period with the (alphanumerically) smallest ID is the matching timetable period.

Attributes:

Name	Type	Use	Values	Description
timetablePeriodID	string	required		ID of the timetablePeriod
description	string	optional		description of the timetable period, ignored on import to Viriato
startDate	date	optional		start date of the timetable period
endDate	date	optional		end date of the timetable period

Sub elements: -

Example: Timetable period 2016

```
<timetablePeriods>
  <timetablePeriod timetablePeriodID="2016" description="2016" startDate="2015-12-13" endDate="2016-12-10"/>
</timetablePeriods>
```

4.2.4 Element <train>

A train in the interface is always a single train, running according to its <operatingPeriod>. It may be one train of a set of interval trains, but all members of such a train family are transferred as separate capacities in the interface.

Forbidden changes within a train

To meet the integrity requirements of a train in Viriato some properties are not allowed to change.

Unchangeable attributes of a train are:

- trainID
- lineID
- intervalGroupID
- trainNumber
- description
- timetablePeriodID

Other unchangeable properties of a train are:

- The train path
- Stops within the train path
- Arrival and departure times

To submit such changes of train properties to the interface (e.g. "on Mondays via XY", "stops in XY only on Tuesdays", "arrival in XY on Sundays 2 minutes later"), it is necessary to split the train into several separate trains and send them separately to the interface.

Allowed changes of train path or timetable

Apart from the fixed properties of a train's timetable, other characteristics such as the operating period may change within the train path (e.g. a reduced train path on Sundays) or a capacity increase by adding a second EMU on part of the train path.

Since such changes are not supported in the interface, these changes have to be described by using several sub trains, being connected by the {trainGroupID}. These train parts may be recomposed to one train in a system that supports changes of train properties.

Example:

A train starting in "HKI" changes its <formation> at "ILA" and further on its <service> at "HPL" before arriving in VMO". This has to be described by 3 different trains connected by their {trainGroupID}.

```
<railml>
<train trainID="t 1 x0020 IC x0020 943.1 2" trainGroupID="943" trainNumber="943">
  <timetableentries>
    <entry posID="HKI" type="begin" departure="06:28:00.0">
      <composition compID="0" kind="IC">
        <formation formationID="Sr2-3201--46" weight="400" length="200" speed="210">
          <engine vehicleID="10?Sr2?3201-46?210"/>
        </formation>
        <service serviceID="-88330378"/>
        <dynamic brakingSystem="R" brakedWeightPercentage="125"/>
      </composition>
      <section sectionID="120" trackID="PR" distance="3.068"/>
    </entry>
    <entry posID="PSL" type="pass">
      <connection trainID="t 1 x0020 IC x0020 943.1 2" connOperation="none"/>
    </entry>
  </timetableentries>
</train>
<train trainID="t 1 x0020 IC x0020 943.1 2" trainGroupID="943" trainNumber="943">
  <timetableentries>
    <entry posID="ILA" minStopTime="PT1M" type="stop" arrival="06:33:00.0" departure="06:34:00.0">
      <composition compID="1" kind="IC">
        <formation formationID="Sr2-3201--46|Sr2-3201--46" weight="400" length="200" speed="210">
          <engine vehicleID="10?Sr2?3201-46?210"/>
          <engine vehicleID="10?Sr2?3201-46?210"/>
        </formation>
        <service serviceID="-88330378"/>
        <dynamic brakingSystem="R" brakedWeightPercentage="125"/>
      </composition>
      <section sectionID="120" trackID="PR" distance="0.267"/>
    </entry>
    <entry posID="KHK" type="pass">
      <connection trainID="t 1 x0020 IC x0020 943.1 2" connOperation="none"/>
      <section sectionID="120" trackID="PR" minRunTime="PT2M" distance="1.646"/>
    </entry>
  </timetableentries>
</train>
<train trainID="t 1 x0020 IC x0020 943.1 2" trainGroupID="943" trainNumber="943">
  <timetableentries>
```

```

<entry posID="HPL" minStopTime="PT2M" type="stop" arrival="06:36:00.0" departure="06:38:00.0">
  <composition compID="1" kind="IC">
    <formation formationID="Sr2-3201--46|Sr2-3201--46" weight="400" length="200" speed="210">
      <engine vehicleID="10?Sr2?3201-46?210"/>
      <engine vehicleID="10?Sr2?3201-46?210"/>
    </formation>
    <service serviceID="705560421"/>
    <dynamic brakingSystem="R" brakedWeightPercentage="125"/>
  </composition>
  <section sectionID="120" trackID="PR" minRunTime="PT2M" distance="1.106"/>
</entry>
<entry posID="VMO" type="end" arrival="06:40:00.0" publishedArrival="06:40:00.0"/>
</timetableentries>
</train>
</railml>

```

4.2.5 Element <train>

The interface contains only one train from the operational point of view. This train may consist of several <train> elements, but only if they are connected by the same {trainGroupID} as described above.

Attributes:

Name	Type	Use	Values	Description
trainID	string	required		Unique key within the interface
trainGroupID	string	optional		TrainID used to group all train parts of one train within the interval train, built to the following pattern: (trainNumber)
lineID	string	required		MD, line number will be checked during import
intervalGroupID	string	required		Used to group all trains of one train family.
trainNumber	string	required		train number
trainStatus	string	optional		not supported by Viriato
debitcode	integer	optional		not supported by Viriato
remarks	string	optional		ignored by Viriato on import.
description	string	optional		train description e. g. "Turku-Helsinki"
seatCapacity	integer	optional		not supported by Viriato
timetablePeriodID	string	required		references a timetable period defined in element <timetablePeriods>, MD, will be checked during import
categoryID	string	optional		not supported by Viriato
operator	string	optional		Operator on train level (MD), will be checked during import
dataSource	string	optional		not supported by Viriato
dataDateTime	dateTime	optional		not supported by Viriato
dataStatus	string	optional		not supported by Viriato
dataDescription	string	optional		not supported by Viriato

Sub elements:

Name		Occurrence	Description
timetableentries		one	timetable data describing the train path, <timetableentries> is used for grouping <entry> elements
ns:orderer	string	none or one	not supported by Viriato
ns:trainName	string	none or one	not supported by Viriato
ns:intervalServiceID	string	none or one	not supported by Viriato
ns:operatingDay	string	none or one	not supported by Viriato
ns:intervalTrainStatus	string	none or one	not supported by Viriato
ns:typeRunCalc	integer	none or one	not supported by Viriato
ns:calcRemark	string	none or one	not supported by Viriato
ns:notes	string	none or one	not supported by Viriato
ns:isFreightTrain	boolean	none or one	not supported by Viriato
ns:addRunInPercent	boolean	none or one	not supported by Viriato
ns:arrDepFlag	string	none or one	not supported by Viriato
ns:calcRTDate	datetime	none or one	not supported by Viriato
ns:intervalOperator	string	none or one	not supported by Viriato
ns:intervalRemarks	string	none or one	notes defined on the train family
ns:trainInfluence	string	none or one	not supported by Viriato
ns:speedIndex	integer	none or one	not supported by Viriato
ns:serviceHints		none or one	not supported by Viriato

4.2.6 Element <timetableentries>

This element is a container for a list of entries.

Attributes: -

Sub elements:

Name	Occurrence	Description
entry	two or more	section node within the train path

4.2.7 Element <entry>

An <entry> in the interface describes an operational point within the train path, typically a station. An <entry> contains the information in the station or while arriving at the station. Therefore there is no running time or section track in the <entry>. Such information is contained in the sub element <section>.

above.

Attributes:

Name	Type	Use	Values	Description
posID	string	required		MD, ID of node

				4 letters abbreviation
trackID	string	optional		Platform ID in the station
trackInfo	string	optional		Name of the Platform selected as Track No in the train, empty if not defined explicitly in the timetable, ignored by Viriato on import.
arrival	time	optional		operational arrival time
arrivalDay	integer	optional	must not be negative	day of arrival, train is starting on day 0 default=0
departure	time	optional		required at least on the first <entry>, operational departure time or passing time
departureDay	integer	optional	must not be negative	day of departure, train is starting on day 0 default=0
publishedArrival	time	optional		arrival time for the customer
publishedDeparture	time	optional		departure time for the customer
minStopTime	duration	optional	must be positive	minimum required stop time
addStopTime	duration	optional	might be negative	stop time buffer used for operational purposes
supStopTime	duration	optional	might be negative	stop time buffer used for other purposes (e.g. timetable optimization)
type	string	optional	stop, pass, begin, end	first/last node with "begin"/"end" stops within path with "stop", others "pass" default=pass ignored on import to Viriato
remarks	string	optional		additional remarks for the train at this station
trainReverse	boolean	optional		not supported by Viriato
alignment	string	optional	head, center, rear	not supported by Viriato
offset	double	optional		not supported by Viriato

Sub elements:

Name		Occurrence	Description
composition		None or one or more	compositions, required at the first <entry> of a <train> and allowed only there
connection		none or one or more	not supported by Viriato
section		none or one	section data between this and next entry, required at all <entry> elements of a <train> except the last <train> in the

			same train group
stopDescription		none or one	stop data for this entry ignored on import to Viriato
ns:isFixingNode	boolean	none or one	not supported by Viriato
ns:addRunTimePercent	double	none or one	not supported by Viriato
ns:operationalStop	string	none or one	MD, abbreviation defined for operational stop
ns:movementType	string	none or one	Liike specific string that describes the type of movement, JUNA: Junaliik (commercial), VAIH: Vaihtoty (non-commercial) default: JUNA
ns:defaultTrackInfo	string	none or one	not supported by Viriato

Additional conditions:

- Arrival of a single train may not be later than the day after its departure
- The last train of an interval group of trains may not arrive later than the day after the first train of the interval group started
- All trains of an interval family must start on the same day
- Stops are only allowed in certain nodes, depending on the node type (MD) and the <stopDescription>.
- no stop time at first/last node of first/last train with same {trainGroupID}

Example: Train is stopping 30 seconds in TKL and passing through HNA.

```
<entry posID="TKL" arrival="09:33:30.0" departure="09:34:00.0" type="stop"
publishedArrival="09:33:00.0" publishedDeparture="09:33:00.0" minStopTime="PT0M30S">
  <section sectionID="140" trackInfo="101"/>
</entry>

<entry posID="HNA" arrival="09:37:00.0" departure="09:37:00.0" type="pass">
  <section sectionID="140" trackInfo="201"/>
</entry>
```

4.2.8 Element <composition>

The <composition> describes the operating period of a train as well as the used rolling stock and properties about its operation. This train data may not change within the path of the train. Therefore a <composition> at the first <entry> element is required, but not in further entries describing the train path.

Attributes:

Name	Type	Use	Values	Description
compID	string	required		unique key for the composition within the train
kind	string	required		MD, train type ID
kindNo	string	optional		not supported by Viriato

Sub elements:

Name		Occurrence	Description
formation		one	rolling stock data
service		one	operating days
dynamic		none or one	properties of train operation
ns:userVelMax	double	none or one	maximum speed for timetable construction, must be greater than zero if specified default=if given, lowest technical maximum speed of all vehicles in the train else default=0
ns:trainConfiguration	string	none or one	not supported by Viriato
ns:places1	integer	none or one	not supported by Viriato
ns:places2	integer	none or one	not supported by Viriato
ns:pushPull	boolean	none or one	true = train configuration is ready for push-pull default=false
ns:tilting	boolean	none or one	true = train configuration is ready for tilting default=false
ns:trainConstruction	boolean	none or one	not supported by Viriato
ns:trainSet	string	none or one	not supported by Viriato
ns:ETCS	string	none or one	not supported by Viriato
ns:speedProfile	integer	one	MD, Speed profile of the train configuration

Example:

```

<composition compID="11 H 420 0" kind="H">
  <formation formationID="0" length="25.2" speed="120" weight="0">
    <engine vehicleID="Sr2"/>
    <wagon vehicleID="" count="7"/>
  </formation>
  <service serviceID="M-L"/>
  <dynamic brakingSystem="R" brakedWeightPercentage="0" timeSurcharge="0"/>
</composition>

```

4.2.9 Element <formation>

A <formation> element within the <composition> describes the properties of the rolling stock. It appears at the first <entry> but is valid for the total train path. The rolling stock consists of the locomotive (referenced via {vehicleID} which is master data), information about the wagons (weight, length) and the maximum speed (technical maximum of engine and coaches) of the formation.

Attributes:

Name	Type	Use	Values	Description
formationID	string	optional		ID for the formation
weight	double	optional		weight of train without locomotive, given in tonnes
length	double	optional		length of train with locomotive, given in metres
speed	double	optional		maximum speed, the train formation is able to run, given in km/h

Sub elements:

Name		Occurrence	Description
engine		none, one or more	MD, locomotives, referenced by {vehicleID}
wagon		none, one or more	MD, wagons, referenced by {vehicleID}

Example: engine information with vehicle type, series and maximum speed of the vehicle

```

<formation formationID="0" speed="120">
  <engine vehicleID="10?Sr2?3201-46?200"/>
  <wagon vehicleID="20?Chfy?29901?160"/>
  <wagon vehicleID="20?Chfa?19901?140"/>
</formation>

<formation formationID="1" length="132" speed="200" weight="250">
  <engine vehicleID="10?Sr2?3201-46?200"/>
</formation>

```

4.2.10 Element <engine>

This element references the used engine.

Attributes:

Name	Type	Use	Values	Description
vehicleID	string	required		MD, ID of one engine

The vehicleID consists of the following values separated with a question mark:

- vehicle kind: "10"
- vehicle type
- vehicle series
- maximum speed in km/h *

* If the maximum speed is left out, the first found engine with this vehicle type will be taken on import.

4.2.11 Element <wagon>

This element references the used wagon.

Attributes:

Name	Type	Use	Values	Description
vehicleID	string	required		MD, ID of one wagon
count	integer.	optional	Must be positive	not supported by Viriato

The vehicleID consists of the following values separated with a question mark:

- vehicle kind: "20"
- vehicle type
- vehicle series
- optional: maximum speed in km/h *

* If they maximum speed is left out, the first found wagon with this vehicle type will be taken on import.

4.2.12 Element <service>

A <service> within the <composition> references an operating period. It appears at the first <entry> but is valid for the total train path.

Attributes:

Name	Type	Use	Values	Description
serviceID	string	required		ID of a service, defined in <operatingPeriod>

4.2.13 Element <dynamic>

The element <dynamic> lists information about the train properties. This is all optional, because it may be unknown.

Attributes:

Name	Type	Use	Values	Description
brakingSystem	string	optional		MD, braking system
brakeWeightPercentage	integer	optional		must be greater than zero, if used for running time calculation
timeSurcharge	double	optional		not supported by Viriato
loadSurcharge	double	optional		not supported by Viriato

Sub elements:

There are no sub elements.

Example:

```
<dynamic brakingSystem="R" brakeWeightPercentage="150" timeSurcharge="0"/>
```

4.2.14 Element <section>

This element describes data concerning the section between the current section node and the next section node.

Attributes:

Name	Type	Use	Values	Description
------	------	-----	--------	-------------

sectionID	string	required		ID of the section used until the next <entry>
trackID	string	optional		ID of the section track used until the next <entry> default=default section track taken from the master data of the target database
minRunTime	duration	optional		technical (minumum) run time needed to reach the next <entry> with a technical running time
addRunTime	duration	optional		buffer for operational purposes, additional to running time
supRunTime	duration	optional		buffer for other purposes (e. g. timetable optimization), additional on running time
distance	double	optional		distance to the next <entry> in [km] default= taken from the master data in the target database

Sub elements:

There are no sub elements.

4.2.15 Element <stopDescription>

This element describes the type and the purpose of the stop.

Attributes:

Name	Type	Use	Values	Description
commercial	boolean	optional		true=stop is commercial ignored on import to Viriato
stopOnRequest	boolean	optional		true=stop is stop on demand
onOff	string	optional	on, off, both	describes the stop type, on=boarding only off=alighting only both=boarding and alighting default=both
purpose	string	optional		free text to describe other purposes for the stop

Sub elements:

There are no sub elements.